Trustworthy speaker recognition with minimal prior knowledge using neural networks

Thèse N° 7285

Présentée le 29 novembre 2019

à la Faculté des sciences et techniques de l’ingénieur
Laboratoire de l’IDIAP
Programme doctoral en génie électrique

pour l’obtention du grade de Docteur ès Sciences

par

Hannah MUCKENHIRN

Acceptée sur proposition du jury

Prof. D. N. A. Van De Ville, président du jury
Prof. H. Bourlard, Dr M. Magimai Doss, directeurs de thèse
Prof. . N. Evans, rapporteur
Dr P. Bell, rapporteur
Prof. M. Jaggi, rapporteur

2019

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

To my parents

Acknowledgements

This thesis would not have been possible without the help and support of many people, to whom I am extremely grateful.

First and foremost, I would like to thank my two PhD supervisors Mathew Magimai Doss and Sébastien Marcel for their guidance during this journey. I would also like to thank my PhD committee for reviewing this thesis and providing useful comments: Dr. Peter Bell, Prof. Nicholas Evans and Prof. Martin Jaggi, as well as Prof. Dimitri Van de Ville for presiding the jury.

I am thankful to my colleagues and friends at Idiap. First, to the past and present members of the biometrics group: Alain, Andre, Anjith, Amir, Artur, David, Guillaume, Ketan, Laurent, Michael, Olegs, Pavel, Saeed, Sushil, Teodors, Tiago, Vedrana and Zohreh. Second, to all the Idiapers that have made my life in Martigny enjoyable, in particular: Angel, Ajay, Angelos, Apoorv, Banri, Bastian, Cijo, Dhananjay, Enno, Francois, Gulcan, Hande, Julian, Nikos, Parvaneh, Pavan, Pierre-Edouard, Pranay, Sandrine, Skanda, Sibo and Weipeng.

During my PhD, I had the opportunity to do an internship in the speaker ID team at Google. This was a very enriching experience and I am grateful to Quan, Prashant, Ignacio and Jason for their mentorship and help. I would also like to thank Laurent for his help and advice before and during the internship.

I would like to thank Benoit for his positive impact and support during these four years. Last but not least, I would like to thank my parents, my sister and my brother for their unconditional support and love.

Abstract

The performance of speaker recognition systems has considerably improved in the last decade. This is mainly due to the development of Gaussian mixture model-based systems and in particular to the use of *i-vectors*. These systems handle relatively well noise and channel mismatches and yield a low error rate when confronted with zero-effort impostors, i.e. impostors using their own voice but claiming to be someone else. However, speaker verification systems are vulnerable to more sophisticated attacks, called presentation or spoofing attacks. In that case, the impostors present a fake sample to the system, which can either be generated with a speech synthesis or voice conversion algorithm or can be a previous recording of the target speaker. One way to make speaker recognition systems robust to this type of attack is to integrate a presentation attack detection system.

Current methods for speaker recognition and presentation attack detection are largely based on short-term spectral processing. This has certain limitations. For instance, state-of-the-art speaker verification systems use cepstral features, which mainly capture vocal tract system characteristics, although voice source characteristics are also speaker discriminative. In the case of presentation attack detection, there is little prior knowledge that can guide us to differentiate bona fide samples from presentation attacks, as they are both speech signals that carry the same high level information, such as message, speaker identity and information about environment.

This thesis focuses on developing speaker verification and presentation attack detection systems that rely on minimal assumptions. Towards that, inspired by recent advances in deep learning, we first develop speaker verification approaches where speaker discriminative information is learned from raw waveforms using convolutional neural networks (CNNs). We show that such approaches are capable of learning both voice source related and vocal tract system related speaker discriminative information and yield performance competitive to state of the art systems, namely *i-vectors* and *x-vectors*-based systems. We then develop two high performing approaches for presentation attack detection: one based on long-term spectral statistics and the other based on raw speech modeling using CNNs. We show that these two approaches are complementary and make the speaker verification systems robust to presentation attacks. Finally, we develop a visualization method inspired from the computer vision community to gain insight about the task-specific information captured by the CNNs from the raw speech signals.

Keywords: speaker recognition, presentation attack detection, convolutional neural networks, raw waveforms, gradient-based visualization.

Résumé

La performance des systèmes de reconnaissance du locuteur s'est considérablement améliorée au cours de la dernière décennie. Ceci est principalement dû au développement de systèmes basés sur des modèles de mélange gaussiens et en particulier à l'utilisation de *i-vectors*. Ces systèmes gèrent relativement bien le bruit et les variations des canaux d'enregistrement et produisent un faible taux d'erreur lorsqu'ils sont confrontés à des imposteurs "sans effort", c'est-à-dire des imposteurs qui utilisent leur propre voix tout en prétendant être quelqu'un d'autre. Cependant, les systèmes de vérification du locuteur sont vulnérables à des attaques d'usurpation d'identité plus sophistiquées, appelées attaques de présentation ou de *spoofing*. Dans ce cas, les imposteurs présentent un faux échantillon au système, qui peut être généré avec un algorithme de synthèse ou de conversion vocale ou qui peut être un enregistrement antérieur du locuteur cible. Il est possible de rendre les systèmes de reconnaissance de locuteurs robustes à ce type d'attaque en y intégrant un système de détection d'attaque de présentation.

Les méthodes actuelles de reconnaissance du locuteur et de détection des attaques de présentation reposent largement sur l'utilisation de transformations spectrales à court terme, ce qui a certaines limites. Par exemple, les systèmes de vérification du locuteur de l'état de l'art utilisent des représentations cepstrales, qui capturent principalement les caractéristiques du conduit vocal, bien que les caractéristiques de la source vocale sont également importants. Dans le cas de la détection d'attaque de présentation, peu de connaissances préalables peuvent nous aider à différencier les échantillons authentiques des attaques de présentation, car il s'agit de signaux qui contiennent les mêmes caractéristiques, telles que le contenu, l'identité du locuteur et les informations relatives à l'environnement.

Cette thèse porte sur le développement de systèmes de vérification et de détection d'attaque de présentation qui reposent sur une utilisation minimale d'hypothèses. Pour ce faire, inspirés par les récents progrès de l'apprentissage en profondeur, nous développons d'abord des approches de vérification du locuteur dans lesquelles des informations discriminantes concernant le locuteur sont apprises à partir des signaux brutes à l'aide de réseaux de neurones à convolution (CNN). Nous montrons que de telles approches sont capables d'apprendre des informations discriminantes sur le locuteur, liées aux sources vocales et aux conduits vocaux, et d'offrir des performances compétitives par rapport à l'état de l'art, à savoir des systèmes basés sur l'extraction de *i-vectors* et *x-vectors*. Nous avons ensuite développé deux approches très performantes pour la détection d'attaque de présentation : l'une basée sur le calcul de statistiques spectrales à long terme et l'autre sur la modélisation des signaux bruts à l'aide de

CNN. Nous montrons que ces deux approches sont complémentaires et rendent les systèmes de vérification du locuteur robustes aux attaques de présentation. Enfin, nous développons une méthode de visualisation inspirée de travaux sur la vision par ordinateur afin de mieux comprendre les informations capturées par les CNN à partir des signaux bruts.

Contents

List of Figures

List of Tables

List of acronyms

ASV Automatic speaker verification

CNN Convolutional neural network

CQCC Constant Q cepstral coefficients

DNN Deep neural network

EER Equal error rate

EPSC Expected Performance and Spoofability Curve

FC Fully connected

FMR False Match Rate

FNMR False Non Match Rate

GMM Gaussian mixture model

HTER Half total error rate

IAPMR Impostor Attack Presentation Match Rate

LDA Linear discriminant analysis

MFCC Mel-frequency cepstral coefficient

minDCF minimum of the detection cost function

MLP Multi-layer perceptron

PAD Presentation attack detection

PLDA Probabilistic linear discriminant analysis

RNN Recurrent neural network

TDNN Time-delay neural network

UBM Universal background model

VAD Voice activity detection

WER Weighted error rate

1 Introduction

Speaker recognition is the process of authenticating or identifying a person from the characteristics of his/her voice. The performance of speaker recognition systems has considerably improved in the past years. They are now used in commercial applications such as banking authentification and virtual assistant [Chen et al., 2015]. However these systems have been shown to be vulnerable to presentation attacks [Kucur Ergunay et al., 2015, Wu et al., 2015a], also called spoofing attacks. These attacks are composed of forged or altered samples that try to emulate the voice of the person of interest and can be generated in several manners: the sample can be artificially created either with a speech synthesis or a voice conversion algorithm or the attacker can use previous recordings of the speaker. To counter these attacks, binary classification systems need to be trained to detect whether a sample is bona fide or is an attack. The speaker verification system and the presentation attack detection need then to be combined, as illustrated in Figure 1.1.

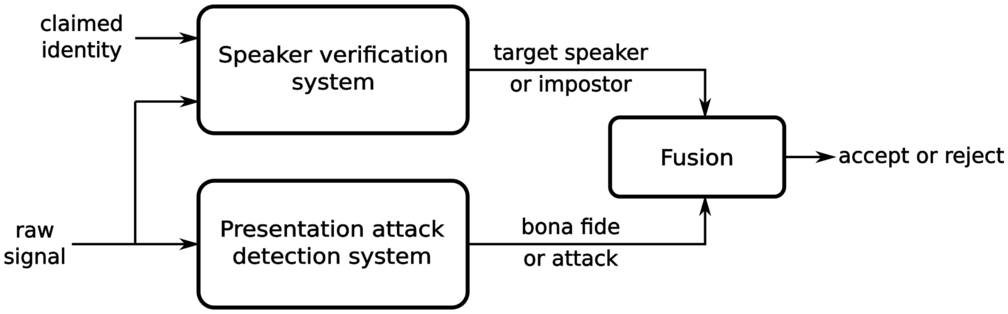

claimed identity
raw signal
Speaker verification system
Presentation attack detection system
target speaker or impostor
bona fide or attack
Fusion
accept or reject

Figure 1.1 – Fusion of a speaker verification and presentation attack detection system.

This thesis focuses on developing speaker verification and presentation attack detection systems that rely on minimum prior knowledge by modeling raw waveforms with neural networks.

1.1 Motivations

State-of-the-art speaker verification and presentation attack detection systems are mostly based on the derivation of short-term spectral features such as Mel-frequency cepstral coefficients. These engineered features rely on knowledge about speech production and perception. They were originally developed for speech coding and speech recognition and mainly characterize the vocal tract system. The use of these features in speaker verification and presentation attack detection systems might be sub-optimal for two main reasons. First, these features contain information about the lexical content, speaker characteristics and environment. Thus, compensation methods are needed to suppress irrelevant information such as the lexical content. Secondly, the characteristics needed for both tasks do not depend solely on vocal tract characteristics. In the case of speaker recognition, speakers characteristics are spread across many dimensions, such as source-related characteristics or prosody patterns. Using only vocal tract information constraints the system. While in the case of presentation attacks, there is little or no prior knowledge about what features to extract. The extracted features should be independent of the speaker and the lexical content and should provide information that differentiates genuine accesses against attacks.

Thus, for both tasks, this thesis takes some distance from current state-of-the-art systems and develops speaker verification and presentation attack detection systems that rely on minimal prior knowledge. To do so, it leverages recent findings in machine learning, which have shown that relevant features and classifier can be learned directly from the raw signal [Palaz et al., 2013, Tüske et al., 2014, Sainath et al., 2015, Trigeorgis et al., 2016, Zazo et al., 2016, Kabil et al., 2018]. Specifically, this thesis is built upon an EPFL PhD thesis [Palaz, 2016], which showed that automatic learning of features and classifier from raw waveforms to estimate phoneme class conditional probabilities leads to better systems than conventional approaches with fewer parameters.

1.2 Objectives and contributions

The goal of this thesis is to:

1. develop approaches to learn speaker discrimination and presentation attack detection by directly modeling raw speech signals with minimal prior knowledge using neural networks; and

2. gain insight into the information learned by such neural networks.

The main contributions of the thesis are the following:

- We develop CNN-based speaker verification systems trained on raw speech that outperform conventional and neural network-based systems. We also show that such neural networks are capable of learning speaker discrimination at voice source and vocal tract system levels.

- We develop two presentation attack detection approaches that do not rely on conventional short term spectral features. The first one is based on spectral statistics while the second one relies on CNNs trained on raw speech, inspired by our successful experiments for speaker verification. We show that the fusion of the two systems yields the best performance on two different databases.
- We demonstrate that combining the two first contributions produces speaker verification systems robust to presentation attacks.
- Taking inspiration from the computer vision community, we develop an approach to analyze what information is extracted from raw speech signals by CNNs.

1.3 Outline

Chapter 2, *Background*, gives an overview of the research field of speaker recognition and presentation attack detection. It also describes the evaluation metrics and the databases used in this thesis.

Chapter 3, *Raw waveform-based CNNs for speaker verification*, develops approaches to model raw waveforms for speaker verification using CNNs and validates them on two different databases. It then analyzes the information captured in the first convolution layer.
Related publications:

- H. Muckenhirn, M. Magimai.-Doss and S. Marcel, "Towards directly modeling raw speech signal for speaker verification using CNNs", in Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018.
- H. Muckenhirn, M. Magimai.-Doss and S. Marcel, "On Learning Vocal Tract System Related Speaker Discriminative Information from Raw Signal Using CNNs", in Proceedings of Interspeech, 2018.
- V. Abrol, H. Muckenhirn, M. Magimai.-Doss and S. Marcel "Learning Complementary Speaker Embeddings in End-to-End Manner from Raw Waveforms", manuscript under preparation.

Chapter 4, *Trustworthy speaker verification*, is concerned with the “trustworthiness” of the speaker verification systems. It first investigates the vulnerability of different speaker verification systems: the systems proposed in Chapter 3 as well as state-of-the-art systems. Then, it proposes two presentation attack detection methods relying on minimal prior knowledge. Finally, it investigates the impact of fusing speaker verification and presentation attack detection systems.

Related publications:

- H. Muckenhirn, M. Magimai.-Doss and S. Marcel, "Presentation Attack Detection Using Long-Term Spectral Statistics for Trustworthy Speaker Verification", in Proceedings of International Conference of the Biometrics Special Interest Group, 2016.
- H. Muckenhirn, P. Korshunov, M. Magimai.-Doss and S. Marcel, "Long-Term Spectral Statistics for Voice Presentation Attack Detection", IEEE/ACM Transactions on Audio, Speech and Language Processing, 25(11):2098-2111, 2017.
- H. Muckenhirn, M. Magimai.-Doss and S. Marcel, "End-to-End Convolutional Neural Network-based Voice Presentation Attack Detection", in Proceedings of International Joint Conference on Biometrics, 2017.

Chapter 5, *Visualizing and understanding raw waveform-based neural networks*, presents a gradient-based visualization method inspired from the computer vision community. This method enables to get better insights about the task-specific information that is learned from the raw waveforms by the CNNs.
Related publications:

- H. Muckenhirn, V. Abrol, M. Magimai.-Doss and S. Marcel, "Understanding and Visualizing Raw Waveform-based CNNs ", in Proceedings of Interspeech, 2019.

Chapter 6, *Conclusion*, concludes the thesis with a summary of salient findings.

2 Background

This chapter presents an overview of the fields of speaker recognition and presentation attacks. It is divided into four sections. Section 2.1 provides an overview of speaker recognition, its main approaches, as well as the metrics employed to evaluate them. Section 2.2 defines presentations attacks and summarizes the main countermeasures. Section 2.3 describes how to evaluate the vulnerability of speaker recognition systems to presentation attacks. Finally, Section 2.4 presents the databases that will be used in this thesis.

2.1 Speaker recognition

2.1.1 Definitions

Speaker recognition corresponds to the task of authenticating or recognizing individuals through their voices. It can be divided into two tasks: speaker identification and speaker verification. In a speaker identification task the goal is to identify a speaker from a set of speakers. This is thus a multiclass classification problem. In a speaker verification task, the goal is to verify whether a voice sample belongs to a given speaker or not. This is a binary classification or a hypothesis testing problem. More specifically, speaker verification systems have two phases: enrollement and test. During the enrollment phase, a speaker-specific model is created and stored. During the test phase, the system take two inputs: a speech sample and an identity. The system then decides whether the identity claim can be accepted or not, based on the model created during the enrollment phase. This process is illustrated in Figure 2.1. In this thesis, although we deal with both tasks, we are mainly interested in speaker verification.

There are two types of speaker verification systems: text-independent and text-dependent. In the first case, the speaker has no constraint on what to say, which is useful when the speaker is uncooperative, e.g., in forensics. In the second case, the speaker has to utter a pre-defined text, which is either fixed or prompted for each probe. This is useful for example for security applications or for virtual assistants, where it is often coupled with keyword spotting. In a text-dependent scenario, the systems usually achieve a higher accuracy with shorter enrollment

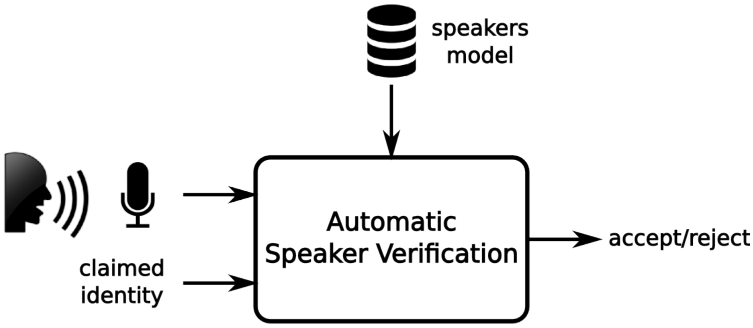


Figure 2.1 – Speaker verification system.

duration since there are constraints on the spoken content, i.e. the variability is lower than in the text-independent scenario.

2.1.2 Speaker characteristics

Speech signal carries different types of information such as the lexical content, speaker characteristics and environment. The first step is thus to be able to find information in speech samples related to the speakers characteristics. These are either related to biological traits or to learned behavioral patterns. For example, the voice of a person is characterized by the length of the vocal tract as well as by the average fundamental frequency. The learned aspects can be related to characteristics such as accent, dialect and prosody patterns. Designing features that capture these characteristics from the speech samples is still an open research problem. Furthermore, such features should be robust to different variabilities, such as age, health, emotions and noise. Finally, the captured characteristics should not be easily modifiable by the speaker.

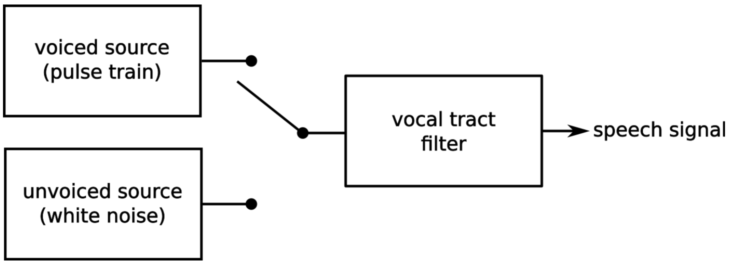


Figure 2.2 – Source-filter model of speech production.

The speech production system is often modelled as a source-filter system, illustrated in Figure 2.2, in which the source corresponds to the glottal excitation and the filter corresponds to the vocal tract system. Most systems developed nowadays are based on cepstral features such as Mel-Frequency Cepstral Coefficients (MFCC) [Davis and Mermelstein, 1980] or Perceptual Linear Prediction (PLP) [Hermansky, 1990] computed over frames of 20-30ms. These features were originally designed for the task of automatic speech recognition and model vocal tract characteristics, such as the formants, i.e., the resonance frequencies of the vocal tract. Vocal tract characteristics depend both on the speaker characteristics and on the uttered sound. Thus, as it will be explained in Section 2.1.3, compensation methods are needed to remove irrelevant information for the task of speaker recognition.

Intuitively, source-related features such as fundamental frequency should be adequate for this task, as they are supposed to be speaker specific. However such features have not been shown to outperform ceptral-based features when used alone. In fact, in one of the earlier work on speaker discriminative features, it was shown that higher formants are more speaker discriminative than source-related features [Sambur, 1975]. On the other hand, it was found that voice source-related features improve the recognition performance when fused with cepstral features [Yegnanarayana et al., 2001].

2.1.3 Approaches

In this section, we present the main approaches to perform speaker verification. We first describe the evolution of the different Gaussian Mixture Models (GMMs)-based systems, which have been the backbone of speaker recognition for the last two decades. A detailed overview can be found in [Kinnunen and Li, 2010] and [Hansen and Hasan, 2015]. We then present the more recent neural network-based approaches.

GMM-UBM approach

A GMM is a mixture of K multi-variate Gaussian components. In this framework, the probability that a sample $\mathbf{x}$ was uttered by the target speaker is:

$$p(\mathbf{x}|\lambda_{target}) = \sum_{k=1}^{K} p_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$

subject to $\sum_{k=1}^{K} p_k = 1$ and $p_k \geq 0$, $\forall k = 1, \ldots, K$. $\mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$ is a multivariate Gaussian distribution with mean μ_k and covariance matrix Σ_k.

The GMM training consists in estimating the parameters $\lambda_{target} = \{p_k, \mu_k, \Sigma_k\}_{k=1}^{K}$. The enrollment of the target speaker is relatively short (a few seconds) and is not enough to estimate these parameters from scratch. Instead, a Universal Background Model (UBM) [Reynolds et al., 2000] is employed to model speaker-independent characteristics of speech signals. It is obtained by training a GMM on a large set of speakers to represent speech characteristics. When a speaker is enrolled in the system, the UBM is used as a prior model and its parameters are adapted with a Maximum a Posteriori method to fit the speaker data. It was shown in [Reynolds et al., 2000] that it is sufficient to adapt only the mean μ_k.

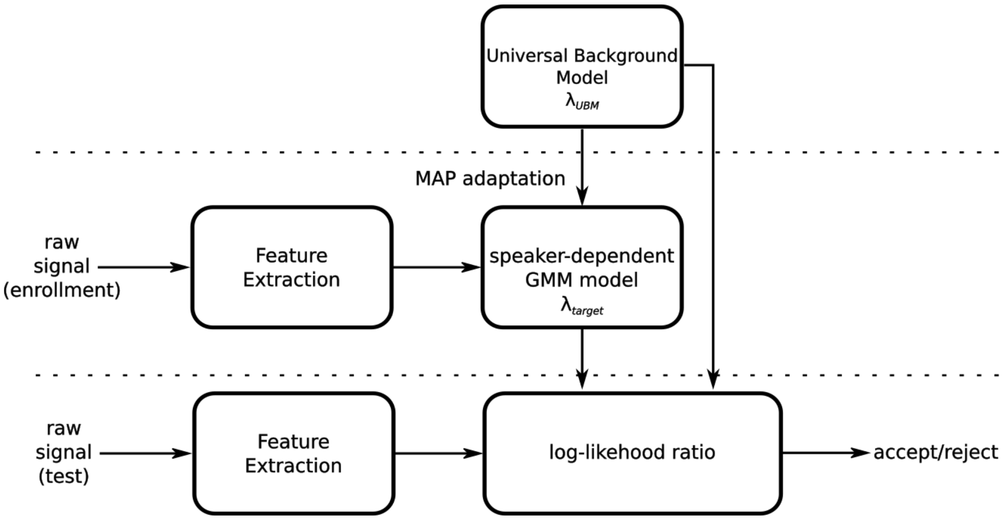


Figure 2.3 – GMM-UBM approach. The universal backgound model is trained offline on the training set.

The task of speaker verification can be seen as an hypothesis testing.

- H_0: **x** is uttered by the target speaker.

- H_1: **x** is not uttered by the target speaker.

The alternative hypothesis H_1 is represented by the UBM. The decision is then taken based on the computation of the likelihood ratio:

$$\frac{p(\mathbf{x}|H_0)}{p(\mathbf{x}|H_1)} = \frac{p(\mathbf{x}|\lambda_{target})}{p(\mathbf{x}|\lambda_{UBM})} \gtrless \theta$$

The GMM-UBM approach is illustrated in Figure 2.3.

Supervectors

Another approach proposed in [Campbell et al., 2006] is to extract supervectors instead of computing a likelihood ratio. A supervector corresponds to the concatenated means of a GMM. If the input has a dimension d and the GMM is composed of K Gaussian components, then the supervectors have a size Kd. The supervectors are then used as feature vectors, classified for example with a Support Vector Machine (SVM). This approach is illustrated in Figure 2.4.

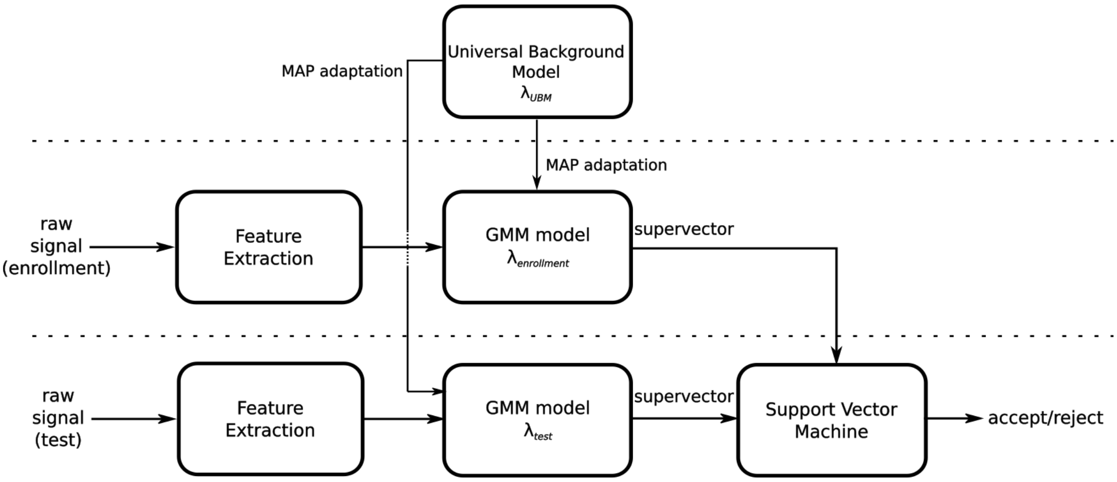


Figure 2.4 – Supervectors-based approach The universal backgound model and the parameters of the support vector machine are trained offline on the training set.

Session variability compensation

The supervectors contain both speaker characteristics and session variability, due to channel or lexical content mismatch. Several works have proposed compensation techniques through the use of latent variable models. The two main methods are inter-session variability (ISV) [Vogt and Sridharan, 2008] and joint factor analysis (JFA) [Kenny et al., 2007]. In these models each sample corresponds to a different session.

$\mu_{i,j}$ corresponds to the supervector of speaker i and utterance (and thus session) j.

The ISV approach uses the following latent model:

$$\mu_{i,j} = \mathbf{m} + \mathbf{U}\mathbf{x}_{i,j} + \mathbf{D}\mathbf{z}_i, \tag{2.1}$$

where **m** is the concatenated means of the UBM, **U** is a low-dimensional rectangular matrix that models the within-speaker variability and **D** is a diagonal matrix. $\mathbf{x}_{i,j}$ and $\mathbf{z}_i \sim \mathcal{N}(0, \mathbf{I})$. The within-speaker variability component $\mathbf{U}\mathbf{x}_{i,j}$ is removed and the supervector becomes:

$$s_i^{\text{ISV}} = \mathbf{m} + \mathbf{D}\mathbf{z}_i \tag{2.2}$$

The JFA approach uses the following latent model:

$$\mu_{i,j} = \mathbf{m} + \mathbf{V}\mathbf{y}_i + \mathbf{U}\mathbf{x}_{i,j} + \mathbf{D}\mathbf{z}_i, \tag{2.3}$$

where **V** is a low-dimensional rectangular matrix and $\mathbf{y}_i \sim \mathcal{N}(0, \mathbf{I})$. As done in the ISV approach, the within-speaker variability component is subtracted and the supervector becomes:

$$s_i^{\text{JFA}} = \mathbf{m} + \mathbf{V}\mathbf{y}_i + \mathbf{D}\mathbf{z}_i \tag{2.4}$$

i-vectors

In [Dehak, 2009], it was shown that JFA actually fails at separating the between and within class variance. Instead, the authors [Dehak et al., 2011] proposed a projection that does not make such distinction:

$$\mu = \mathbf{m} + \mathbf{T}\mathbf{v}, \tag{2.5}$$

where **T** is a low rank matrix and **v** is a low-dimensional vector, called *i-vector*. The total variability matrix **T** is estimated on the training set, i.e., with the data used to train the UBM.

Since that projection does not remove any session variability, *i-vectors* need to be further processed with compensation methods. Two popular methods are the within-class covariance normalization (WCCN) [Hatch et al., 2006] and the probabilistic linear discriminant analysis (PLDA) [Prince and Elder, 2007]. PLDA is a generative probability model, which models within-class and between-class variations, and performs both session compensation and classification.

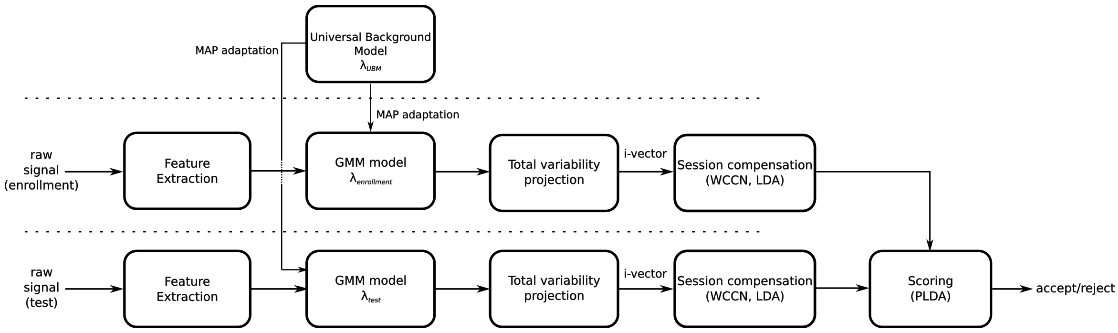

Universal Background Model λ_{UBM}
MAP adaptation
MAP adaptation
raw signal (enrollment)
Feature Extraction
GMM model $\lambda_{enrollment}$
Total variability projection
i-vector
Session compensation (WCCN, LDA)
raw signal (test)
Feature Extraction
GMM model λ_{test}
Total variability projection
i-vector
Session compensation (WCCN, LDA)
Scoring (PLDA)
accept/reject

Figure 2.5 – *i-vectors*-based appproach. The universal background model, total variability matrix and the parameters of the different session compensations and/or scoring methods are trained offline on the training set.

Neural network-based systems

In recent years, neural networks have become an important part of speaker recognition systems. Neural networks are trained on either MFCCs [Chen and Salman, 2011], output of filterbanks [Variani et al., 2014, Heigold et al., 2016] or spectrograms [Zhang and Koishida, 2017, Nagrani et al., 2017]. They were first used to replace the UBM in the *i-vector* framework [Kenny et al., 2014, Lei et al., 2014]. A neural network was trained for speech recognition to predict senone posteriors. It was then used to compute the Baum-Welch statistics, which are necessary for the derivation of *i-vectors*.

Another use of neural networks is to extract speaker embeddings, also called bottleneck features. Speakers embeddings are used as feature vectors and should be a robust representation of speakers characteristics. To compute these embeddings, a neural network is first trained to discriminate between speakers, i.e., it is trained to solve a speaker identification task. This training is done on a large number of speakers and is akin to a UBM training, except that it is a discriminative instead of a generative training. Once the network has been trained, the embeddings are obtained by forwarding a sample and extracting the output of a specific layer (either a bottleneck layer or the penultimate layer). If needed, the embeddings are then aggregated (usually by averaging them over the utterance) and used as a feature vector. They are then classified either with a simple cosine distance metric or with more elaborated classifiers such as a PLDA. Such an approach was first presented in [Variani et al., 2014] in a text-dependent scenario, where a fully connected neural network is trained on filterbank energy features and is used to extract frame-level embeddings, called *d-vectors*. In [Nagrani et al., 2017], a VGG-inspired neural network is trained on spectrograms to extract frame-level embeddings. In both works, utterance-level (respectively speaker-level) embeddings are obtained by simply averaging all the frame-level embeddings of an utterance (respectively speaker). The verification scores are then obtained by computing a cosine distance between enrollment and test utterances. In [Snyder et al., 2018] a time-delay neural network (TDNN) is trained on MFCCs to extract utterance-level embeddings called *x-vectors*. This is achieved through the use of a global statistics layer which aggregates frame-level input to obtain an utterance-level output. These embeddings are then projected into a lower dimensional space with LDA and classified with PLDA. Some end-to-end approaches have also been proposed. For example in [Nagrani et al., 2017] the authors train a siamese CNN network, which outperforms the embedding-based approach.

2.1.4 Evaluation

To train and assess the performance of a speaker verification system the data should be divided into three subsets [Friedman et al., 2001] with non-overlapping speakers [Lui et al., 2012]: a *training* set, a *development* set and an *evaluation* set. The training set usually contains a large number of speakers and is used for the initial training of the system. In conventional UBM-GMM based systems this corresponds to training the background model. In neural network-based systems, this set is used to train the network for speaker identification. The development and evaluation sets are usually split into 2 subsets: enrollment and probe set. The enrollment data is used to create a model for a given speaker and the probe set is used during the verification phase, also called test phase. The parameters of the trained systems, e.g., the decision threshold, are tuned on the development set. Finally, the evaluation set is used to evaluate the performance of the system once all the parameters are fixed.

Speaker verification is a hypothesis testing problem. Thus, two types of error can occur:

- a false acceptance error, i.e., accepting an impostor claim.
- a false rejection error, i.e., rejecting a genuine speaker claim.

Two measures are derived from these two types of error. The false acceptance rate (FAR), which corresponds to the number of false acceptance errors divided by the number of negative samples and the false rejection rate (FRR), which corresponds to the number of false rejection errors divided by the number of positive samples. The decision threshold will balance the values of FAR and FRR. One standard criterion to choose this threshold is the equal error rate (EER), i.e., a threshold at which the FAR and FRR are as close as possible:

$$\tau_* = \arg\min_{\tau} |\mathrm{FAR}_\tau - \mathrm{FRR}_\tau|$$

Once this threshold has been fixed on the development set, several measures are employed on the evaluation set. A popular metric is the half total error rate (HTER), which corresponds to:

$$\mathrm{HTER}_{\tau_*} = \frac{\mathrm{FAR}_{\tau_*} + \mathrm{FRR}_{\tau_*}}{2}$$

HTER measures the performance of a system at one operating point. Graphical representations such as the detection error trade-off (DET) can compare systems at different operating points by plotting the FAR against the FRR in a logarithmic scale.

Some databases are split into two subsets instead of three: training set and evaluation set. In that case, the most common evaluation metrics are either the EER or the minimum of the detection cost function (minDCF) [Martin and Przybocki, 2000]. The detection cost function is a weighted sum of false acceptance and false rejection rate and is defined in the following manner:

$$C_{\mathrm{DCF}} = C_{\mathrm{FR}} P_{\mathrm{FR}|target} P_{target} + C_{\mathrm{FA}} P_{\mathrm{FA}|non\ target}(1 - P_{target}) \tag{2.6}$$

$P_{\mathrm{FR}|target}$ and $P_{\mathrm{FA}|non\ target}$ are the system-dependent false rejection rate and false acceptance rate. C_{FR} is the cost of a false rejection, C_{FA} is the cost of a false acceptance and P_{target} is the prior probability of the target speaker. In practice, the costs of false rejection and false acceptance C_{FR} and C_{FA} are set to 1. P_{target} are set to small values such as 0.01 or 0.001. The minimum value of this cost function is reported.

2.2 Presentation attack detection

Like any biometric system, speaker verification-based authentication systems are vulnerable to attacks. In this section we first define what are presentation attacks. We then provide an overview about the countermeasures proposed in the literature for presentation attack detection. Finally, we briefly present the metrics used to evaluate the presentation attack detection systems.

2.2.1 Attacks

A speaker verification system can be attacked at different points [Ratha et al., 2001], as illustrated in Figure 2.6. In this thesis, our interest lies in attacks at point (1) and point (2), called spoofing attacks or presentation attacks, where the system can be attacked by presenting a spoofed signal as input. It has been shown that speaker verification systems are vulnerable to such elaborated attacks [Kucur Ergunay et al., 2015, Wu et al., 2015a]. As for points of attack (3) - (9), the attacker needs to be aware of the computing system as well as the operational details of the biometric system. Preventing or countering such attacks is more related to cyber-security, and is thus out of the scope of the present thesis.

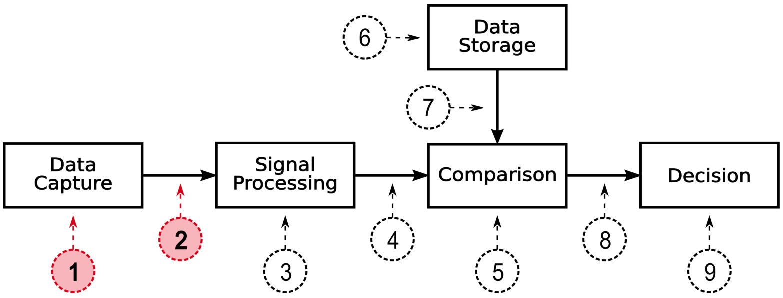


Figure 2.6 – Potential points of attack in a biometric system, as defined in the ISO-standard 30107-1 [ISO/IEC JTC 1/SC 37 Biometrics, 2016a]. Points 1 and 2 correspond respectively to attacks performed via physical and via logical access.

Attack at point (1) is referred to as *presentation attack* as per ISO-standard 30107-1 [ISO/IEC JTC 1/SC 37 Biometrics, 2016a] or as *physical access attack*. Formally, it refers to the case where falsified or altered samples are presented to the biometric sensor (microphone in the case of speaker verification system) to induce illegitimate acceptance. Attack at point (2) is referred to as *logical access attack* where the sensor is bypassed and the spoofed signal is directly injected into the speaker verification system process. The main difference between these two kinds of attacks is that in the case of physical access attacks, the attacker, apart from having access to the sensor, needs less expertise or little knowledge about the underlying software. Whilst in the case of logical access attacks, the attacker needs the skills to hack into the system as well as knowledge of the underlying software process. In that respect, physical access attacks are more likely or practically feasible than logical access attacks. Despite the technical differences, in an abstract sense we treat physical access attacks and logical access attacks as presentation attacks, as both are related to presentation of falsified or altered signals as input to the speaker verification system.

There are three prominent methods through which these attacks can be carried out, namely, (a) recording and replaying the target speaker speech, (b) synthesizing speech that carries the target speaker characteristics, and (c) applying voice conversion methods to convert impostor speech into the target speaker speech. Among these three, replay attack is the most viable attack, as the attacker mainly needs a recording and playback device. In the literature, it has been found that speaker verification systems, while immune to “zero-effort” impostor claims and mimicry attacks [Mariéthoz and Bengio, 2005], are vulnerable to such elaborated attacks [Kucur Ergunay et al., 2015]. The vulnerability could arise due to the fact that speaker verification systems are inherently built to handle undesirable variabilities. The attack samples can exhibit variabilities that speaker verification systems are robust to and thus, can pass undetected. As a consequence, developing countermeasures to detect presentation attacks is of paramount interest, and is constantly gaining interest in the speech community [Wu et al., 2015a].

2.2.2 Countermeasures

Countermeasures are implemented by training a binary classification system to detect presentation attacks, as illustrated in Figure 2.7. In this section, we provide a brief overview about state-of-the-art systems. For a more comprehensive survey, please refer to [Wu et al., 2015a, 2017].

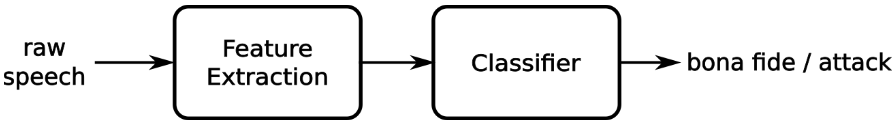


Figure 2.7 – Presentation attack detection system.

Developping countermeasures against presentation attacks is a relatively recent field of research and has been strongly guided by challenges. In particular the ASVspoof 2015 challenge [Wu et al., 2015b], which focused on logical access speech synthesis and voice conversion attacks, and the ASVspoof 2017 challenge [Kinnunen et al., 2017], which focused on replay attacks.

Features

We first focus on the features developed for the detection of speech synthesis and voice conversion as the research community has largely focused on these two types of attacks, driven by the ASVspoof 2015 challenge.

In the literature, different feature representations based on short-term spectrum have been proposed for synthetic speech detection. These features can be grouped as follows:

1. magnitude spectrum based features with temporal derivatives [De Leon et al., 2012a, Wu et al., 2012]: this includes standard cepstral features (e.g., mel frequency cepstral coefficients, perceptual linear prediction cepstral coefficients, linear prediction cepstral coefficients), spectral flux-based features that represent changes in power spectrum on frame-to-frame basis, sub-band spectral centroid based features, and shifted delta coefficients. Constant Q cepstral coefficients (CQCC) [Todisco et al., 2016] have led to significant improvement of the systems.

2. phase spectrum based features [De Leon et al., 2012a, Wu et al., 2013]: this includes group delay-based features, cosine-phase function, and relative phase shift.

3. spectral-temporal features: this includes modulation spectrum [Wu et al., 2013], frequency domain linear prediction [Sahidullah et al., 2015], extraction of local binary patterns in the cepstral domain [Alegre et al., 2013a,b], and spectrogram based features [Gałka et al., 2015].

The magnitude spectrum-based features and phase spectrum-based features have been investigated individually as well as in combination [Patel and Patil, 2015, Alam et al., 2015, Wang et al., 2015, Liu et al., 2015]. All the aforementioned features are based on short-term processing. However, some features such as modulation spectrum or frequency domain linear prediction tend to model phonetic structure-related long-term information.

In addition to these spectral-based features, features based on pitch frequency patterns have been proposed [De Leon et al., 2012b, Ogihara et al., 2005]. There are also methods that aim to extract “pop-noise” related information that is indicative of the breathing effect inherent in normal human speech [Shiota et al., 2015].

Intuitively, the information needed for the detection of voice conversion and speech synthesis attacks is different from the one needed for the detection of replay attacks. In the first case, the systems might focus on artefacts created by the voice conversion and speech synthesis algorithms such as phase mismatch. In the second case, the systems might focus more on channel response and voice quality. Before the BTAS 2016 challenge [Korshunov et al., 2016] and the ASVspoof 2017 challenge, only a few works investigated replay attacks. This detection was mainly based on characteristics related to channel noise and reverberation [Villalba and Lleida, 2011, Wang et al., 2011]. In the ASVspoof 2017 challenge, most submitted systems relied on a mixture of similar features as the ones used for speech synthesis and voice conversion attacks. In particular features based on cepstral coefficients such as LFCC [Lavrentyeva et al., 2017], MFCCs and CQCCs [Ji et al., 2017].

Classifiers

Choosing a reliable classifier is especially important given the possibly unpredictable nature of attacks in a practical system, since it is unknown what kind of attack the perpetrator may use when spoofing the verification system. Different classification methods have been investigated in conjunction with the above described features such as logistic regression, support vector machine (SVM) [Sahidullah et al., 2015, Alegre et al., 2013a], neural networks [Chen et al., 2015, Xiao et al., 2015], and Gaussian mixture models (GMMs) [Sahidullah et al., 2015, De Leon et al., 2012a, Wu et al., 2013, Patel and Patil, 2015, Alam et al., 2015, Wang et al., 2015, Liu et al., 2015]. The choice of classifier is also dictated by factors like dimensionality of features and characteristics of features. For example, in [Sahidullah et al., 2015], GMMs were able to model sufficiently well the de-correlated spectral-based features of dimension 20-60 and yielded highly competitive systems. Whilst in [Tian et al., 2016], neural networks were used to model large dimensional heterogeneous features.

The classifiers are trained in a supervised manner, i.e., the training data is labeled in terms of genuine accesses and attacks. The classifier outputs frame level scores, which are then combined to make a final decision. For instance, in the case of GMM-based classifier, one GMM is trained for the bona fide class and one for the attack class. The log-likelihood ratio between these two models is computed, similarly to a Gaussian Mixture Model-Universal Background Model (GMM-UBM) speaker verification system, and is then compared to a preset threshold to make the final decision.

Leveraging on recent findings in machine learning, deep neural networks are also employed to learn automatically the features using intermediate representations as input such as log-scale spectrograms [Zhang et al., 2017] or filterbanks [Chen et al., 2015, Villalba et al., 2015, Qian et al., 2016]. Some works also employ end-to-end approaches [Dinkel et al., 2017, Lavrentyeva et al., 2017].

2.2.3 Evaluation

Presentation attack detection is a binary task, as is speaker verification. Thus, their evaluation is similar.

Databases for presentation attack detection are also usually split into 3 subsets with non-overlapping speakers: *training*, *development* and *evaluation* subsets. The ISO standard [ISO/IEC JTC 1/SC 37 Biometrics, 2016b] defines two metrics for presentation attack detection: the attack presentation classification error rate (APCER), which is equivalent to false acceptance rate, and the bona fide presentation classification error rate (BPCER), which is equivalent to false rejection rate. However these notations are not widely adopted in the speaker recognition community, instead the evaluation metrics used are either HTER or EER.

2.3 Vulnerability analysis

A vulnerability analysis can be applied on a stand-alone speaker verification system or on one fused with a presentation attack detection system. As illustrated in Figure 2.8, three types of samples need to be considered:

- The genuine samples, which are bona fide samples pronounced by the true speaker.
- The zero-effort impostor samples, which are bona fide samples pronounced by an impostor.
- The presentation attacks.

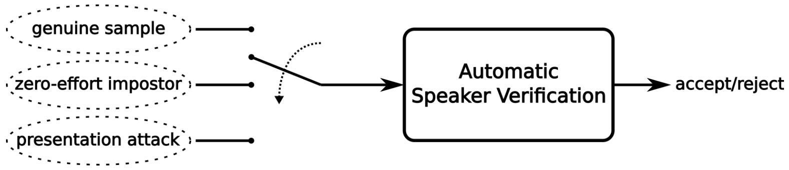


Figure 2.8 – Different types of input to speaker verification system under attack. The system should accept genuine accesses and reject impostors and presentation attacks.

The speaker verification system can be evaluated in two scenarios:

- The licit scenario, where there is no presentation attacks and only genuine and zero-effort impostor samples are considered.
- The spoof scenario, where there are only genuine samples and presentation attacks.

Following this, we can consider three types of errors on the evaluation set:

- the false non match rate (FNMR), which corresponds to the number of genuine samples rejected and is the same in the licit and spoof protocol;
- the false match rate (FMR), which corresponds to the number of zero-effort impostors accepted in the licit scenario;
- the impostor attack presentation match rate (IAPMR), which corresponds to the number of presentation attacks accepted in the spoof scenario.

Another method to evaluate the vulnerability of the system is the expected performance and spoofability curve (EPSC) [Chingovska et al., 2014]. This approach enables to take into account both zero-effort impostors and presentation attacks when choosing the operating threshold. To do so, it defines a parameter $\omega \in [0, 1]$ that weights the FMR and the IAPMR. $\omega = 0$ corresponds to the licit scenario, i.e., there is no attack, and $\omega = 1$ correspond to the spoof scenario, i.e., there is no zero-effort impostor.

$$\mathrm{FAR}_\omega = \omega\ \mathrm{IAPMR} + (1 - \omega)\ \mathrm{FMR} \tag{2.7}$$

The threshold $\tau_{\omega,\beta}$ is then fixed on the development set as:

$$\tau^*_{\omega,\beta} = \arg\min_\tau \left| \beta\ \mathrm{FAR}_\omega(\tau) - (1 - \beta)\ \mathrm{FNMR}(\tau) \right| \tag{2.8}$$

The parameter β enables to weight the positive samples and the negative samples (zero-effort impostors and presentation attacks) depending on the type of application. In our evaluation, we will set $\beta = 0.5$. Once the threshold $\tau_{\omega,\beta}$ is fixed, we can then compute metrics on the evaluation set. One such metric is the weighted error rate (WER):

$$\mathrm{WER}_{\omega,\beta}(\tau^*_{\omega,\beta}) = \beta\,\mathrm{FAR}_{\omega}(\tau^*_{\omega,\beta}) + (1-\beta)\,\mathrm{FNMR}(\tau^*_{\omega,\beta}) \tag{2.9}$$

2.4 Databases

This section provides a description of the databases used in the present thesis.

2.4.1 Speaker recognition

Voxforge

Voxforge is an open source speech database,[1] where different speakers have voluntarily contributed speech data for development of open resource speech recognition systems. Our main reason for choosing the Voxforge database was that most of the corpora for speaker verification have been designed from the perspective of addressing issues like channel variation, session variation and noise robustness. On the other hand we can expect the Voxforge database to have low variability as the text is read and the data is likely to be collected in a clean and consistent environment as each individual records his own speech.

From this database, we selected 300 speakers who have recorded at least 20 utterances. We split this data into three subsets, each containing 100 speakers[2]: the training, the development and the evaluation set. The 100 speakers with the largest number of recorded utterances are in the training set, while the remaining 200 were randomly split between the development and evaluation sets. The statistics for each set is presented in Table 2.1.

Table 2.1 – Number of speakers and utterances for each set of the Voxforge database: training, development, evaluation.

	train	dev		eval	
		enrollment	probe	enrollment	probe
number of utterances/speaker	60-298	10	10-50	10	10-50
number of speakers	100	100		100	

VoxCeleb

VoxCeleb [Nagrani et al., 2017] is a large-scale speaker recognition database. It contains more than 100 000 utterances from 1251 speakers. The audio samples are automatically obtained from videos on YouTube of interviews of celebrities. The data is challenging as the recording conditions are not controlled: the interviews can for example be recorded outdoor, in a quiet studio or with a very large audience. Thus, there can be a high amount of noise such as

[1] http://www.voxforge.org/

[2] The files in each subsets are listed in https://gitlab.idiap.ch/biometric/CNN-speaker-verification-icassp-2018

applause, chatter, laughter and outdoor noise. The celebrities have different ethnicities and accents and the genders are relatively balanced (690 male and 561 female speakers). Each speaker has on average 116 utterances. The utterances last from 4 seconds to 145 seconds with an average of 8.2 seconds.

The data is split into two subsets with non-overlapping speakers: the training set, which contains 1211 speakers, and the evaluation set, which contains 40 speakers. The evaluation set is not split into enrollment and probing subsets. Instead, during test time the system is provided with a list of pairs of utterances and needs to decide whether the two utterances in each pair are from the same speaker or not.

2.4.2 Vulnerability analysis and presentation attack detection

ASVspoof 2015 The ASVspoof 2015[3] database contains genuine and spoofed samples from 45 male and 61 female speakers. This database contains only speech synthesis and voice conversion attacks produced via logical access, i.e., they are directly injected in the system. The attacks in this database were generated with 10 different speech synthesis and voice conversion algorithms. Only 5 types of attacks are in the training and development set (S1 to S5), while 10 types are in the evaluation set (S1 to S10). This allows to evaluate the systems on known and unknown attacks. The full description of the database and the evaluation protocol are given in [Wu et al., 2015c]. This database was used for the ASVspoof 2015 Challenge and is a good basis for system comparison as several systems have already been tested on it. The statistics of the database are presented in Table 2.2.

Table 2.2 – Number of speakers and utterances for each set of the ASVspoof 2015 database: training, development and evaluation.

data set	speakers		utterances	
	male	female	genuine	LA attacks
train	10	15	3750	12625
development	15	20	3497	49875
evaluation	20	26	9404	184000

AVspoof The AVspoof database[4] contains replay attacks, as well as speech synthesis and voice conversion attacks both produced via logical and physical access.

This database contains the recordings of 31 male and 13 female participants divided into four sessions. Each session is recorded in different environments and different setups. For each session, there are three types of speech:

- Reading: pre-defined sentences read by the participants,

[3] http://dx.doi.org/10.7488/ds/298

[4] https://www.idiap.ch/dataset/avspoof

- Pass-phrase: short prompts,
- Free speech: the participants talk freely for 3 to 10 minutes.

In the physical access attacks scenario, the attacks are played with four different loudspeakers: the loudspeakers of the laptop used for the ASV system, external high-quality loudspeakers, the loudspeakers of a Samsung Galaxy S4 and the loudspeakers of an iPhone 3GS. For the replay attacks, the original samples are recorded with: the microphone of the ASV system, a good-quality microphone AT2020USB+, the microphone of a Samsung Galaxy S4 and the microphone of an iPhone 3GS. The use of diverse devices for physical access attacks enables the database to be more realistic. This database is a subset of the one used for the BTAS challenge [Korshunov et al., 2016]. The training and development sets are the same while some additional attacks were recorded for the BTAS challenge in order to have "unknown" attacks in the evaluation set. Here, the types of attacks are the same in the three sets. The statistics of the database are presented in Table 2.3.

Table 2.3 – Number of speakers and utterances for each set of the AVspoof database: training, development and evaluation.

data set	speakers		utterances		
	male	female	genuine	PA attacks	LA attacks
train	10	4	4973	38580	17890
development	10	4	4995	38580	17890
evaluation	11	5	5576	43320	20060

3 Raw waveform-based CNNs for speaker verification

State-of-the-art speaker recognition systems are conventionally based on modeling short-term spectral features, as discussed in Section 2.1. In recent years, with the advances in deep learning, novel approaches have emerged where speaker verification systems are trained in an end-to-end manner [Variani et al., 2014, Heigold et al., 2016, Zhang et al., 2017]. These neural network-based systems take as input either filterbanks outputs [Variani et al., 2014, Heigold et al., 2016] or spectrograms [Zhang et al., 2017, Nagrani et al., 2017]. In this chapter, we aim to go a step further and train such systems directly on raw waveforms. Our motivation is the following. Speaker differences occur at both voice source level and vocal tract system level [Wolf, 1972, Sambur, 1975]. However, speaker recognition research has focused to a large extent on modeling features such as cepstral features and filter bank energies, which carry information mainly related to the vocal tract system, with considerable success. Modeling raw speech signal instead of short-term spectral features enables to make minimal assumptions about the speech signals. Employing little or no prior knowledge could potentially provide alternate features or means for speaker discrimination. Furthermore, in recent works, it has been shown that raw speech signal can be directly modeled to yield competitive systems for speech recognition [Palaz et al., 2013, Tüske et al., 2014, Sainath et al., 2015], emotion recognition [Trigeorgis et al., 2016], voice activity detection [Zazo et al., 2016] and gender classification [Kabil et al., 2018] to name a few.

Motivated by these works, this chapter aims to answer two research questions:

1. Can we achieve state of the art performance by learning speaker discriminative information directly from raw waveforms using neural networks?

2. If yes, what kind of information do the neural networks learn? Do they model source or vocal tract system-related information? Are the extracted information complementary to the information modeled by systems based on short-term spectral features?

This chapter will first present our approach to learn directly speaker discrimination from raw waveforms. Next, the proposed approach is validated through investigations on two datasets. Finally, we analyze the neural networks to gain insight about the information learned.

3.1 Proposed raw speech modeling-based approach

The proposed approach consists in training neural networks directly on raw waveforms instead of using short-term spectral features such as cepstral coefficients or filter-banks outputs. We use convolutional neural networks (CNNs), as done in [Palaz, 2016]. This type of network is well suited to deal with raw waveforms as the weight sharing and pooling operations enables temporal shift invariance. We propose two approaches based on CNNs trained on raw waveforms. The first scheme uses the CNN to extract speaker discriminative embeddings that we refer to as *r-vectors* (*r* stands for "raw"), as done for example in [Variani et al., 2014, Nagrani et al., 2017]. In the second scheme, speaker specific detectors are developped in an end-to-end manner.

CNN training for speaker identification: In both verification schemes, the first step consists in training a CNN as a speaker identifier, as illustrated in Figure 3.3. This CNN takes raw waveforms as input and is trained to classify *n* speakers, which are different from the ones that will later be enrolled in the speaker verification system. This step is akin to the UBM step in standard speaker verification approaches, except that here a speaker discriminative model is trained instead of a generative one.

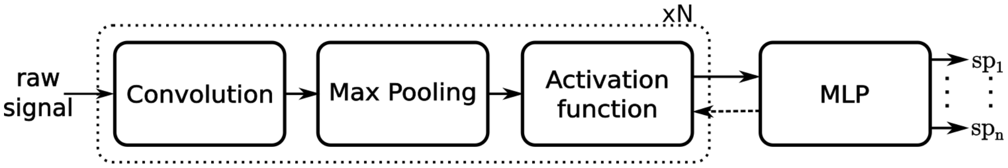


Figure 3.1 – Diagram of a CNN trained for speaker identification.

The CNN consists of: *N* convolution layers followed by a multilayer perceptron (MLP), also referred to as fully connected layers in the literature. Each convolution layer is composed of 3 operations: convolution, max-pooling and a non-linear activation function. This architecture was first proposed in the context of speech recognition [Palaz et al., 2013, Palaz, 2016, Palaz et al., 2019] and has later been used successfully for other tasks such as gender recognition [Kabil et al., 2018], depression detection [Dubagunta et al., 2019] and paralinguistic speech processing [Vlasenko et al., 2018].

Each utterance is split into overlapping sequences of length w_{seq} and shifted by 10 ms. Each sequence is normalized to have zero mean and unit variance and is then fed to the CNN independently. Figure 3.2 illustrates the first convolution layer processing of the raw input.

Besides w_{seq}, the system based on the proposed approach has the following hyper parameters: (i) number of convolution layers N, (ii) for each convolution layer $i \in \{1, \cdots N\}$, kernel width kW_i, kernel shift dW_i, number of filters n_{fi} and max-pooling size mp_i and (iii) number of hidden layers and hidden units in the MLP. These hyperparameters are determined with a coarse grid search based on the validation error. In doing so, the system also automatically de-

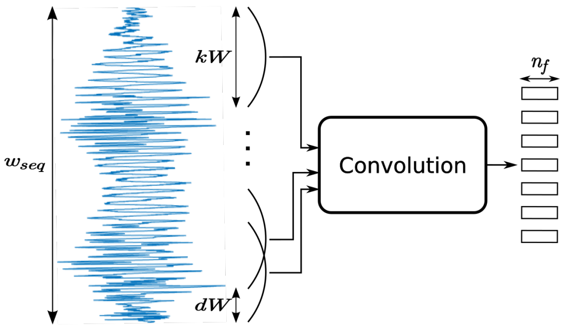


Figure 3.2 – Illustration of the first convolution layer processing.

termines the short-term processing applied on the speech signal to learn speaker information. More precisely, the first convolution layer kernel width kW_1 and kernel shift dW_1 are the frame size and frame shift that operate on the signal. The first convolution either processes the raw waveform in a sub-segmental manner, i.e., with kW_1 below 1 pitch period or in a segmental manner, i.e., with kW_1 corresponding to 1-4 pitch periods. The latter case corresponds to the conventional short-term processing of speech signals.

Embeddings extraction: Once the speaker identification CNN is trained, the first approach consists in extracting embeddings called *r-vectors*, which are expected to be speaker discriminative and robust to variabilities such as recording conditions. Each sequence of length w_{seq} is forwarded into the CNN and the output of the penultimate layer, i.e. the hidden layer of the MLP, is extracted. An utterance-level embedding is obtained by averaging the frame-level embeddings of the utterance. Similarly, during enrollment a speaker-level embedding is obtained by averaging the utterance-level embeddings if the enrollment data is composed of several utterances. The embeddings are then classified with the same methods as the ones used with *i-vectors*: they are first projected in a lower dimensional space with a linear discriminant analysis (LDA) and subsequently classified with a probabilistic linear discriminant analysis (PLDA). This is a common approach and is used for example in [Kenny, 2010, Snyder et al., 2018], as explained in Section 2.1.

Speaker specific adaptation: The proposed end-to-end speaker specific adaptation scheme in illustrated in Figure 3.4. For each speaker s_m, $m = 1, \ldots, M$ that needs to be enrolled in the speaker verification system, the CNN-based speaker identification system is converted into a speaker specific detector by: (a) replacing the output layer by two classes (genuine, impostor) and randomly initializing the weights between the output layer and the MLP hidden layer; and (b) adapting the CNN in a discriminative manner with the target speaker enrollment data and impostor speech data from speakers contained in the set used to train the CNN-based speaker identification system, i.e., the set containing speakers that will never be enrolled in the system.

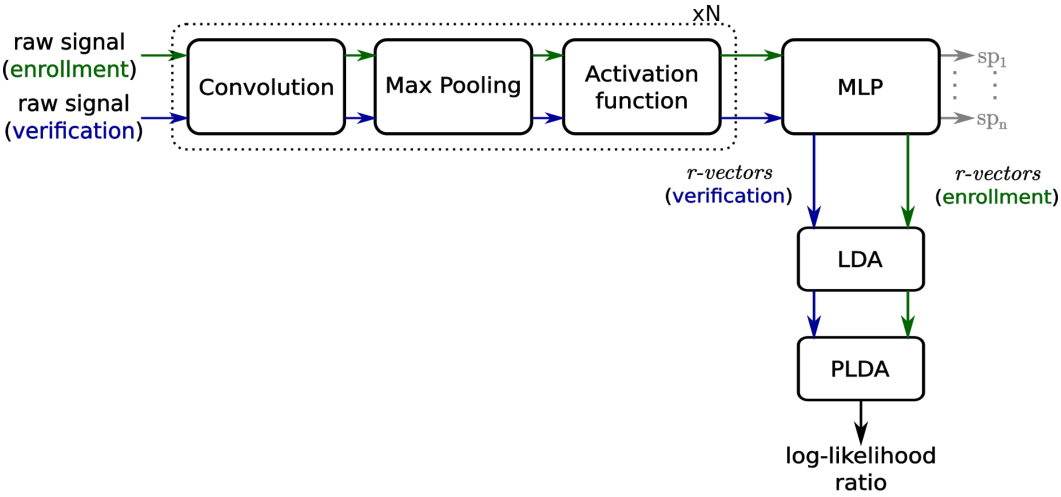


Figure 3.3 – Illustration of the *r-vectors*-based approach.

In the verification phase, the test speech is passed through the binary speaker classifier of the claimed speaker and the decision is made by averaging the output log posterior probability for genuine class and impostor class over time frames. The decision is taken by thresholding the average log probability.

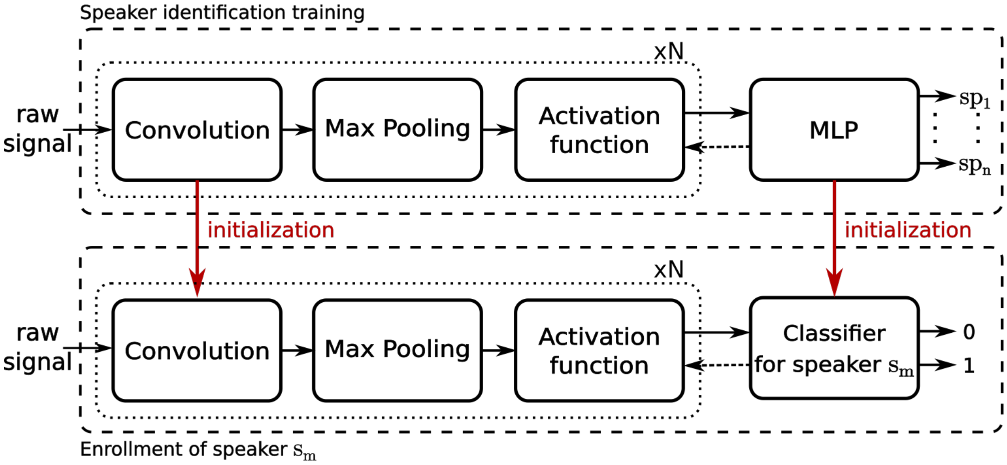


Figure 3.4 – Illustration of the proposed end-to-end speaker specific adaptation scheme.

3.2 Investigations on clean conditions

In this section, we analyze the performance of the proposed approaches on a relatively clean database, before moving to a more challenging one in the next section. We first describe the database and the evaluation protocol. We then describe the baseline systems used in the experiments and provide the implementation details of the proposed systems. Finally, we present the results.

3.2.1 Experimental protocol

The experiments are conducted on the Voxforge database, described in details in Section 2.4.1. It is an open source speech database,[1] where different speakers have voluntarily contributed speech data for development of open resource speech recognition systems. Our main reason for choosing the Voxforge database was that most of the corpora for speaker verification have been designed from the perspective of addressing issues like channel variation, session variation and noise robustness. As a first step, our aim was to see whether the proposed approach could learn speaker discriminative information directly from the speech signal. We can expect the Voxforge database to have low variability as the text is read and the data is likely to be collected in a clean environment as each individual records his own speech. However, the database consists of short utterances of about 5 seconds length recorded by speakers over the time. From this database, we selected 300 speakers who have recorded at least 20 utterances. We split this data into three subsets, each containing 100 speakers: the training, the development and the evaluation set. The statistics of the subsets are presented in Section 2.4.1.

The training set is used by the baseline systems to obtain a UBM. Whilst, it is used to obtain a speaker identification system in the proposed approach. The development and evaluation sets are split into enrollment and probe data. The enrollment data is used to train each speaker's model and contains 10 utterances per speaker. The probe part of the development data is used to fix the score threshold to achieve an equal error rate (EER), while the half total error rate (HTER) is computed on the probe data of the evaluation set based on this threshold, as explained in Section 2.1.4.

3.2.2 Systems

Baseline systems

We trained two baseline systems on the Voxforge database with Kaldi using recipes originally developed for the VoxCeleb database [2] and corresponding to the systems used in [Snyder et al., 2018].

The first baseline system is based on *i-vectors* [Dehak et al., 2011], which corresponds to a UBM-GMM based system as explained in Section 2.1 and still yields state-of-the-art performance in many cases. The input features are 20 dimensional MFCC with delta and double delta, which yield vectors of dimension 60. The UBM is a 2048 component full-covariance GMM and the *i-vectors* have a dimension of 600.

The second system is based on *x-vectors* [Snyder et al., 2018], which are 512-dimensional embeddings obtained by taking the output of the third-last layer of a TDNN trained to classify

[1] http://www.voxforge.org/

[2] https://github.com/kaldi-asr/kaldi/tree/master/egs/sre16/v2

speakers. This TDNN takes as input 30 dimensional MFCCs and is composed of 5 fully connected layers, followed by a global statistical pooling layer that aggregates frame-level output to obtain an utterance-level output and followed by 3 fully connected layers.

In both systems, the embeddings are projected into a lower dimensional space of dimension 200 with a LDA and then classified with a PLDA, both trained with the embeddings (*i-vectors* or *x-vectors*) obtained from the training set of the Voxforge database.

Proposed systems

An energy-based [Bimbot et al., 2004, Magrin-Chagnolleau et al., 2001] voice activity detection (VAD) is first applied on the sequences of raw waveforms that will be fed to the CNN. Frame-level energy values are computed, normalized and then classified into two classes: voice and silence. If the 10 ms middle frame of the sequence of length w_{seq} is classified as silent we discard it. Otherwise, each sequence is normalized to have zero mean and unit variance and is fed to the CNN.

In the first step, the CNN-based speaker identification system was trained on the training data by splitting it into a training part (90%) and a validation part (10%) for early stopping. Following the work in [Palaz et al., 2013], the activation function used is a hard hyperbolic tangent function, i.e.,

$$f(x) = \begin{cases} 1, & \text{if } x > 1 \\ -1, & \text{if } x < -1 \\ x, & \text{otherwise.} \end{cases}$$

The proposed system has several hyperparameters. These hyperparameters were determined through a coarse grid search and based on validation accuracy. The best validation accuracy was obtained for an architecture with two convolution layers and one hidden layer in the MLP and an input sequence of length $w_{seq} = 510$ ms. The architecture of the CNN is presented in Table 3.1. Stochastic gradient descent based training with early stopping was performed with a cost function based on cross entropy using Torch software [Collobert et al., 2011].

Using $kW_1 = 300$ samples yields the lowest validation error rate. As we will see in Section 3.4 and as published in [Muckenhirn et al., 2018], when using a long kernel kW_1 in the first layer, the CNN tends to focus mainly on source-related information such as fundamental frequency. On the other hand, CNN-based speech recognition studies [Palaz et al., 2019] have shown that when using a short kernel $kW_1 = 30$, i.e. processing the waveform in a sub-segmental manner, the CNN is able to learn formant information. In an attempt to make the CNN focus on vocal tract system-related information instead of voice source, we also train a CNN with exactly the same architecture, except that we set $kW_1 = 30$.

Table 3.1 – Architecture of the CNNs trained on Voxforge. n_f denotes the number of filters in the convolution layer. n_{hu} denotes the number of hidden units in the hidden layer. kW denotes kernel width. dW denotes kernel shift (stride). Mpool refers to max pooling.

Layer	kW	dW	n_f	n_{hu}
Conv1	30 or 300	10	80	-
Mpool + HardTanh	5	5	-	-
Conv2	10	1	80	-
Mpool + HardTanh	5	5	-	-
MLP + HardTanh	-	-	-	100
MLP + Softmax	-	-	-	100

Embedding-based system: The embeddings, referred to as *r-vectors*, correspond to the output of the hidden layer of the MLP and thus have a dimension of 100. Utterance-level *r-vectors* are simply computed by averaging the *r-vectors* obtained for each sequence of length w_{seq} across the utterance. In the same manner as it is done with the baseline systems, the *r-vectors* are then projected in a lower-dimensional space of dimension 70 (this dimension yielded the lowest error rate on the development set) with a LDA and classified with a PLDA, trained on the training set, i.e., the same set used to train the CNN in the first place. This system is referred to as "*r-vectors* $kW_1 = 30$" and "*r-vectors* $kW_1 = 300$" depending on which kernel size the CNN was trained with.

End-to-end speaker specific adaptation system: For the adaptation of the CNN, the enrollment data of each speaker was split into a training part (80%) and a validation part (20%). The impostor examples were the same for all speakers and were obtained by randomly selecting 300 utterances from the training set, which was used to build the speaker identification system. This system is referred to as "end-to-end $kW_1 = 30$" and "end-to-end $kW_1 = 300$" depending on which kernel size the CNN was trained with.

3.2.3 Results

Table 3.2 presents the HTER obtained with the baseline systems and the proposed CNN-based systems on the evaluation set of the Voxforge database. Among the two baseline systems, *i-vectors* clearly outperforms *x-vectors*. Our embedding based *r-vectors* systems outperform *x-vectors* but yields a higher HTER than *i-vectors*. On the other hand, we see that using the proposed end-to-end speaker specific adaptation scheme significantly improves the performance and yields the lowest HTER, outperforming *i-vectors*.

To test the complementarity of using a short and long kernel in the first convolution layer of the proposed approaches, we average the scores of the systems. We see that in both cases, the fusion lowers the HTER of the systems, indicating that the long and short kernel systems are indeed focusing on different information.

The last 8 lines of the table corresponds to score-level fusion of the different embedding-based systems, which consists in a simple score average. We observe that, while none of the fusion schemes outperform our end-to-end approaches, the embeddings are complementary. In particular, the best performance is yielded by the fusion of *i-vectors*-based and *r-vectors*-based systems. This suggests that the proposed *r-vectors* capture different information from *i-vectors*. It is also interesting to note that the two baseline systems *i-vectors* and *x-vectors* are complementary. While they both use similar MFCC features, the first system is based on generative training while the latter is based on discriminative training. This difference could explain their complementarity.

Table 3.2 – Performance of the baseline and proposed systems on the evaluation set of Voxforge.

System	**HTER (%)**
i-vectors	2.21
x-vectors	5.16
r-vectors $kW_1 = 30$	4.08
r-vectors $kW_1 = 300$	4.10
r-vectors $kW_1 = 30$ +*r-vectors* $kW_1 = 300$	3.57
end-to-end $kW_1 = 30$	1.15
end-to-end $kW_1 = 300$	0.80
end-to-end $kW_1 = 30$ + end-to-end $kW_1 = 300$	**0.75**
i-vectors + *x-vectors*	1.92
i-vectors + *r-vectors* $kW_1 = 30$	1.46
i-vectors + *r-vectors* $kW_1 = 300$	1.35
i-vectors + *r-vectors* $kW_1 = 30$ + *r-vectors* $kW_1 = 300$	1.85
x-vectors + *r-vectors* $kW_1 = 30$	2.81
x-vectors + *r-vectors* $kW_1 = 300$	2.65
x-vectors + *r-vectors* $kW_1 = 30$ + *r-vectors* $kW_1 = 300$	2.93
i-vectors + *x-vectors* + *r-vectors* $kW_1 = 30$ + *r-vectors* $kW_1 = 300$	1.61

3.3 Investigations on challenging conditions

In order to validate the proposed approach on challenging conditions, we conducted investigations on the VoxCeleb database. First the experimental protocol is described, i.e., the database used as well as its evaluation protocols. Then the systems are detailed, in particular the architecture modification of the CNNs to improve the performance in challenging scenarios. Finally the results are presented.

3.3.1 Experimental protocol

We perform the experiments on the VoxCeleb database, detailed in Section 2.4.1, which contains $\approx$ 100000 utterances from 1251 speakers. This database was created by downloading videos of interviews of famous people, which were then automatically segmented and split into utterances that contain only the voice of the person of interest. Since there is no control about the recording conditions, they can be quite different from one video to another, e.g., quality of the microphones and background noise. The speaker recognition task is thus challenging. This database can be used for both speaker identification and speaker verification. Here we focus on the verification task.

In contrast to the Voxforge database, this database is split in two subsets instead of three: the training set (1211 speakers) and the evaluation set (40 speakers), i.e. there is no development set. As before, the training set is either used to train a UBM in the GMM-UBM based systems such as *i-vectors* or to train a neural network such as a DNN in the case of *x-vectors* and a CNN in our proposed approach. Another difference with the Voxforge database is that the test set is not split into enrollment and probing subsets as it is usually the case in biometric applications. Instead, during test time the system is provided with a list of pairs of utterances and needs to output whether the utterances in each pair are from the same speaker or not.

This protocol is well suited for embedding-based systems but not for the end-to-end approach for two reasons. First, one utterance might not be sufficient to adapt the parameters of the CNN. Second, we need to adapt the CNN to each target utterance, which is computationally intensive. There are 40 speakers in the evaluation set and 4975 target utterances in this protocol, which means that we need to adapt 4975 CNNs instead of 40. This end-to-end adaptation approach is proposed in the context of authentication applications, where several utterances are used for enrollment of a speaker. We thus created a new protocol where we use 3 utterances for enrollment. More precisely, out of all the videos of each speaker we randomly select three of them. We then take the longest utterance of each video and discard all the others to ensure that the utterances used for enrollment are not from the same videos as the ones used for probing. We then use the utterances extracted from the remaining videos for probing.

We follow the evaluation metics used in the original paper [Nagrani et al., 2017]: EER and the minimum of the detection cost function (minDCF) with $P_{target} = 0.01$, described in Section 2.1.4.

3.3.2 Systems

Baseline systems

As done on the Voxforge database, we train two baseline systems on VoxCeleb using the Kaldi toolkit: the first one is based on *i-vectors* and the second on *x-vectors.* These systems are the same as the one used on Voxforge and are described in details in Section 3.2.2.

Proposed systems

We initiated the development of the proposed system based on the architecture used on Voxforge. However, this architecture did not generalize well on VoxCeleb. This is probably due to the fact that it does not capture well high variabilities. We thus took cues from speaker embedding learning, specifically:

1. we increased the length of the input sequences w_{seq},
2. we added more convolution layers,
3. we added a global statistical pooling layer before the MLP stage, as done in [Snyder et al., 2018]. The global statistics pooling layer computes the mean and standard deviation along each filter. If there are n_f filters in the previous convolution layer, then this yields a mean vector and a standard deviation vectors of size n_f, which are concatenated. The output dimension does not depend on the size of the inputs and enables the CNN to deal with inputs of variable sizes,
4. each convolution layer is followed by a batch normalization layer and the activation function is a Rectified Linear Unit (ReLU) function instead of a hard hyperbolic tangent function.

The architecture is described in Table 3.3. The value of the hyperparameters were obtained through a coarse grid search based on the validation error. The CNN contains 6 convolution layers instead of 2 and has more filters in each convolution layer as well as more hidden units in the fully connected layer. w_{seq} is now equal to 2.41 seconds instead of 510 ms.

Furthermore, the raw waveforms are now pre-emphasized with a coefficient of 0.97 before being fed to the CNN. We also modified the training scheme of the CNN. The training is done with mini-batches (batch size of 20 samples) instead of stochastic gradient descent and the Adam optimizer is used.

As explain in Section 3.1, we investigate two different schemes to perform speaker verification: extracting *r-vectors* and the end-to-end speaker specific adaptation. *x-vectors*, which correspond to the output of the first fully connected layer before the ReLU layer, now have a dimension of 512 instead of 100. They are then reduced to a dimension of 200 with a LDA and classified with a PLDA, following what is done for the two baseline systems. In the second case, a CNN is adapted for each speaker. 100 impostor utterances taken randomly from the training set are used to adapt the CNN in the original protocol and 300 impostor utterances in the modified protocol.

While the training is done with fixed size inputs of 2.41 seconds, during the forward passes the CNN takes the whole utterances as inputs. We have seen that it improved the performance compared to splitting each utterance into several frames of 2.41 seconds and averaging the outputs, as it was done on the Voxforge database.

Table 3.3 – Architecture of the CNNs trained on VoxCeleb. n_f denotes the number of filters in the convolution layer. n_{hu} denotes the number of hidden units in the hidden layer. kW denotes kernel width. dW denotes kernel shift (stride). Mpool refers to max pooling and Spool refer to statistics pooling.

Layer	kW	dW	n_f	n_{hu}
Conv1	30 or 300	5	100	-
Mpool + ReLU	3	3	-	-
Conv2	10	1	300	-
Mpool + ReLU	3	3	-	-
Conv3	3	1	300	-
Mpool + ReLU	3	3	-	-
Conv4	3	1	512	-
Mpool + ReLU	5	1	-	-
Conv5	3	1	512	-
Mpool + ReLU	5	1	-	-
Conv6	1	1	1000	-
ReLU + Spool	-	-	-	-
MLP + ReLU	-	-	-	512
MLP + Softmax	-	-	-	1211

3.3.3 Results

In this section, we present the results on the two evaluation protocols: the original protocol as well as our modified version with a fixed enrollment subset for each speaker.

Original protocol

Table 3.4 presents the performance of several systems on the original protocol of the VoxCeleb database using EER and minDCF. *i-vectors* and *r-vectors* correspond to the two baseline systems. The "reported" systems correspond to the best results reported in the literature. "*r-vectors* $kW_1 = 30$" and "*r-vectors* $kW_1 = 300$" correspond to the proposed system where embedding, a.k.a. *r-vectors*, are extracted. "end-to-end $kW_1 = 30$" and "end-to-end $kW_1 = 300$" correspond to the proposed end-to-end speaker specific adaptation approach.

We first observe that among the stand-alone embedding-based systems (*i-vectors*, *x-vectors* and proposed *r-vectors*), the *i-vectors* system performs the best and thus outperforms the neural network-based approaches. Secondly, the *x-vectors* yield a significantly higher error rate than our proposed *r-vectors*. Third, *r-vectors* with $kW_1 = 30$ and $kW_1 = 300$ are complementary and a simple score-level fusion outperforms significantly the *i-vectors* based systems. This reaffirms the observation made on Voxforge and the hypothesis that short and long kernel CNNs focus on different information. This is analyzed further in Section 3.4.

The end-to-end speaker specific adaptation approaches perform poorly compared to the

Table 3.4 – Performance of the baseline and proposed systems on the evaluation set of VoxCeleb.

	Systems	Input	Scoring	EER	minDCF
Baseline	*i-vectors*	MFCC	LDA-PLDA	5.42	0.51
	x-vectors	MFCC	LDA-PLDA	7.29	0.61
Reported	2D CNN (VGG) + contrastive loss [Nagrani et al., 2017]	Spectrogram	Siamese NN	7.8	0.71
	2D CNN (VGG) [Nagrani et al., 2017]	Spectrogram	cosine	10.2	0.75
	2D CNN (VGG) + center loss [Yadav and Rai, 2018]	Spectrogram	cosine	4.9	-
	ResNet [Wang et al., 2019]	FBank	cosine	4.86	0.51
	1D residual-CNN-LSTM [Jung et al., 2018]	Raw	NN	7.4	-
Proposed	*r-vectors* kW_1*=30*	Raw	LDA-PLDA	5.83	0.53
	r-vectors kW_1*=300*	Raw	LDA-PLDA	5.72	0.52
	r-vectors kW_1*=30* + *r-vectors* kW_1*=300*	Raw	LDA-PLDA	4.99	0.47
	end-to-end kW_1=30	Raw	end-to-end	14.67	0.82
	end-to-end kW_1=300	Raw	end-to-end	14.34	0.80
Fusion	*i-vectors* + *x-vectors*	MFCC	LDA-PLDA	5.15	0.46
	r-vectors kW_1*=30* + *i-vectors*	Raw, MFCC	LDA-PLDA	4.19	0.40
	r-vectors kW_1*=300* + *i-vectors*	Raw, MFCC	LDA-PLDA	4.17	0.42
	r-vectors kW_1*=30* + *r-vectors* kW_1*=300* + *i-vectors*	Raw, MFCC	LDA-PLDA	**4.07**	**0.39**
	r-vectors kW_1*=30* + *x-vectors*	Raw, MFCC	LDA-PLDA	4.90	0.46
	r-vectors kW_1*=300* + *x-vectors*	Raw, MFCC	LDA-PLDA	4.96	0.47
	r-vectors kW_1*=30* + *r-vectors* kW_1*=300* + *x-vectors*	Raw, MFCC	LDA-PLDA	4.50	0.45
	r-vectors kW_1*=30* + *r-vectors* kW_1*=300* + *i-vectors* + *x-vectors*	Raw , MFCC	LDA-PLDA	4.10	0.40

embedding-based ones. As explained in Section 3.3.1, this adaptation scheme was designed with an authentication setup in mind, i.e., with several utterances used for enrolling a speaker. Clearly, one utterance is not sufficient.

The last 8 rows of Table 3.4 correspond to the score-level fusion of the different embedding-based systems: *i-vectors*, *x-vectors* and the proposed *r-vectors*. This fusion is obtained by simply averaging the scores output of each system, which are all log-likelihood computed with a PLDA. We observe that fusing the *i-vectors* system with a *r-vectors* system ($kW_1 = 30$ or $kW_1 = 300$) yields a much higher improvement than fusing it with the *x-vectors* system. More specifically, when compared to the stand-alone *i-vectors* system, fusing it with a *r-vectors* system decreases the EER relatively by 23% while fusing it with the *x-vectors* system decreases the EER by only 5%. The same observation is true for *x-vectors*, i.e. fusing the *x-vectors* system with the proposed *r-vectors* system ($kW_1 = 30$ or $kW_1 = 300$) achieves a higher improvement than fusing it with the *i-vectors* based system, even though taken separately the *i-vectors* based system actually outperforms the proposed *r-vectors* based systems. This suggests that *x-vectors* and *i-vectors* might focus on information that are more similar compared to the *r-vectors*.

Finally, the best performance (EER of 4.07%) is obtained by fusing the *i-vectors* system with the two *r-vectors* systems. Adding the *x-vectors* to this fusion slightly drops the performance (EER of 4.10%).

Modified protocol

The results are presented in Table 3.5. The EER and minDCF of all systems are lower, which is expected since we have more data to derive the embeddings. However, all the observations made on the original protocol remain the same for the embedding-based systems as well as for the different fused systems.

The end-to-end systems perform clearly better than on the original protocol, demonstrating that this approach indeed needs more data. However, it does not outperform the *r-vectors* based systems, contrary to what we observed on Voxforge in Section 3.2.3. One explanation is that the end-to-end system does not generalize as well as the embeddings to different recording conditions. This was not an issue on Voxforge since the recording conditions had a low variability.

Table 3.5 – Performance of the baseline and proposed systems on the evaluation set of the modified protocol of VoxCeleb.

System	**EER (%)**	**minDCF**
i-vectors	2.24	0.251
x-vectors	3.45	0.382
r-vectors $kW_1 = 30$	2.93	0.288
r-vectors $kW_1 = 300$	3.19	0.327
r-vectors $kW_1 = 30$ + *r-vectors* $kW_1 = 300$	2.70	0.275
end-to-end $kW_1 = 30$	4.81	0.545
end-to-end $kW_1 = 300$	5.41	0.536
end-to-end $kW_1 = 30$ + end-to-end $kW_1 = 300$	4.50	0.504
i-vectors + *x-vectors*	2.24	0.268
i-vectors + *r-vectors* $kW_1 = 30$	**1.82**	**0.198**
i-vectors + *r-vectors* $kW_1 = 300$	1.96	0.217
i-vectors + *r-vectors* $kW_1 = 30$ + *r-vectors* $kW_1 = 300$	1.96	0.217
x-vectors + *r-vectors* $kW_1 = 30$	2.32	0.255
x-vectors + *r-vectors* $kW_1 = 300$	2.47	0.275
x-vectors + *r-vectors* $kW_1 = 30$ + *r-vectors* $kW_1 = 300$	2.27	0.247
i-vectors + *x-vectors* + *r-vectors* $kW_1 = 30$ + *r-vectors* $kW_1 = 300$	1.87	0.218

3.4 Analysis

In the previous sections we showed that raw waveform modeling with CNNs yield systems competitive to short-term spectral processing-based approaches. A question that arises is: what information is modeled by the CNNs? We focus on the analysis of the first convolution layer with two methods. We will first explain these methods. We will then use them to analyze the CNNs trained on the Voxforge database and the CNNs trained on the VoxCeleb database.

3.4.1 Methods

To analyze what information is modelled by the first convolution layer we use two methods.

The first method consists in computing the cumulative frequency response of the learned filters, similarly to [Palaz et al., 2019, 2015]:

$$F_{cum} = \sum_{k=1}^{n_{f1}} \frac{\mathscr{F}_k}{\|\mathscr{F}_k\|_2}, \tag{3.1}$$

where n_{f1} is the number of filters in the first convolution layer and $\mathscr{F}_k$ is the magnitude spectrum of filter f_k, $k = 1, \ldots, n_{f1}$, computed with a 512-point Discrete Fourier Transform (DFT). The cumulative frequency response indicates which frequency regions the filters, when averaged, focus on.

The second method focuses on how the filters respond to an input speech. In the work on speech recognition [Palaz et al., 2019], which formed the basis for the present work, it was found that the filters can be interpreted as a spectral dictionary, [3] and the magnitude frequency response S_t of the input signal $s_t = \left\{s_t^1, \cdots s_t^{kW_1}\right\}$ can be estimated, as

$$S_t = \left| \sum_{k=1}^{n_{f1}} \langle s_t, f_k \rangle \mathrm{DFT}\{f_k\} \right|, \tag{3.2}$$

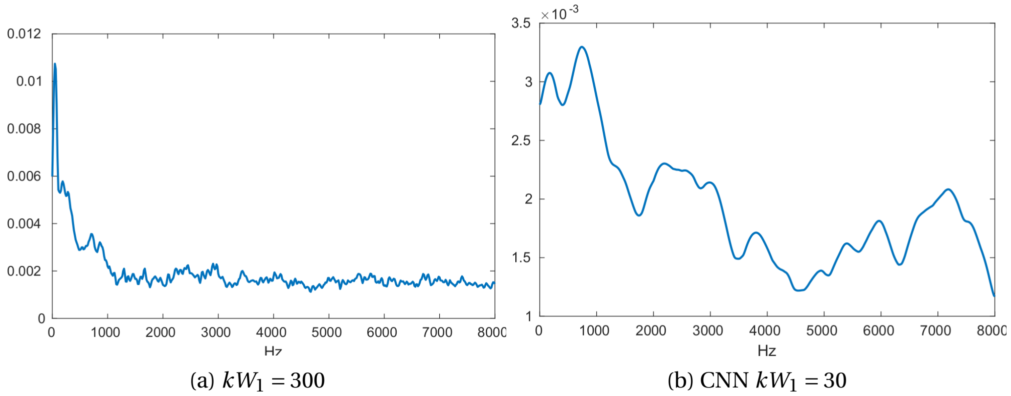
and analyzed to understand the discriminative information that is being modeled. If the atoms of the dictionary, i.e., f_k, were to correspond to Fourier sines and cosines and $kW_1 = n_{f1}$, then S_t would simply be the Fourier magnitude spectrum of s_t. In regular case, the dictionary is usually overcomplete and the inner product $\langle s_t, f_k \rangle$ represents the weights (which are usually sparse) corresponding to the spectral contribution of atoms/filters.

3.4.2 Cumulative frequency response

Voxforge database

In Figure 3.5, we show the cumulative frequency response of the filters of the first convolution layer of the CNNs trained on the Voxforge database, with a kernel size kW_1 of 30 and 300 samples. When $kW_1 = 300$ the filters give emphasis to the information lying below 1000 Hz. On the other hand, when $kW_1 = 30$, the filters focus on different frequency regions, with the highest peaks in the very low frequency region (below 500 Hz) and between $\approx 800 - 1000$ Hz. This indicates that the speaker discriminative information learned by the two systems are different.

[3] It is worth mentioning that such interpretations of CNN filters have also been put forward in the signal processing community [Papyan et al., 2017, Mallat, 2016].

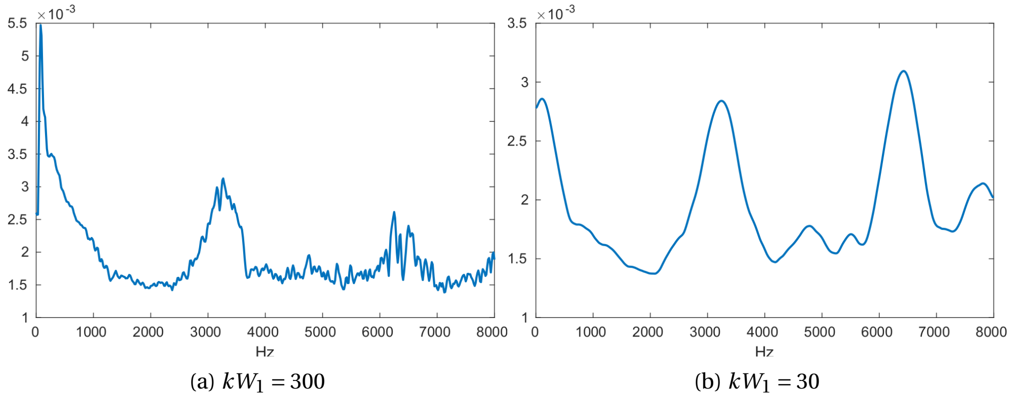


(a) $kW_1 = 300$ (b) CNN $kW_1 = 30$

Figure 3.5 – Cumulative frequency responses of first layer filters, trained on the Voxforge database.

VoxCeleb database

In Figure 3.6, we show the cumulative frequency response of the filters of the first convolution layer of the CNNs trained on the VoxCeleb database, with a kernel size kW_1 of 30 and 300 samples. When $kW_1 = 300$ the response in the low frequency is similar to what we observe on the CNN trained on the Voxforge database, however there are also two smaller peaks in the higher frequency regions: one between $\approx 2500 - 3500$ Hz and a smaller one between $6000 - 7000$ Hz. The cumulative frequency response of the system with $kW_1 = 30$ is similar to the one with $kW_1 = 300$ with a lower resolution, except that the spectral balance is different.

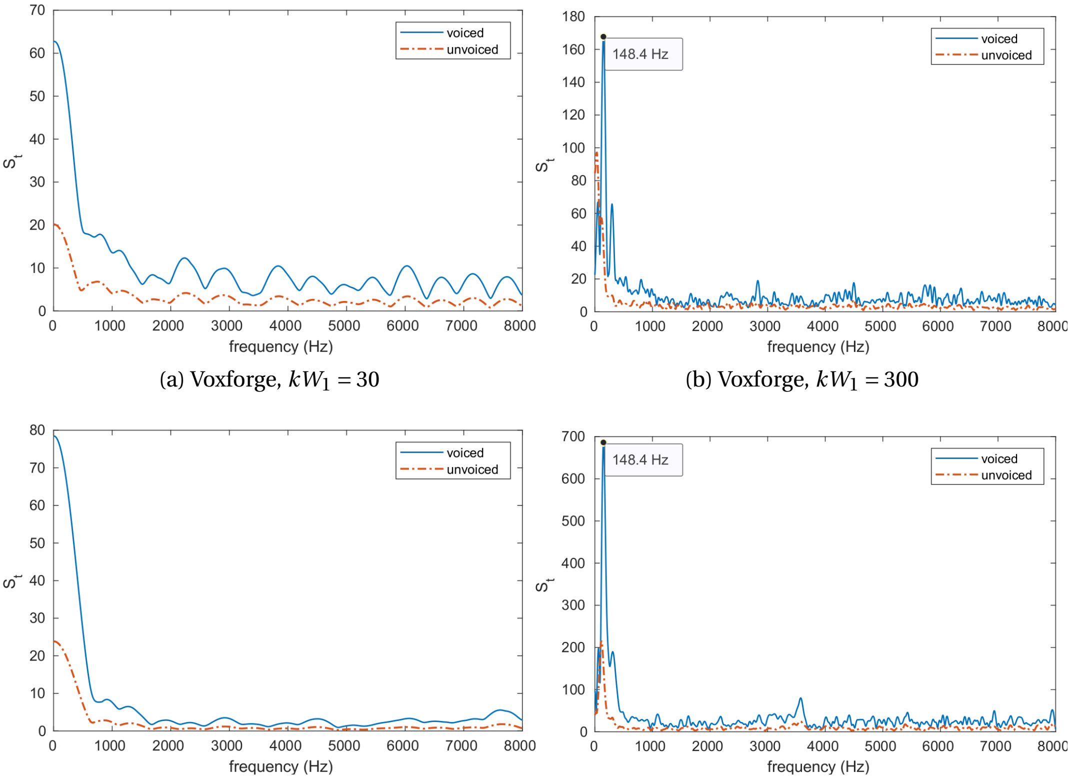


(a) $kW_1 = 300$ (b) $kW_1 = 30$

Figure 3.6 – Cumulative frequency responses of first layer filters, trained on the VoxCeleb database.

3.4.3 Frequency response to input: fundamental frequency analysis

In the previous section, we observed that in all cases the filters are giving emphasis to low frequency information. One of the speaker-specific information that lies below 500 Hz is fundamental frequency. Considering this point we performed an analysis of voiced speech and unvoiced speech of a few male and female speakers by computing the magnitude frequency response S_t. An example of such analysis on a voiced and unvoiced frame is shown in Figure 3.7 for a male speaker and in Figure 3.8 for a female speaker. In the case of voiced speech, we see a distinctive peak occurring near the fundamental frequency (F_0) with both systems using a long kernel, i.e. when $kW_1 = 300$, while no such distinctive peak appears for unvoiced speech. This suggests that the first convolution layer is learning F_0 modeling. In the case of the CNNs with $kW_1 = 30$ the interpretation is more difficult due to the very low resolution. We observe that in both voiced and unvoiced case there is a peak in the low frequency below 500 Hz, which reaches its maximum at 0 Hz and that this peak is higher in case of voiced speech than unvoiced. This peak suggests that the systems focus on voice source related information as well.



(a) Voxforge, $kW_1 = 30$

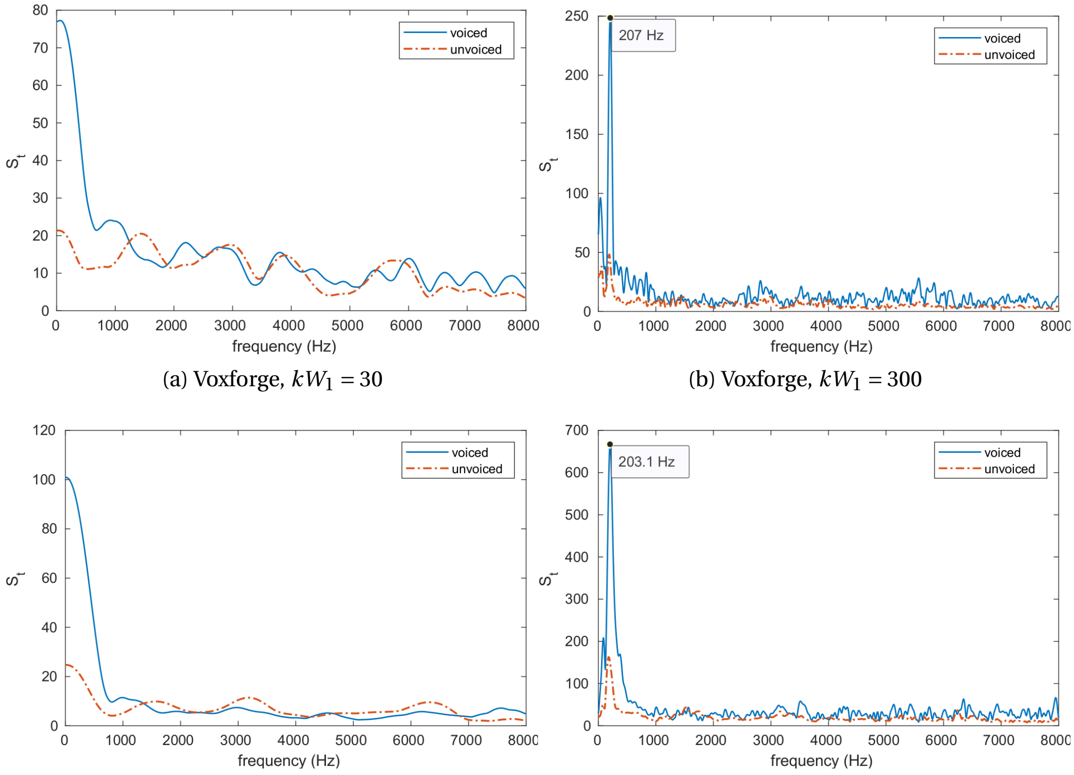
(b) Voxforge, $kW_1 = 300$

(c) VoxCeleb, $kW_1 = 30$

(d) VoxCeleb, $kW_1 = 300$

Figure 3.7 – Magnitude frequency response S_t of the first layer convolution filters on several systems given a voiced and unvoiced frame of a male speaker. $F_0 = 149$ Hz for the voiced frame input, estimated using wavesurfer [Sjölander and Beskow, 2000].

Now that we have observed that when $kW_1 = 300$ the main peak of the magnitude frequency response S_t seems to correspond to the fundamental frequency, we conduct a quantitative experiment to ascertain that. We implement a simple F_0 estimator based on the observations made in the previous section and evaluate it on the Keele Pitch database [Plante et al.,



(a) Voxforge, $kW_1 = 30$

(b) Voxforge, $kW_1 = 300$

(c) VoxCeleb, $kW_1 = 30$

(d) VoxCeleb, $kW_1 = 300$

Figure 3.8 – Magnitude frequency response S_t of the first layer convolution filters on several systems given a voiced and unvoiced frame of a female speaker. $F_0 = 206$ Hz for the voiced frame input, estimated using wavesurfer [Sjölander and Beskow, 2000].

1995]. This database contains the speech and laryngograph signals for 5 male and 5 female speakers reading a phonetically balanced text as well as hand-corrected F_0 estimates from the laryngograph signals. The steps involved in the F_0 estimation are as follows:

- For each frame of input signal of length $kW_1 = 300$ samples, the frequency response S_t is estimated using Eqn. (3.2).
- The DFT bin with the maximum energy in the frequency range 70 Hz - 400 Hz is selected.
- The peak energy is thresholded to decide if the frame is voiced or unvoiced. If it is voiced then the frequency corresponding to the DFT bin is the F_0 estimate.
- A median filter is applied on the estimated F_0 contour in order to smooth it.

The speech was down-sampled from 20 kHz to 16 kHz to match the sampling frequency of the Voxforge and VoxCeleb databases. The frame shift was set to 10ms, as done in the Keele database for determining the reference F_0 from the laryngograph signal. The number of points for DFT was set as 4096 points. The energy threshold to decide voiced/unvoiced and the size of the median filter were determined on the female speaker `f1n0000w` speech, such that low voiced/unvoiced (V/UV) error and gross error (i.e. deviation of estimated F_0 is within 20% of reference F_0 or not) is obtained. This threshold and the median filter size (=7) was used when estimating F_0 contours of the remaining nine speakers data and for evaluating the F_0 estimator. Figure 3.9 shows the F_0 contours for the first phrase spoken by a female and a male speaker estimated with the CNN trained on Voxforge with $kW_1 = 300$. It can be observed that the estimated F_0 contours are reasonably close to the reference F_0 contours.

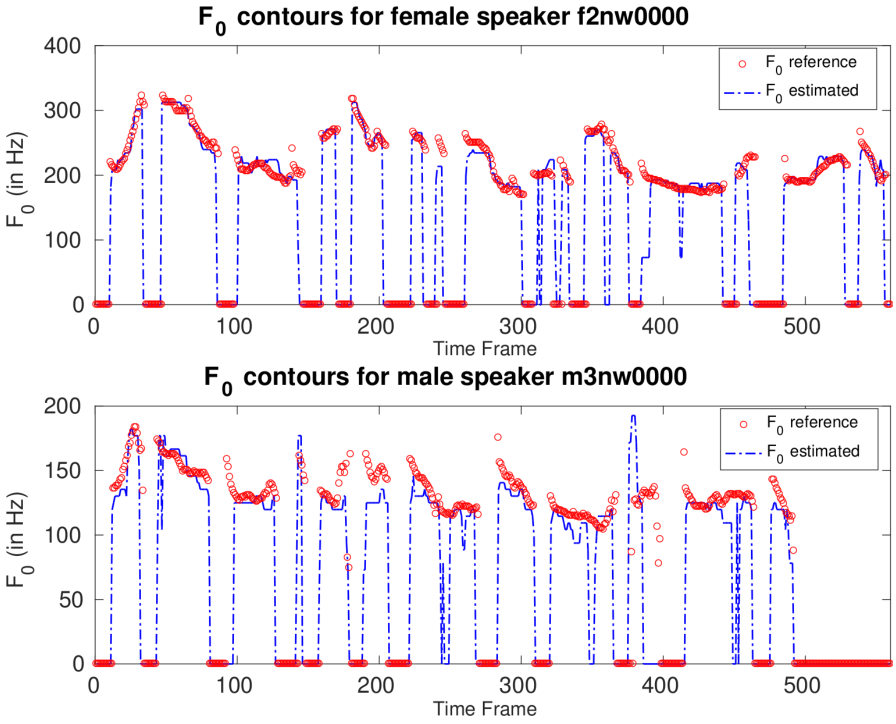


Figure 3.9 – Two examples of the F_0 contours estimated using the first layer filters compared to the reference F_0 from the Keele Pitch database. The CNN corresponds to the system trained on Voxforge with $kW_1 = 300$.

Table 3.6 presents the results of the evaluation. As it can be seen, the performance of this simple F_0 estimator is clearly beyond chance-level performance. The estimation for females are better than for males. The reason for this could be that frames of $kW_1 = 300$ samples, which is about to 19ms, do not contain enough pitch cycles for very low F_0. We have indeed observed that through informal analysis of the errors.

Table 3.6 – F_0 estimation evaluation on the Keele database

	V/UV error (%)		**Gross error (%)**	
	female	male	female	male
Voxforge, $kW_1 = 300$	16.1	22.3	3.6	24.0
VoxCeleb, $kW_1 = 300$	14.2	34.4	2.13	13.9

3.4.4 Frequency response to input: formant analysis

In speech recognition studies [Palaz et al., 2019], it was found that the convolution filters of the first layer model formant information. We observe a similar trend when looking at the cumulative frequency response in Section 3.4.2, more on the CNN trained on Voxforge than on VoxCeleb. In particular, the CNN trained on Voxforge with $kW_1 = 30$ has its highest peak in the range 500-1500 Hz, which could correspond to the first formant information. To validate this hypothesis, we performed an analysis on American English Vowel database [Hillenbrand et al., 1995]. This database contains recordings of 12 vowels uttered by 45 men, 48 women and 46 children with fixed context.

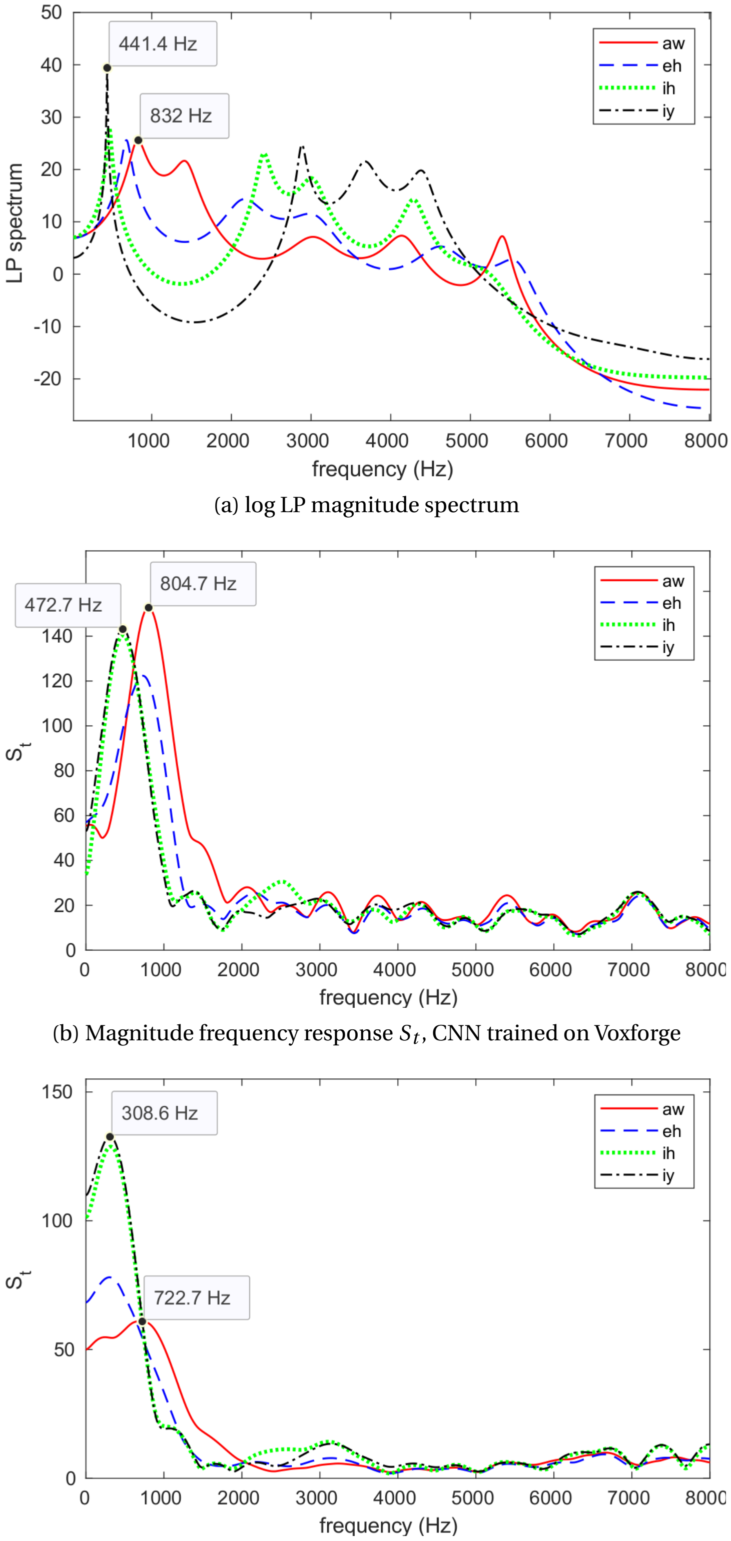
Figure 3.10a shows the linear prediction (LP) spectrum estimated for a frame of 30 ms speech signal for /aw/, /eh/, /ih/ and /iy/ produced by female speaker `w02`. The order of linear prediction is 20. In Figure 3.10b and 3.10c, we show the corresponding magnitude frequency response estimated based on Eqn. (3.2) for exactly the same 30 ms frames with the CNN trained respectively on Voxforge and on VoxCeleb database with $kW_1 = 30$. This is done by computing S_t after every dW_1 samples ($dW_1 = 10$ samples on Voxforge and $dW_1 = 5$ samples on VoxCeleb) in the 30 ms speech signal and averaging it [Palaz et al., 2019]. We observe that, as hypothesized, in the case of the CNN trained on Voxforge, the main peak seems to correspond to the first formant while it is not the case for the one trained on VoxCeleb.

Figure 3.11a shows the exact same linear prediction (LP) spectrum as in Figure 3.10a except that we show only the sounds /aw/ and /iy/ for clarity reasons. In Figure 3.11b and 3.11c, we show the corresponding magnitude frequency response estimated based on Eqn. (3.2) for the same 30 ms frames with the CNN trained respectively on Voxforge and on VoxCeleb database with $kW_1 = 300$. In both cases that there seems to be a peak corresponding to the first formant information. This indicates that the first convolution layer does not focus only on fundamental frequency but also on first formant information.

In order to ascertain that these observations generalize to other samples, we conducted a quantitative study on this database by comparing the first spectral peak locations with first formant location information provided with the American vowel database[4] in the following manner:

- The location (frequency) of the first peak of the LP magnitude spectrum is selected.
- The first spectral peak location is extracted from the magnitude frequency response S_t of the CNN-based system.
- We consider that the first formant location is correctly estimated if it is in the range $F_1 \pm (1+\Delta)$, where F_1 is the value of the first formant.

[4] https://homepages.wmich.edu/ hillenbr/voweldata.html

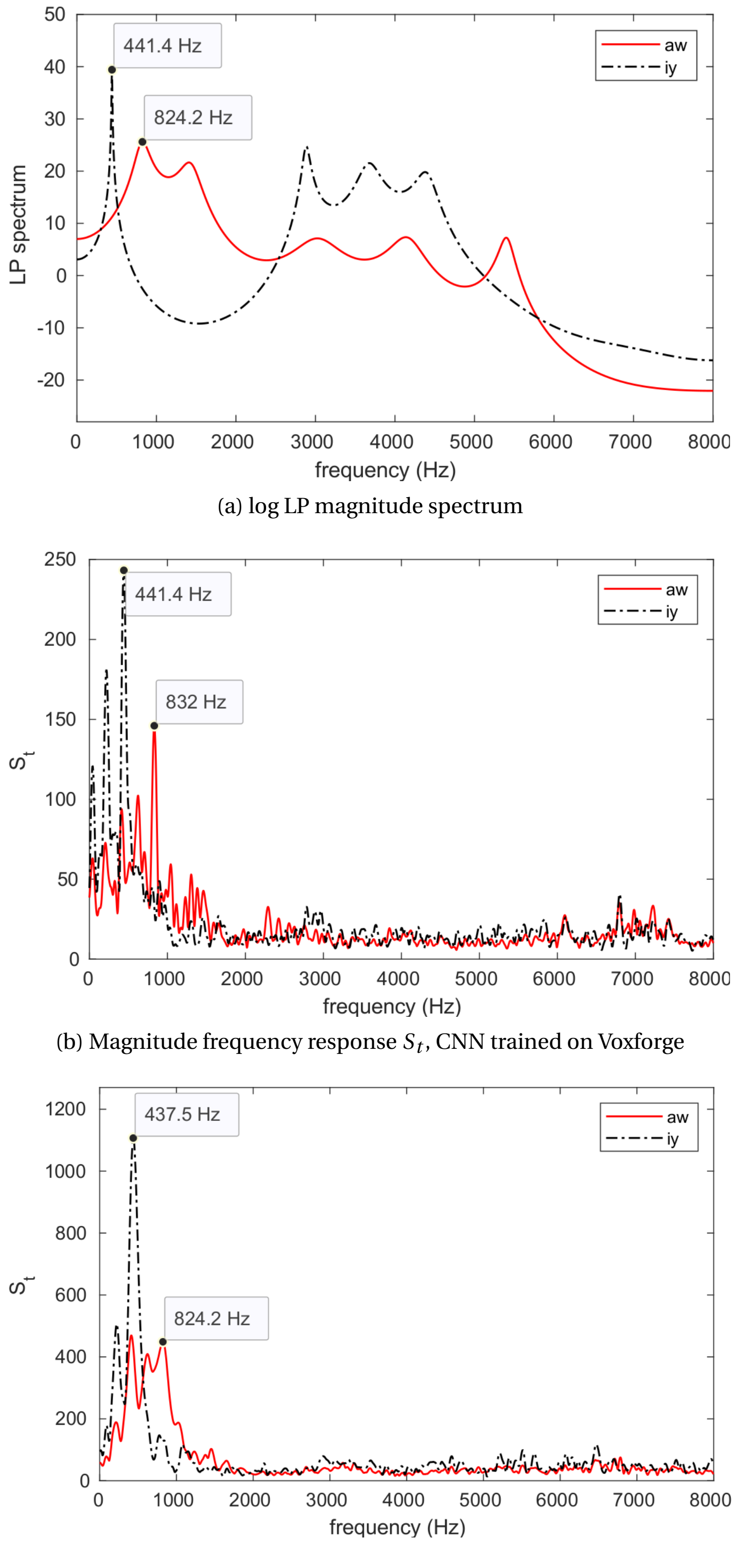


(a) log LP magnitude spectrum

(b) Magnitude frequency response S_t, CNN trained on Voxforge

(c) Magnitude frequency response S_t, CNN trained on VoxCeleb

Figure 3.10 – Analysis of different vowels spoken by female speaker w02. CNNs with $kW_1 = 30$.



(a) log LP magnitude spectrum

(b) Magnitude frequency response S_t, CNN trained on Voxforge

(c) Magnitude frequency response S_t, CNN trained on VoxCeleb

Figure 3.11 – Analysis of different vowels spoken by female speaker w02. CNNs with $kW_1 = 300$.

We varied the Δ and computed accuracy over the whole database composed of 1668 utterances. Table 3.7 presents the estimation accuracy for different values of Δ with the four systems. As hypothesized previously, we observe that the estimation with the CNN trained on VoxCeleb with $kW_1 = 30$ yields a very poor results. However, the formant information estimation with the three other systems yields a relatively high score: for respectively 83%, 75% and 70% of the samples the main peak of S_t is in the range $[0.85F_1, 1.15F_1]$. On the other hand, the first peak of the LP spectrum is in the range $[0.85F_1, 1.15F_1]$ for 97% of the samples. This indicates that in those three cases the CNN is focusing on speaker discriminative information present in the formant regions but is less precise about the speech sound when compared to the LP spectrum.

Table 3.7 – Accuracy of first formant estimation in range $[F_1(1-\Delta), F_1(1+\Delta)]$.

Δ	0.1	0.15	0.2
First peak of LP spectrum	0.93	0.97	0.98
First peak of CNN $kW_1 = 30$, Voxforge	0.55	0.83	0.93
First peak of CNN $kW_1 = 300$, Voxforge	0.66	0.75	0.82
First peak of CNN $kW_1 = 30$, VoxCeleb	0.16	0.23	0.31
First peak of CNN $kW_1 = 300$, VoxCeleb	0.62	0.70	0.77

3.5 Summary

This chapter investigated two methods to build speaker verification systems modeling directly raw waveforms with CNNs: an approach based on an end-to-end speaker specific adaptation and an approach based on the extraction of embeddings, called *r-vectors*. We demonstrated that modeling raw waveforms with minimal assumptions is possible and yields systems that perform comparably or better than state-of-the-art systems. By analyzing the first layer of the CNNs it was found that such systems capture both voice source-related information, such as fundamental frequency, and vocal tract-related information, such as formants. Furthermore, it was found on two corpora that sub-segmental and segmental raw waveform-based CNNs are complementary. It was also found that the proposed *r-vectors* are complementary to the short-term spectral features-based *i-vectors* and *x-vectors*. In particular, the score-level fusion of *r-vectors*-based and *i-vectors*-based systems yields, to the best of our knowledge, the best reported performance on the VoxCeleb database.

4 Trustworthy speaker verification

In the previous chapter, we proposed a new approach for speaker verification, based on CNNs trained on raw waveforms. We showed that this approach yields competitive performance compared to state-of-the-art systems, which rely on short term spectral features. In this chapter, we focus on the robustness of these speaker recognition systems to presentation attacks.

As discussed in Section 2.2, it is now well known that speaker recognition systems are vulnerable to presentation attacks, i.e., audio samples that are altered or forged by an attacker to try to be successfully authenticated as someone else. Three types of presentation attacks exist: replay, speech synthesis and voice conversion. As illustrated in Figure 4.1 and explained in Section 2.2, attacks can be performed at two points: physical access attacks are presented to the microphone (point 1 in Figure 4.1) while logical access attacks are injected into the speaker verification system without being recorded by the microphone (point 2 in Figure 4.1). This chapter investigates both types of attacks.

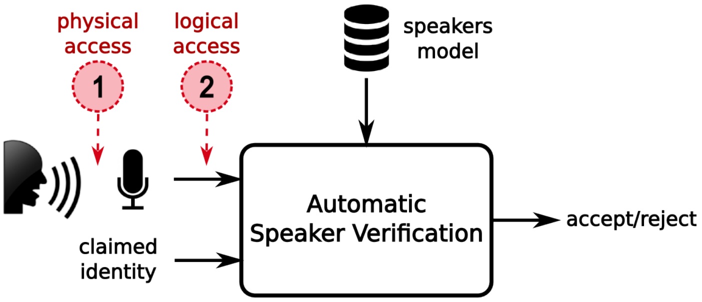


Figure 4.1 – Physical access (point 1) and logical access (point 2) presentation attacks on a speaker verification system.

In this chapter, we aim to answer three research questions:

1. How vulnerable are the proposed CNN-based speaker verification systems to presentation attacks compared to state-of-the-art systems?

2. How to detect presentation attacks?
3. Does the addition of presentation attack detection systems make the speaker verification systems robust to such attacks while not degrading too much the verification performance?

We will first analyze the vulnerability of the systems presented in the previous chapter to such attacks. After showing on two different databases that all systems are vulnerable to both logical and physical access attacks, we focus on building systems to detect such attacks. To do so, we propose two different approaches that employ minimal prior knowledge: one based on the computation of long-term spectral statistics and one based on CNNs trained on raw waveforms. Finally, we fuse the speaker verification and presentation attack detection (PAD) systems and quantify the impact this fusion has on the robustness as well as on the recognition performance.

4.1 Vulnerability analysis

In this section, we analyze the vulnerability of speaker verification systems to logical and physical access attacks.

4.1.1 Experimental protocol

Databases

We conduct the vulnerability analysis on two databases: the Automatic Speaker Verification Spoofing (ASVspoof) 2015 database and the Audio-Visual Spoofing (AVspoof) database. The ASVspoof 2015 database contains only logical access attacks, while the AVspoof database contains both logical access (LA) and physical access (PA) attacks. A detailed description of the two databases can be found in Section 2.4.2.

Both databases have a speaker verification protocol designed to test vulnerability to presentation attacks. Such a protocol needs three types of samples:

- genuine accesses, i.e., bona fide audio samples from the true speaker;
- zero-effort impostors, i.e., bona fide audio samples from a different speaker;
- presentation attacks.

The vulnerability protocol of the ASVspoof 2015 database was designed such that the zero-effort impostor samples are not contained in the presentation attack detection protocol. We were not able to reproduce some of the baseline systems based on cepstral features [Korshunov and Marcel, 2016] and instead used the scores that were available. As a consequence we had to slightly modify the protocol of the ASVspoof database to remove the extra utterances and made it similar to the AVspoof database, where the zero-effort impostors are simply the genuine samples spoken by other speakers than the target speaker.

The training data of the AVspoof and ASVspoof databases is relatively small. Thus, in addition to the bona fide samples of the training sets of these two databases, we use the VoxCeleb database, described in Section 2.4.1, to train the speaker verification systems.

Systems

The speaker verification systems are the ones presented in our speaker recognition study on the VoxCeleb database in the previous chapter. Details can be found in Section 3.3. There are four systems:

- *i-vectors*: extraction of *i-vectors* [Dehak et al., 2011] from MFCC features (20 first coefficients appended with delta and delta-delta features), followed by LDA and by PLDA.
- *x-vectors*: extraction of *x-vectors* [Snyder et al., 2018] obtained with a TDNN trained on 30 dimensional MFCC features, followed by LDA and by PLDA.
- *r-vectors*: extraction of *r-vectors* from a CNN trained on raw waveform, followed by LDA and by PLDA. The first layer of the CNN use a kernel width of either 30 samples (referred to as *r-vectors* $kW_1 = 30$) or 300 samples (referred to as *r-vectors* $kW_1 = 300$).
- end-to-end: speaker-specific adaptation of a CNN trained on raw waveforms. As in the previous case, the first layer of the CNN use a kernel width of either 30 samples (referred to as “end-to-end $kW_1 = 30$”) or 300 samples (referred to as “end-to-end $kW_1 = 300$”).

Unlike the VoxCeleb database, where the speech signals are well segmented, AVspoof and ASVspoof 2015 databases contain silences. So, we first perform a VAD using an energy-based algorithm [Bimbot et al., 2004, Magrin-Chagnolleau et al., 2001]: energies are computed over frames of 20ms with an overlap of 10ms, normalized and then classified into two classes: speech and silence. VAD is an important step in particular for the AVspoof database as some utterances contain long silences, including samples in the enrollment data.

All the aforementionned speaker verification systems are trained from scratch. The hyperparameters of the baselines and proposed systems are the same as the one listed in Section 3.3. The only difference is that the training set contains more data than previously ($\approx$ 162K utterances instead of $\approx$ 149K) due to the fact that we use all the data of the VoxCeleb database instead of just the training set and we also use the bona fide training samples of the AVspoof and ASVspoof 2015 databases.

Evaluation

As explained in Section 2.3, two protocols are used to evaluate the vulnerability of a system. The first protocol is the "licit protocol" and contains genuine speakers and zero-effort impostors, i.e., there are no presentation attacks and this corresponds to the conventional way of evaluating speaker verification systems. The second one is the "spoof protocol" and contains genuine speakers and presentation attacks. The threshold is determined on the development set of the licit scenario (i.e., without taking into account the attacks) as to obtain an equal error rate. We then measure the following on the evaluation set:

- the False Non Match Rate (FNMR), which corresponds to the number of genuine samples rejected and is the same in the licit and spoof protocol;
- the False Match Rate (FMR), which corresponds to the number of zero-effort impostors accepted in the licit scenario;
- the Impostor Attack Presentation Match Rate (IAPMR), which corresponds to the number of presentation attacks accepted in the spoof scenario.

Ideally, all these measures should be low. We also evaluate the vulnerability with Expected Performance and Spoofability Curve (EPSC) [Chingovska et al., 2014], presented in details in Section 2.3, which takes into account both zero-effort impostors and presentation attacks to determine the threshold on the development set.

4.1.2 Results

We present the vulnerability results of the speaker verification systems on the ASVspoof database in Table 4.1 and on the AVspoof database in Table 4.2. On both databases we observe that among the baseline systems, *i-vectors* outperform *r-vectors* on both databases in the licit scenario. This is in line with the results obtained on the VoxCeleb database in Section 3.3.

Table 4.1 – Vulnerability analysis on the evaluation set of the ASVspoof database.

Systems	FNMR (%)	FMR (%)	IAPMR(%)
i-vectors	3.16	4.56	45.91
x-vectors	9.02	19.61	39.54
r-vectors $kW_1 = 300$	3.02	3.66	54.12
r-vectors $kW_1 = 30$	3.72	3.08	51.67
end-to-end $kW_1 = 300$	3.42	3.77	58.30
end-to-end $kW_1 = 30$	3.60	2.95	80.94

On the ASVspoof database, all the proposed approaches perform comparably or better than the *i-vectors*-based system in the licit scenario. The end-to-end approach performs comparably to the *r-vectors* based approach. On the other hand, the *x-vectors* based system performs poorly

Table 4.2 – Vulnerability analysis on the evaluation set of the AVspoof database.

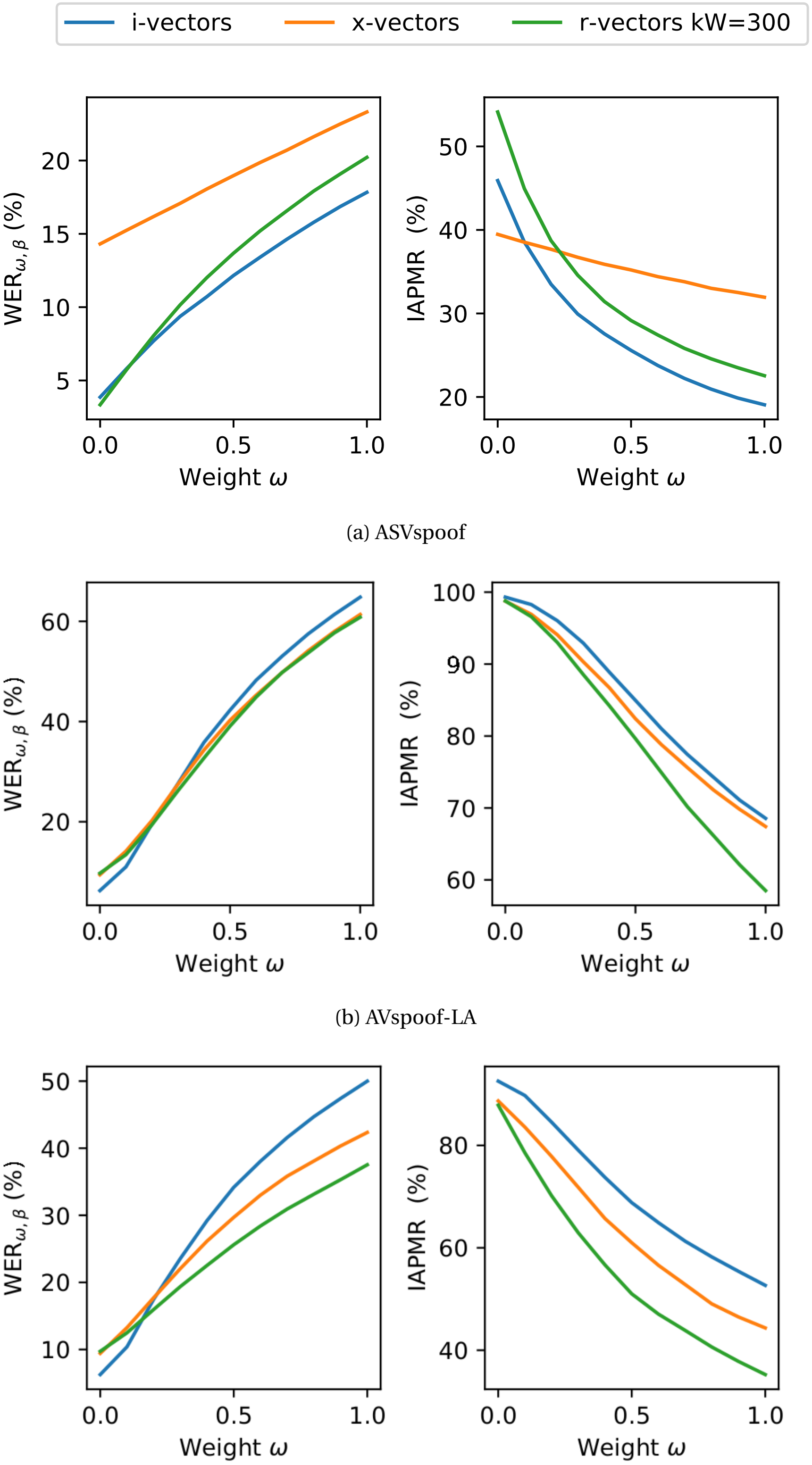
Systems	**FNMR (%)**	**FMR (%)**	**IAPMR(%)**	
			PA	LA
i-vectors	4.61	7.91	92.55	99.31
x-vectors	6.99	11.85	88.68	98.75
r-vectors kW_1=300	5.73	13.73	87.86	98.77
r-vectors kW_1=30	4.18	18.45	91.02	99.01
end-to-end kW_1=300	18.08	7.71	93.33	98.04
end-to-end kW_1=30	17.20	7.93	97.04	96.63

compared to other systems. *x-vectors* are trained on 3 seconds-long segments. One possible explanation for its poor performance could be that some samples in this database are very short. We reduced the length of the training inputs but it did not improve the performance.

On the AVspoof database, our "*r-vectors* $kW_1 = 300$" performs on average comparably to the *x-vectors* system in the licit scenario - the FNMR is slightly lower and the FMR is slightly higher. The end-to-end approaches yield a high FNMR compared to the baseline systems, which is different from what we observe on the ASVspoof database. The main difference between the two databases are the recording conditions. The ASVspoof data was recorded in a hemi-anechoic chamber and hence all samples are very clean. On the other hand, the AVspoof database was recorded on different devices and in different recording environments. In particular, the samples used for enrollment were in addition recorded with a laptop, while the probe samples were recorded with a high-quality microphone and with smartphones. This means that there is a channel mismatch that needs to be taken care of. Thus, one possible explanation for the observed performance is that the *i-vectors*-based system compensates better channel mismatches than the neural network-based approaches (*x-vectors*, *r-vectors* and end-to-end approaches). This is in line with the observations made on the VoxCeleb database in Chapter 3.

We observe on the two databases that the IAPMR is high for all systems, especially on AVspoof. This indicates that all systems are vulnerable to presentation attacks, but are more vulnerable to the attacks in the AVspoof database than on the ASVspoof database. We also see that systems that perform better in the licit scenario, i.e. with lower FNMR and FMR, tend to be more vulnerable to attacks. For example, the *x-vectors*-based system has the lowest IAPMR on ASVspoof but a high FNMR and FMR, while the *r-vectors*-based systems have a high IAPMR but a low FNMR and FMR.

In Figure 4.2, we plot the EPSC of the *i-vectors*, *x-vectors* and *r-vectors* $kW = 300$ systems on ASVspoof, AVspoof-LA and AVspoof-PA. We present separately the Weighted Error rate (WER) and IAPMR. We fixed $\beta = 0.5$, which means that negative samples (zero-effort impostors and presentation attacks) and positive samples (genuine access) have the same weight and thus the WER corresponds to a HTER. We varied the value of ω, which weights the FMR and the IAPMR. $\omega = 0$ corresponds to the licit scenario, i.e., the attacks are not considered. When



(a) ASVspoof

(b) AVspoof-LA

(c) AVspoof-PA

Figure 4.2 – EPSC: weighted error rate (left) and impostor attack presentation match rate (right) on three databases. $\beta = 0.5$

$\omega = 1$, the threshold is chosen on the developement without the zero-effort impostor samples. We observe that when we take into account the presentation attack, i.e. when $\omega > 0$, the proposed *r-vectors* yields a lower WER and IAPMR than the two baseline systems on the AVspoof database and a lower WER and IAPMR than *x-vectors* on the ASVspoof 2015 database.

4.2 Presentation attack detection

In the previous section, we observed that all speaker verification systems, both proposed and state-of-the-art, are vulnerable to presentation attacks. In this section, we focus on the development of systems to detect such attacks that can be then combined with the speaker verification systems.

Most of the countermeasures developed until now have been built on top of standard short-term speech processing techniques that enable a decomposition of the speech signal into source and system, and develop countermeasures focusing on either one of them or both. However, both bona fide accesses and presentation attacks are speech signals that carry the same high level information, such as message, speaker identity and information about environment. Thus, standard speech related assumptions, such as the source filter modeling and the auditory filtering may hold well for both bona fide and forged signals. There is little prior knowledge that can guide us to differentiate bona fide samples from presentation attacks. Hence, a question that arises is: are there alternatives to short-term spectral features-based presentation attack detection? Aiming to answer this question, we develop two novel approaches that make minimum speech signal modeling related assumptions. The first approach simply computes the first and the second order spectral statistics over Fourier magnitude spectrum to detect presentation attacks. The second approach learns to detect presentation attacks from raw waveforms using CNNs.

In the remainder of this section we first describe these two proposed approaches. We then define the experimental protocol and present the results. Finally, we provide further analyses of the proposed approaches.

4.2.1 Long-term spectral statistics-based approach

This section first motivates the use of long-term spectral statistics for presentation attack detection, and then presents the details of the proposed approach.

Motivation

Instead of relying on standard short-term speech processing techniques used in state-of-the-art systems, such as computing cepstral features, we propose an approach where we make minimal assumptions about the signal. We assume that bona fide samples and attacks have different statistical characteristics, irrespective of what is spoken and who has spoken, and we want to use these characteristics to differentiate them. One such statistical property are the first and second-order statistics (i.e., mean and variance) of the energy distributed in the different frequency bins.

first order statistics: Long-term average spectrum (LTAS) is a set of first order spectral statistics that can be estimated either by performing a single Fourier transform of the whole utterance or by averaging the spectrum computed by windowing the speech signal over the utterance [Kinnunen et al., 2006, Löfqvist, 1986]. Originally, the interest in estimating LTAS emerged from the studies on speech transmission [Dunn and White, 1940] and the studies on intelligibility of speech sounds, specifically measurement of articulation index, which represents the proportion of average speech signal that is audible to a human subject [French and Steinberg, 1947]. Later in the literature, LTAS has been extensively used to study voice characteristics [Löfqvist, 1986]. It is employed for example for the early detection of voice pathology [Tanner et al., 2005] or Parkinson disease [Smith and Goberman, 2014], or for evaluating the effect of speech therapy or surgery on the voice quality [Master et al., 2006]. In addition to assessing voice quality, LTAS has also been used to differentiate between speakers gender [Mendoza et al., 1997] and speakers age [Linville and Rens, 2001], to study singers and actors voices [Leino, 1993, Sundberg, 1999] and also to perform speaker verification [Kinnunen et al., 2006]. First order statistics are interesting for developing countermeasures for presentation attacks as natural speech and synthetic speech differ in terms of both intelligibility and quality. In particular,when computing the LTAS, the short-term variation due to phonetic structures gets averaged out, and thus facilitates study of voice source [Löfqvist, 1986]. Modeling effectively voice source in statistical parametric speech synthesis systems is still an open challenge [Cabral et al., 2007, Drugman and Raitio, 2014]. This aspect can be potentially exploited to detect attacks by using LTAS as features.

second order statistics: Speech is a non-stationary signal. The energy in each frequency bin changes over the time. Natural speech and synthetic speech can differ in terms of such dynamics. Indeed, one successful approach to classify natural and synthetic speech signals is to use of dynamic temporal derivative information of short-term spectrum instead of static information [Sahidullah et al., 2015]. Variance of magnitude spectrum can be seen as a gross estimate of such dynamics. More precisely, standard deviation is indicative of the dynamic range of the magnitude in a frequency bin. Thus, variance could be useful for detecting attacks.

The speech signal is acquired through a sensor, which has its own channel characteristics. Information about the channel characteristics can be modeled through spectral statistics. State-of-the-art speech and speaker recognition systems employ the first order spectral statistics, e.g. mean of cepstral coefficients[1] [Furui, 1981] and the second order spectral statistics, e.g. variance of cepstral coefficients to make the system robust to channel variability. Channel information, however, is a desirable information for the detection of both physical access attacks and logical access attacks. In the case of physical access attacks, the spoofed signal is

[1] Formally, the cepstrum is the Fourier transform of the log magnitude spectrum [Bogert et al., 1963, Oppenheim and Schafer, 2004].

played through a loud speaker, which is captured via the system microphone. Such channel effects are cues for detecting attacks. For instance, hypothetically should the channel effect of the recording sensor and the loud speaker be "perfectly" removed then detecting record-and-replay attack is a non-trivial task. Channel information is also interesting for detecting logical access attacks, as the spoofed speech signal obtained from speech synthesis or voice conversion systems is injected into the system, while the bona fide speech signal is captured through the sensor of the system. In the literature it has been shown that first order and second order spectral statistics can be used to predict speech quality or quality assessment [Narwaria et al., 2012, Soni and Patil, 2016]. In the case of both physical access attacks and logical access attacks, we can expect the speech quality to differ w.r.t the bona fide speech signal.

The simplest approach to make minimal assumptions about the signal is to use the raw log-magnitude spectrum directly as feature input to the classifier. In that direction, the use of the short-term raw log-magnitude spectrum has been investigated in several works [Sahidullah et al., 2015, Xiao et al., 2015, Tian et al., 2016]. However, it has been found to perform poorly when compared to standard features such as Mel-frequency cepstral coefficients (MFCC). A potential reason for that can be that short-term raw log-magnitude spectrum contains several types of information, such as message, speaker, channel and environment. As we shall see later in Section 4.2.6, this puts onus on the classification method to learn the information that discriminates bona fide access and attack. On the contrary, as explained above, the long term spectral statistics average out phonetic structure information [Löfqvist, 1986, Huang et al., 2001] and are indicative of voice quality as well as speech quality. Thus, we hypothesize that statistics of raw log magnitude spectrum can be effectively modeled for PAD when compared to raw log magnitude spectrum. The following section presents our approach in detail.

Approach

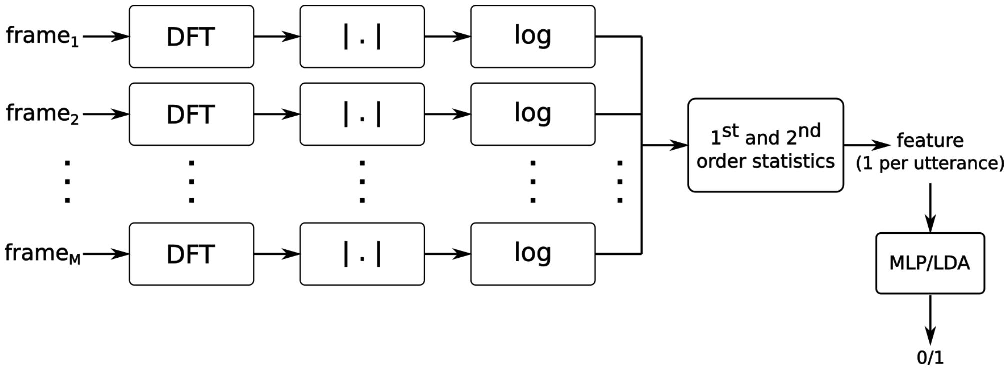

frame$_1$
frame$_2$
frame$_M$
DFT
|.|
log
1st and 2nd order statistics
feature (1 per utterance)
MLP/LDA
0/1

Figure 4.3 – LTSS-based presentation attack detection system.

As illustrated in Figure 4.3, the approach consists of three main steps:

1. *Fourier magnitude spectrum computation*: the input utterance or speech signal x is split into M frames using a frame size of w_l samples and a frame shift of w_s samples. We

first pre-emphasize each frame to enhance the high frequency components, and then compute the N-point discrete Fourier transform (DFT) $\mathscr{F}$, i.e., for frame m, $m \in \{1 \cdots M\}$:

$$X_m[k] = \mathscr{F}(x_m[n]), \tag{4.1}$$

where $n = 0 \cdots N-1$, with $N = 2^{\lceil \log_2(w_l) \rceil}$, and $k = 0 \cdots \frac{N}{2} - 1$, since the signal is symmetric around $\frac{N}{2}$ in the frequency domain. If $|X_m[k]| < 1$, we floor it to 1, i.e., we set $|X_m[k]| = 1$ so that the log spectrum is always positive. For each frame m, this process yields a vector of DFT coefficients $\mathbf{X}_m = [X_m[0] \cdots X_m[k] \cdots X_m[\frac{N}{2} - 1]]^{\mathrm{T}}$.

The number of frequency bins depends upon the frame size w_l as $N = 2^{\lceil \log_2(w_l) \rceil}$. In our approach, it is a hyper-parameter that is determined based on the performance obtained on the development set.

2. *Estimation of utterance level first order (mean) and second order (variance) statistics per Fourier frequency bin*: given the sequence of DFT coefficient vectors $\{\mathbf{X}_1, \cdots \mathbf{X}_m, \cdots \mathbf{X}_M\}$, we compute the mean $\mu[k]$ and the standard deviation $\sigma[k]$ over the M frames of the log magnitude of the DFT coefficients:

$$\mu[k] = \frac{1}{M} \sum_{m=1}^{M} \log|X_m[k]|, \tag{4.2}$$

$$\sigma^2[k] = \frac{1}{M} \sum_{m=1}^{M} \left(\log|X_m[k]| - \mu[k]\right)^2, \tag{4.3}$$

$k = 0 \cdots \frac{N}{2} - 1$.

The mean and standard deviation are concatenated, which yields a single vector representation for each utterance.

3. *Classification*: the single vector long-term spectral statistic representation of the input signal is fed into a binary classifier to decide if the utterance is a bona fide sample or an attack. In the present work, we investigate two discriminative classifiers: a linear classifier based on linear discriminant analysis (LDA) and a multi-layer perceptron (MLP) with one hidden layer.

4.2.2 CNN-based approach

The second approach goes one step further compared to the long-term spectral statistics. Rather than transforming the speech signal from time domain to frequency domain through Fourier transform and then building classifiers, the transformation of the speech signal, the features and the classifier are learned jointly from the raw speech signal.

This approach is similar to the one proposed for speaker recognition in Chapter 3 and follows the CNN-based end-to-end acoustic modeling approach originally proposed for automatic speech recognition in [Palaz et al., 2013] and developed further in [Palaz et al., 2015, 2019]. As before, as illustrated in Fig. 5.4, the CNN consists of a feature stage modeled by N convolution layers followed by a classification stage modeled by a MLP.

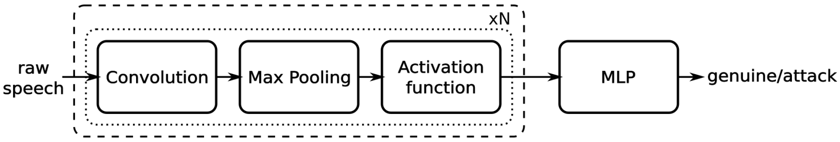


Figure 4.4 – Diagram of a convolutional neural network.

Each speech sample is split into blocks of length w_{seq} ms and shifted by w_{shift} ms that are fed successively and independently to the CNN, i.e., the CNN outputs one score per block. Fig. 4.5 shows the processing carried out in the first convolution layer. Specifically, the convolution layer consisting of n_{f1} filters processes a block of signal of length w_{seq} ms in short segments based on the length of the filters kW_1 (kernel width) and shift dW_1.

The MLP has one hidden layer composed of n_{hu} hidden units followed by a ReLU activation function. The output layer of the MLP is a softmax layer composed of two units corresponding to the bona fide class and the attack class. The parameters of the classifier and feature stages are randomly initialized and trained via the mini-batch gradient descent algorithm using a cross entropy optimization criterion.

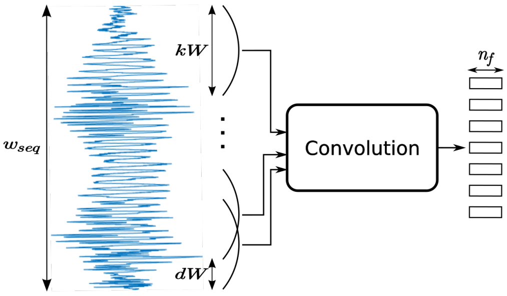


Figure 4.5 – Illustration of the convolution layer processing.

In our studies, we first investigate using only one convolution layer ($N = 1$) (corresponding to the feature stage) followed by either a single layer perceptron (SLP), i.e., no hidden layer, or a multi-layer perceptron (MLP) with a single hidden layer. The motivation behind such a simple architecture choice comes from the LTSS-based system, where the speech is transformed once through Fourier transform and the first order and second order statistics of the magnitude spectrum estimated over the utterance are classified using a linear discriminant analysis classifier or a MLP with a single hidden layer. In comparison to that, we could interpret the convolution layer as a transformation of the signal that is learned from the data in a task driven manner, as opposed to the Fourier transform, and the classification stage as a linear classifier in the case of SLP and as a non-linear classifier in the case of MLP. Furthermore, we do not perform any max-pooling as we experimentally observed that it did not improve the performance of the system. The hyper parameters w_{shift}, w_{seq}, kW_1, dW_1, n_{f1} and n_{hu} are determined based on the frame-level error rate computed over a development set.

We then compare the performance of these simple architectures to the more complex ones that we previously developped for speaker recognition in chapter 3:

1. The architecture developed on the VoxForge database in Section 3.2, which consists of 2 convolution layers with max-pooling and a fully connected layer.
2. The architecture developed on the VoxCeleb database in Section 3.3, which consists of 6 convolution layers with max-pooling and batch normalization, a global statistics layer and a fully connected layer.

4.2.3 Experimental protocol

We describe the details of the experimental setup in this section.

Databases and evaluation measures

We present experiments on the same two databases that were used for the vulnerability analysis: (a) the automatic speaker verification spoofing (ASVspoof) database, which was used during the ASVspoof 2015 Challenge and contains only logical access attacks; and (b) the audio-visual spoofing (AVspoof) database, which contains both logical and physical access attacks.

The evaluation measure used during the ASVspoof 2015 Challenge was a per-attack equal error rate (EER), i.e. the threshold is fixed independently for each type of attack with the EER criterion in both development and evaluation sets. Then, the performance of the system is evaluated by averaging the EER over the known attacks (S1-S5), the unknown attacks (S6-S10) and all the attacks. In realistic applications the decision threshold is a hyper-parameter that has to be set a priori. Thus, as presented in the following section, we adopt HTER as the evaluation measure for both ASVspoof and AVspoof databases.

Methodology

We study the two proposed approaches along with baseline systems in the following manner:

1. we first conduct experiments on the ASVspoof 2015 database using the evaluation measure employed in the Interspeech 2015 competition, i.e., per attack EER, since most of the results presented in the literature use this metric. We then extend the experiments with HTER as the evaluation measure;
2. next, we conduct experiments on the AVspoof database and study both logical access

and physical access attacks with HTER as the evaluation measure;

3. and finally, we investigate the generalization of the systems through cross-database experiments. More specifically, we use the training and development sets of one database to train the system and determine the decision threshold, and then evaluate the systems on the evaluation set of the other database with HTER as the evaluation measure.

4.2.4 Systems

In this section, we present the systems investigated, namely, (1) the baseline systems, (2) the LTSS-based systems and (3) the CNN-based systems.

Baseline systems

Most state-of-the-art presentation attack detection systems rely on the extraction of short-term cepstral features classified with a Gaussian Mixture Model (GMM). In [Korshunov and Marcel, 2016] conducted an evaluation of several such systems on the ASVspoof 2015 and AVspoof database, which was inspired by a previous evaluation in [Sahidullah et al., 2015]. We selected the systems that performed the best on each dataset: linear frequency cepstral coefficients (LFCC)-based system for the ASVspoof 2015 database as well as for the logical access attacks of AVspoof and rectangular frequency cepstral coefficients (RFCC)-based system for the physical access attacks of AVspoof. LFCC and RFCC only differ in the filter shapes: triangular for LFCC and rectangular for RFCC.

Moreover, the Constant Q Cepstral Coefficients (CQCC)-based system [Todisco et al., 2016] achieves the lowest EER on the ASVspoof 2015 database. This system was also re-implemented in [Korshunov and Marcel, 2017] on the ASVspoof 2015 and AVspoof database, which we also use as a baseline.

Feature extraction RFCC, LFCC and CQCC are computed from short-term power spectrum and the first 20 coefficients are taken. Only deltas and double-deltas of the LFCC and RFCC features are used as it was reported that static features degrade performance of PAD systems [Sahidullah et al., 2015], while the static features and their deltas and double-deltas are used in the case of CQCC.

Classifier The classifier is based on 512 mixture GMMs: one model corresponds to bona fide accesses and one to attacks, since it yields better systems when compared to SVM [Korshunov and Marcel, 2016]. The score for each utterance in the evaluation set is computed as a ratio of the log-likelihoods of the bona fide access model and attack model over the utterance.

LTSS-based systems

Preprocessing We first apply an energy-based VAD algorithm to remove silences at the beginning and end of utterances.

Feature extraction The underlying idea of the proposed approach is that the attacks could be detected based on spectral statistics. It is well known that when applying Fourier transform there is a trade-off between time and frequency resolution, i.e., the smaller the frame size, the lower the frequency resolution and the larger the frame size, the higher the frequency resolution. So, the frame size affects the estimation of the spectral statistics.

For both logical access attack and physical access attacks, we determined the frame sizes based on cross validation, while using a frame shift of 10 ms. More precisely, we varied the frame size from 16 ms to 512 ms and chose the frame size that yielded the lowest EER on the development set. For the case of logical access attacks, we found that frame size of 256 ms yields 0% EER on both ASVspoof and AVspoof databases. In the case of physical access attacks on AVspoof database, we found that 32 ms yields the lowest EER, which is 0.02%. A potential reason for this difference could be that the channel information inherent in physical access attacks is spread across frequency bins while in the case of logical access attacks the relevant information may be localized. We dwell in more detail about it later in Section 4.2.6.

Classifier We investigate two classifiers, namely, a linear classifier based on LDA and a non-linear classifier based on MLP. The input to the classifiers are the spectral statistics estimated at the utterance level as given in Equation (4.2) and Equation (4.3), i.e., one input feature vector per utterance.
LDA: the input features are projected onto one dimension with LDA , i.e., by finding the linear projection of the features components that minimizes intra-class variance and maximizes inter-class variance. We then directly use the values as scores.
MLP: we use an MLP with one hidden layer and two output units. The MLP was trained with a cost function based on the cross entropy using the back propagation algorithm and early stopping criteria. We used the Quicknet software[2] to train the MLP. The number of hidden units was determined through a coarse grid search based on the performance on the development set: 100 hidden units for AVspoof-LA and AVspoof-PA and 10000 hidden units for ASVspoof.

We also carried out investigations using GMMs. However, we do not present those studies as the error rates were significantly higher. This is potentially due to a combination of factors: (a) curse of dimensionality and (b) insufficient data for robust parameter estimation, as we obtain only one feature vector per utterance.

[2] http://www.icsi.berkeley.edu/Speech/qn.html

Table 4.3 – Hyper-parameters of the CNN trained on the three datasets: AVspoof-PA, AVspoof-LA and ASVspoof.

	w_{shift} (ms)	w_{seq} (ms)	kW_1 (samples)	dW_1 (samples)	n_{f1}	n_{hu}
AVspoof-PA	10	310	30 or 300	100	20	–
	10	310	30 or 300	10	20	100
AVspoof-LA	10	310	30 or 300	100	100	–
	10	310	30 or 300	100	20	20
ASVspoof	10	310	30 or 300	100	100	–
	10	310	30 or 300	100	20	2000

CNN-based systems

Before feeding the raw speech signal to the CNN, we normalize the signal in each frame of width w_{seq} by its mean and variance, as done in the earlier work on speech recognition [Palaz et al., 2013, 2019] and done in the speaker verification approach presented in Chapter 3.

As detailed in Section 4.2.2, there are several hyper-parameters that need to be set: w_{shift}, w_{seq}, kW_i, dW_i, n_{fi} and n_{hu}, $i = 1 \dots N$, where N is the number of convolution layers. As explained in Section 4.2.2, we first developed systems with only one convolution layer. In that case the hyper-parameters are chosen based on the frame-level accuracy achieved on the development set during the training phase. Table 4.3 presents the values of these hyper-parameters for each dataset: AVspoof-LA, AVspoof-PA and ASVspoof, found through a coarse grid search. In the case of SLPs, there is no hidden layer ($n_{hu} = 0$).

While it was found that with one convolution layer, $kW_1 = 300$ yields the lowest EER, we also trained systems with $kW_1 = 30$ based on the observations from the speaker verification studies.

Moreover, we also trained systems with exactly the same architectures used in the case of speaker verification systems (without tuning any hyper-parameters), i.e.:

- 2 convolution layers with max-pooling, detailed in Section 3.2;
- 6 convolution layers with max-pooling, batch normalization and a global statistical pooling layer, detailed in Section 3.3.

4.2.5 Results

This section presents the performance of the different systems investigated. We first present the studies on the ASVspoof 2015 database, followed by the studies on the AVspoof database and finally the cross database studies.

Performance on ASVspoof

We first compare the performance of the proposed CNN-based systems with different architectures on the ASVspoof database. In Table 4.4, we present the per attack EER obtained on the evaluation set, i.e., the metric used in the ASVspoof 2015 challenge.

Table 4.4 – Per attack EER(%) of CNN-based PAD systems on the evaluation set of ASVspoof.

Architecture	kW_1	**EER (%)**				
		Known	**Unknown**			**all**
		S1-S5	S6-S9	S10	S6-S10	S1-S10
1 conv layer, SLP	30	**0.00**	**0.01**	62.88	12.58	6.29
1 conv layer, SLP	300	0.02	0.05	58.64	1.77	5.90
1 conv layer, MLP	30	0.03	0.03	43.50	8.73	4.38
1 conv layer, MLP	300	0.09	0.15	46.70	9.46	4.78
2 conv layers, MLP	30	0.04	0.05	**23.56**	**4.75**	**2.40**
2 conv layers, MLP	300	0.16	0.15	30.17	6.15	3.16
6 conv layers, MLP	30	0.04	0.05	48.93	9.82	4.93
6 conv layers, MLP	300	0.02	0.27	49.99	10.21	5.12

The results for known and unknown attacks of the evaluation set are presented separately. As explained in Section 2.4.2, the evaluation set of the ASVspoof database contains 10 different types of attacks, denoted respectively S1 to S10, which are either voice conversion or speech synthesis attacks. The "known" attacks S1 to S5 are present in the training, development and evaluation set, while the "unknown" attacks S6 to S10 are in the evaluation set only. The attacks S1 to S4 and S6 to S9 are all based on the same "STRAIGHT" vocoder [Kawahara et al., 1999], while S5 is based on the MLSA vocoder [Fukada et al., 1992] and S10 is a unit-selection based attack, which does not require any vocoder.

We observe that for the detection of the attacks S1 to S9 all the CNN systems achieve a low error rate. The system with only one convolution layer followed by a SLP and a kernel width of 30 samples is the one that achieves the best performance. This shows that without taking the S10 attack into account, a very simple system can be sufficient to detect attacks. On the other hand, we observe that the systems with 2 convolution layers detect significantly better the S10 attack than the other systems.

In order to choose the best performing CNN system for the remainder of this section, we select the one that achieves the lowest EER on the *development* set, which corresponds to the CNN composed of 6 convolution layers with a kernel width of 300 samples, even though it does not correspond to the one achieving the lowest error rate on the evaluation set.

It is worth mentioning that the results of the systems with one convolution layer are slightly different from the ones we obtained previously and published in [Muckenhirn et al., 2017]. This is due to the fact that at the time the systems were implemented with Torch [Collobert et al., 2011], while the systems presented here are implemented with Pytorch [Paszke et al., 2017]. While the error rates obtained are comparable on the S1-S9 attacks, they are significantly higher on the S10 attack.

Table 4.5 presents the best per-attack EER reported in the literature as well as the ones obtained with our approaches. Systems “A”, “B”, “C”, “D” and “E” correspond to the best systems of the Interspeech 2015 ASVspoof competition. These systems typically employ multiple features and fusion techniques. For example, system A [Patel and Patil, 2015] uses a fusion of cochlear filter cepstral coefficients, instantaneous frequency and Mel-frequency cepstral coefficients, classified with a GMM. Similarly, system B [Novoselov et al., 2016] employs a fusion of multiple features based on Mel-frequency cepstrum and phase spectrum; transforming them into *i-vectors*; and finally classifying the *i-vectors* with a support vector machine. More information can be found in the respective citations provided in the table. “LFCC” and “CQCC” correspond to the baseline systems described in Section 4.2.4, implemented respectively in [Sahidullah et al., 2015] and [Todisco et al., 2016]. Furthermore, we alo present results obtained with deep neural networks. {DNN,RNN} corresponds to the best system obtained in [Qian et al., 2016], which is a score-level fusion of features learned with a Deep Neural Network (DNN) and classified with a LDA and features learned with a Recurrent Neural Network (RNN) and classified with a support vector machine. In both cases the features are learned from filter bank energies. The system {CNN,RNN,CNN+RNN} was developed in [Zhang et al., 2017] and is a score-level fusion of a CNN, a RNN and a combined CNN and RNN, all trained on the log-scale spectrogram of the speech utterances.

“LTSS, LDA” and “LTSS, MLP” correspond to the first proposed approach: LTSS features respectively classified with a LDA and with a MLP, “best CNN” correspond to the second proposed approach and is the CNN-based system that yields the lowest EER on the development set (6 convolution layers, $kW_1 = 300$). We then present score-level fusions of the two approaches. Score-level fusions are obtained by normalizing each score with a mean and standard deviation estimated on the development set, and simply averaging them.

The main source of error is the S10 attack and influences significantly the overall performance of the systems. More precisely, among the baseline systems, System B and System D in the ASVspoof 2015 challenge as well as system {DNN, RNN}, yield the best performance across all the attacks except for S10. On the other hand, the CQCC-based approach achieves the best performance on S10 attack, and as a consequence the best overall average performance among the baseline systems.

Similarly, we can see that, in our LTSS-based approach, the LDA classifier consistently yields a comparable or better system than the MLP classifier, except for the S10 attack. This indicates that a more sophisticated classifier is needed to detect attacks arising from concatenative speech synthesis systems. Otherwise, a linear classifier is sufficient to discriminate bona fide accesses and attacks based on LTSS. As seen in Table 4.4, the CNN-based system has a very low EER on the known attacks S1-S5, a slightly higher EER on the unknown attacks S6-S9 and a very high EER on the unknow attack S10. However, this system is complementary to the “LTSS, LDA" and "LTSS, MLP" systems. In particular the fusion of the CNN-based system with the "LTSS, MLP" system yields the lowest overall EER of 0.161%.

Table 4.5 – Per attack EER(%) of PAD systems on the evaluation set of ASVspoof.

System	**EER (%)**				
	Known	**Unknown**			**all**
	S1-S5	S6-S9	S10	S6-S10	S1-S10
A [Patel and Patil, 2015]	0.408	0.40	8.49	2.013	1.211
B [Novoselov et al., 2016]	0.008	0.01	19.57	3.922	1.965
C [Chen et al., 2015]	0.058	-	-	4.998	2.528
D [Xiao et al., 2015]	0.003	0.003	26.1	5.231	2.617
E [Alam et al., 2015]	0.041	0.085	26.393	5.347	5.694
LFCC [Korshunov and Marcel, 2016]	0.132	0.107	5.561	1.198	0.665
CQCC [Todisco et al., 2016]	0.048	0.312	**1.065**	0.463	0.256
{DNN,RNN} [Qian et al., 2016]	0.0	0.0	10.7	2.2	1.1
{CNN,RNN,CNN+RNN} [Zhang et al., 2017]	0.27	0.41	11.67	2.66	1.47
LTSS, LDA	0.026	0.056	10.220	2.089	1.057
LTSS, MLP	0.117	0.103	1.381	0.359	0.238
best CNN	0.021	0.267	49.986	10.211	5.116
fusion{best CNN, LTSS LDA}	**0.000**	**0.002**	10.220	2.046	1.023
fusion{best CNN, LTSS MLP}	0.019	0.031	1.395	**0.304**	**0.161**

As previously explained, the per attack EER may not be a good metric as the threshold is optimized on each type of attack in the evaluation set instead of being optimized on the development set. The baseline systems "LFCC" and "CQCC" were re-implemented in [Korshunov and Marcel, 2016, 2017] and evaluated instead with the HTER metric. We first show the performance difference of the original systems and of the reproduced systems in table 4.6 in order to show that both implementations lead to similar results.

Table 4.6 – Per attack EER(%) of PAD systems on the evaluation set of ASVspoof. Comparison of results originally reported in the literature and results reproduced in [Korshunov and Marcel, 2016, 2017].

System	**EER (%)** from literature		**EER (%)** reproduced	
	Known	**Unknown**	**Known**	**Unknown**
LFCC [Sahidullah et al., 2015]	0.11	1.67	0.13	1.20
CQCC [Todisco et al., 2016]	0.05	0.46	0.10	0.51

Table 4.7 presents the results of the baseline systems and proposed approaches in terms of HTER. The conclusions are similar to the ones obtained with the per attack EER metric. The "LTSS, LDA" and "best CNN" systems achieve a low HTER on the known attacks but a high HTER on the unknown attacks. On the other hand, the "LTSS, MLP" achieves a low HTER on the unknown attacks and yields the lowest HTER, significantly lower than the state-of-the-art system based on CQCC. The difference compared to the per attack EER metric is that the

fusion of the LTSS-based and CNN-based systems does not improve the overall performance.

Table 4.7 – HTER(%) of PAD systems on the evaluation set of the ASVspoof.

System	HTER (%)		
	Known	Unknown	All
LFCC [Korshunov and Marcel, 2016]	0.27	1.77	1.02
CQCC [Korshunov and Marcel, 2017]	0.15	0.91	0.53
LTSS,LDA	0.03	6.36	3.20
LTSS, MLP	0.15	**0.51**	**0.33**
best CNN (6 conv layer, MLP, $kW_1 = 300$)	0.04	9.75	4.90
fusion{best CNN, LTSS LDA}	**0.00**	7.90	3.95
fusion{best CNN, LTSS MLP}	0.05	0.68	0.37

Performance on AVspoof

Table 4.8 and 4.9 present the results on the AVspoof database, which contains both logical access (LA) attacks and physical access (PA) attacks.The baseline systems and the proposed systems were trained and evaluated independently on each type of attack.

Table 4.8 presents the performance of the proposed CNN-based systems with different architectures. On AVspoof-LA and AVspoof-PA we observe that, contrarily to what we saw on ASVspoof, increasing the number of convolution layers improves the detection performance of the systems. It is especially significant when using two convolution layers instead of one. On AVspoof-LA, using a kernel width of 30 or 300 samples lead to similar results. However, the kernel width has a high impact on the performance on AVspoof-PA when there is only one convolutional layer, using $kW_1 = 30$ instead of $kW_1 = 300$ yields a significantly higher HTER.

As we did for the ASVspoof database, we select the CNN architecture that yields the lowest EER on the development set. For both AVspoof-LA and AVspoof-PA this corresponds to the CNN with 6 convolution layers and $kW_1 = 30$.

Table 4.8 – HTER (%) of CNN-based PAD systems on the evaluation set of AVspoof, separately trained for the detection of physical access (PA) and logical access (LA) attacks.

Architecture	kW_1	LA	PA
1 conv layer, SLP	30	0.64	3.15
1 conv layer, SLP	300	0.54	0.38
1 conv layer, MLP	30	0.48	0.32
1 conv layer, MLP	300	0.49	0.10
2 conv layers, MLP	30	0.01	0.05
2 conv layers, MLP	300	0.08	0.03
6 conv layers, MLP	30	**0.00**	0.04
6 conv layers, MLP	300	0.04	**0.02**

Next, we compare our proposed approaches to the baseline systems in Table 4.9. The first

Table 4.9 – HTER (%) of PAD systems on the evaluation set of AVspoof, separately trained for the detection of Physical Access (PA) and Logical Access (LA) attacks.

System	LA	PA
RFCC [Korshunov and Marcel, 2016]	0.03	2.70
LFCC [Korshunov and Marcel, 2016]	**0.00**	5.00
CQCC [Korshunov and Marcel, 2017]	1.71	0.13
LTSS, LDA	0.04	0.18
LTSS, MLP	1.00	0.14
best CNN	0.05	**0.01**
fusion{best CNN, best LTSS}	**0.00**	0.09

thing to notice is that none of the baseline systems perform well on both LA and PA attacks. On AVspoof-LA, the LFCC and RFCC-based systems have a very low HTER. Surprisingly, the CQCC-based system performs quite poorly in comparison. On AVspoof-PA, the trend is exactly the opposite: the LFCC and RFCC-based systems perform quite poorly while the CQCC-based system yield a low HTER. On the other hand, the proposed systems "LTSS, LDA" and "best CNN" yield a low HTER on both AVspoof-PA and AVspoof-LA. We then fuse with a simple score average the best CNN with the best LTSS system (LTSS LDA for AVspoof-LA and LTSS MLP for AVspoof-PA). This fusion leads to a slight improvement on logical access attacks but to a slight degradation on physical access attacks.

Cross-database testing

Table 4.10 – HTER (%) on the evaluation sets of ASVspoof and AVspoof databases in cross database scenario. RFCC and LFCC results are taken from [Korshunov and Marcel, 2016] and CQCC results from [Korshunov and Marcel, 2017].

System	ASVspoof (Train/Dev)		AVspoof-LA (Train/Dev)	
	AVspoof-LA (Eval)	AVspoof-PA (Eval)	ASVspoof (Eval)	AVspoof-PA (Eval)
RFCC	34.93	38.54	25.58	13.20
LFCC	**0.71**	**10.58**	18.44	**8.40**
CQCC	50.02	50.01	49.34	23.72
LTSS, LDA	43.35	45.62	14.08	36.64
LTSS, MLP	50.00	50.00	46.13	23.01
best CNN	39.40	62.08	**13.34**	14.65

This section presents the study on generalization capabilities of the systems. To do so, as mentioned earlier in Section 4.2.3, we used the training and development sets of one database and the evaluation set of another database. We train the systems on the detection of logical access attacks and observe whether or not it can generalize to the detection of logical access attacks of another database and to the detection of physical access attacks.

The results are reported in Table 4.10. We observe that the LFCC-based system generalizes better to unseen data than the other systems. All the other systems perform poorly when trained on ASVspoof and evaluated on AVspoof-LA or AVspoof-PA. This is probably due to the fact that the ASVspoof data was recorded in a hemi anechoic chamber and is very clean. On the other hand, the AVspoof data was recorded in a more realistic setup with different devices and recording environments. When trained on AVspoof-LA, our CNN-based system and the LFCC-based systems perform the best.

4.2.6 Analysis

In this section, we provide further insights into the long-term spectral statistics and CNN-based approaches. We first focus on the LTSS-based approach: (1) we compare it to systems based on raw log-magnitude spectrum to show the advantage of computing long-term statistics; (2) we study the impact of the frames length, which is directly related to the frequency resolution, on the performance of the system; and (3) we analyze the LDA classifier to understand the information modeled for logical and physical access attacks. After that, we focus on the CNN-based approach and analyze the information modeled by the first convolution layer.

Comparison of LTSS to magnitude spectrum-based systems

The raw log-magnitude spectrum computed over short time frames of $\approx 20-25$ ms has been used as features in several works, classified either with a GMM [Sahidullah et al., 2015], a SVM [Villalba et al., 2015], a MLP [Xiao et al., 2015, Tian et al., 2016] or with deep architectures [Chen et al., 2015, Zhang et al., 2017, Villalba et al., 2015, Qian et al., 2016]. In Table 4.11, we present the available results on the evaluation set of the ASVspoof 2015 database with systems using either raw log-magnitude spectrum (“spec”), filter banks applied after computing the raw log-magnitude spectrum (“fbanks”) or with a log-scale spectrogram (“spectro”). “spec + MLP” corresponds to the results presented in [Tian et al., 2016], where the raw log-magnitude spectra are classified with a one hidden layer MLP. “fbanks + SVM” and “fbanks + DNN” were presented in [Villalba et al., 2015], filter banks outputs are respectively classified with a SVM and with a 2 hidden layers DNN. “fbanks + DNN [Chen et al., 2015]” corresponds to filter banks fed to a 5-layers DNN to extract features and classification using Mahalanobis distance. “fbanks + {DNN,RNN}” and “spectro + {CNN,RNN,CNN+RNN}” correspond to the systems “{DNN,RNN}” and “{CNN,RNN,CNN+RNN}” in Table 4.5. “spec + GMM” and “cep + GMM" correspond to our implementation of log-magnitude spectrum classified with a 512 mixtures GMM. “LTSS + SVM", “LTSS + LR", “LTSS + LDA” and “LTSS + MLP” correspond to long-term spectral statistics based systems with different classifiers: SVM, logistic regression (LR), LDA and MLP, respectively. We can observe that the LTSS linearly classified with LR or LDA outperforms all other systems, even the ones using neural networks with deep architectures to model magnitude spectrum. This shows that indeed the statistics are more informative than the conventional short-term raw log magnitude spectrum, as hypothesized in Section 4.2.1.

Yet, another way to understand these results is through the current trends in speaker verification, where the state-of-the-art systems are built on top of statistics of cepstral features. More precisely, a GMM-UBM trained with cepstral features is adapted on the speaker data. The parameters, more precisely the mean vectors, of the adapted GMM are then further processed to extract *i-vectors*, to build systems that are better than the standard GMM-UBM likelihood ratio based system [Dehak et al., 2011]. In our case, we observe a similar trend, i.e., modeling statistics of the raw log magnitude spectrum yields a better system than modeling the raw log magnitude spectrum.

Table 4.11 – EER(%) of magnitude spectrum-based systems on the evaluation set of ASVspoof.

System	Known	Unknown	Average
spec + MLP [Tian et al., 2016]	0.06	8.33	4.20
spec + SVM [Villalba et al., 2015]	0.13	9.58	4.85
spec + DNN [Villalba et al., 2015]	0.05	8.70	4.38
fbanks + DNN [Chen et al., 2015]	0.05	4.52	2.28
fbanks + {DNN,RNN} [Qian et al., 2016]	0.0	2.2	1.1
spectro + {CNN,RNN,CNN+RNN} [Zhang et al., 2017]	0.27	2.66	1.47
spec + GMM	0.16	3.05	1.60
cep + GMM	0.05	6.23	3.14
LTSS + SVM	0.25	2.70	1.47
LTSS + LR	0.02	1.58	0.80
LTSS + LDA	0.03	2.09	1.06
LTSS + MLP	0.10	0.40	0.25

Analysis of the impact of the frame length in LTSS

In the experimental studies, we observed that physical access attacks and logical access attacks need two different window sizes (found through cross-validation). A question that arises is: what is the role of window size or frame lengths in the proposed approach? In order to understand that, we performed evaluation studies by varying the frame lengths: 16ms, 32 ms, 64 ms, 128 ms, 256 ms and 512 ms with a frame shift of 10 ms. The length of each feature is $2^{\lceil log_2 w_l \rceil}$. For example, a frame length of 32 ms will yield features of 512 components. Figure 4.6 presents the HTER computed on the evaluation set for different frame lengths. We compare the performance impact on the detection of physical and logical access attacks of the AVspoof database and on the logical access attacks of the ASVspoof database. For the sake of clarity, unknown S10 attack results are presented separately than the rest of unknown attacks S6-S9.

For physical access attacks AVspoof-PA, it can be observed that the HTER slightly decreases from 16 ms to 128 ms and after that it increases. A likely reason for the increase after 128 ms is that in physical access attacks there is a channel effect. For that effect to be separable and meaningful for the task at hand, the channel needs to be stationary. We speculate that the stationary assumption is not holding well on longer window sizes.

For logical access attacks, it can be observed that for AVspoof-LA, ASVspoof S1-S5 (known)

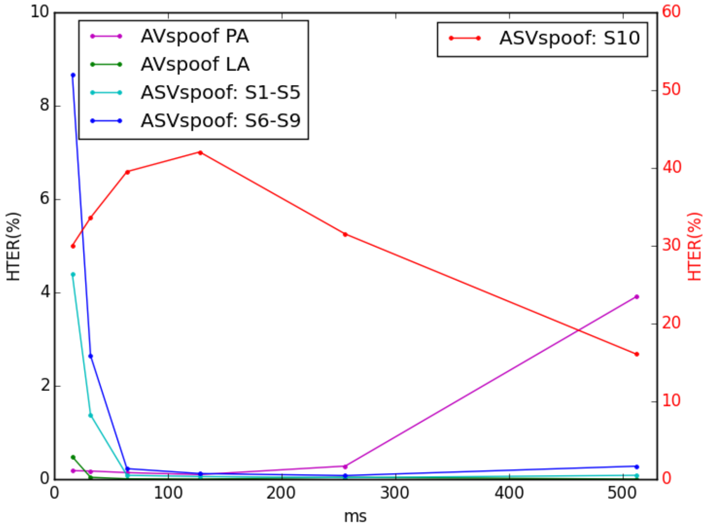


Figure 4.6 – Impact of frames lengths on the performance of the proposed LDA-based approach, evaluated on the three datasets: ASVspoof, AVspoof-LA and AVspoof-PA.

and ASVspoof S6-S9 (unknown), the HTER steadily drops from 16 ms until 256 ms with a slight increase at 512 ms. Whilst for ASVspoof S10, which contains attacks synthesized using unit selection speech synthesis system, the performance degrades at first and then steadily improves with increase of window size. This could be due to the fact that long-term temporal information is important to detect concatenated speech, since artefacts can happen at the phoneme joint areas. Our results indicate that for attacks based on parametric modeling of speech, as in the case of ASVspoof S1-S9 and AVspoof-LA, frequency resolution is not an important factor while for unit selection based concatenative synthesis, where the speech is synthesized by concatenating speech waveforms, high frequency resolution is advantageous or helpful. More specifically, together with the observations made in the previous section, we conclude that the relevant information to discriminate bona fide access and logical access attacks based on concatenative speech synthesis is highly localized in the low frequency region. This conclusion is in line with the observations made with the use of CQCC features [Todisco et al., 2016], which also provide high frequency resolution in the low frequency regions and leads to large gains on S10 attack condition.

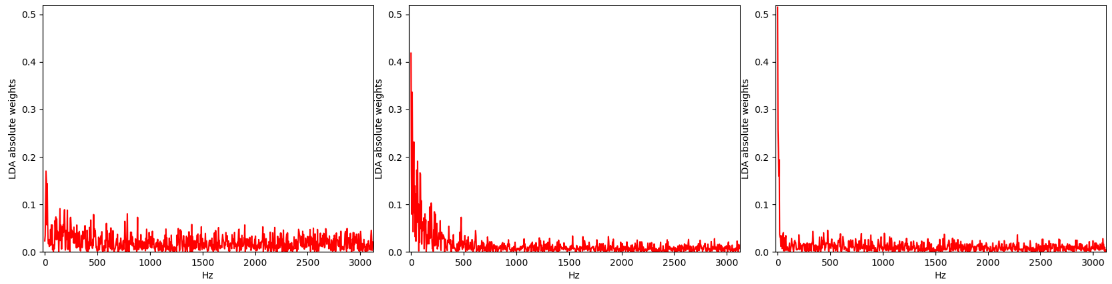
Building on these observations, we asked a question: what is the impact of window length on modeling raw log-magnitude spectrum features? We conducted an experiment, where similar to the analysis, the window length was varied as 16ms, 32ms, 64ms, 128ms and 256ms, always shifted by 10ms, and the raw log-magnitude spectrum was modeled by 512-components GMM. The EERs obtained on the evaluation set of the ASVspoof 2015 database are shown in Table 4.12. We can observe that for statistical parametric speech synthesis based attacks (S1-S9), the optimal frame length is 64ms, while it is 128ms for unit-selection based attacks (S10). Hypothetically, increase of window size should converge towards LTAS, as it would average out phonetic structure information. However, when compared to spectral statistics, increasing the window size beyond 128 ms starts degrading the performance. This could be potentially due to the difficulty in modeling discriminative information in the high dimensional raw log magnitude spectrum. In fact, in the present study modeling raw log magnitude spectrum of 512ms window became prohibitive both in terms of storage and computation.

Taken together, these analyses clearly show that typical short-term speech processing with 20-30 ms window size and other speech signal related assumptions such as source-system modeling is not a must for detecting presentation attacks.

Table 4.12 – Impact of frames lengths on the performance of raw log-magnitude spectrum classified with a GMM. EER(%) on the evaluation set of the ASVspoof database.

Frames length	**Known**	**Unknown**		**Average**
(ms)	S1-S5	S6-S9	S10	S1-S10
16	0.11	0.10	17.95	1.89
32	0.06	0.08	18.59	1.92
64	0.04	0.04	8.97	0.94
128	0.05	0.05	6.81	0.72
256	0.06	0.09	8.26	0.89

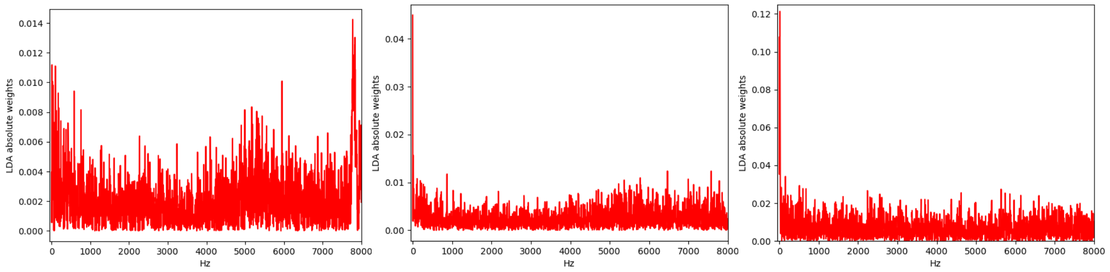
Analysis of the discrimination for the LTSS-based system



(a) physical access attacks (AVspoof(b) logical access attacks (AVspoof LA) (c) logical access attacks (ASVspoof) PA)

Figure 4.7 – 800 first LDA weights for physical and logical attacks of AVspoof and ASVspoof databases, corresponding to the frequency range [0, 3128] Hz of the spectral mean.

When classifying the features with a LDA, we project them into one dimension that best separates the bona fide accesses from the attacks in the sense that it maximizes the ratio of the "between class variance" to the "within-class variance". By analyzing this projection, we can gain insight about the importance of each component in the original space. More precisely, each extracted feature vector is a concatenation of a spectral mean and a spectral standard deviation. Thus, each half of a feature vector lies in the frequency domain, and their components are linearly spaced between 0 and 8 kHz. For example, if we compute the spectral statistics over frames of 256 ms, each spectral mean and spectral standard deviation



(a) physical access attacks (AVspoof-PA)

(b) logical access attacks (AVspoof-LA)

(c) logical access attacks (ASVspoof)

Figure 4.8 – LDA weights corresponding to the spectral standard deviation for physical and logical attacks of AVspoof and ASVspoof databases.

vectors are composed of 2048 components and the i^{th} component will correspond to the frequency ≈ $i \times 3.91$Hz. Analyzing the LDA projection vector can thus lead us to understand the importance of each frequency region.

Figure 4.7 shows the plot of the absolute values of the first 800 components of the projection vector learned by the LDA classifier trained to detect the physical access (AVspoof-PA) and logical access (AVspoof-LA) attacks on the AVspoof database, and the logical access attacks on the ASVspoof database (ASVspoof). These components correspond to the spectral mean between 0 and ≈ 3128 Hz. As the frequency increase above this value, the average amplitude of the LDA weights remains constant, which is why the high-frequency components are not shown on this figure.

We observe that when detecting physical access attacks, even though the weights are slightly higher in the low frequencies, importance is given to all the frequency bins. This can be explained by the fact that playing the fake sample through loudspeakers will modify the channel impulse response across the whole bandwidth. Thus, the relevant information to detect such attacks is spread across all frequency bins. However, in the case of logical access attacks, we observe that the largest weights correspond to a few frequency bins that are well below 50 Hz, i.e., the discriminative information in the frequency domain is highly localized in the low frequencies.

Figure 4.8 presents the LDA weights corresponding to the spectral standard deviation. The observations are similar to the ones made on the spectral mean. For the detection of physical access attacks, i.e., on AVspoof-PA, the information is spread across all the frequencies. On the other hand, in the case of logical access attacks, i.e., on AVspoof-LA and ASVspoof, the emphasis is given to the low frequencies. Furthermore we can observe that the LDA weights are smaller when compared to the spectral mean. This suggests that the mean is more discriminative than the standard deviation. To confirm this hypothesis, we conducted an investigation using stand-alone features. Table 4.13 presents the results. It can be seen that the stand-alone mean (μ) features yields a better system than the stand-alone standard deviation

(σ) features, including cross-database scenarios (systems trained on ASVspoof and evaluated on AVspoof-LA and conversely). The combined feature leads to a better system, except on ASVspoof known attacks.

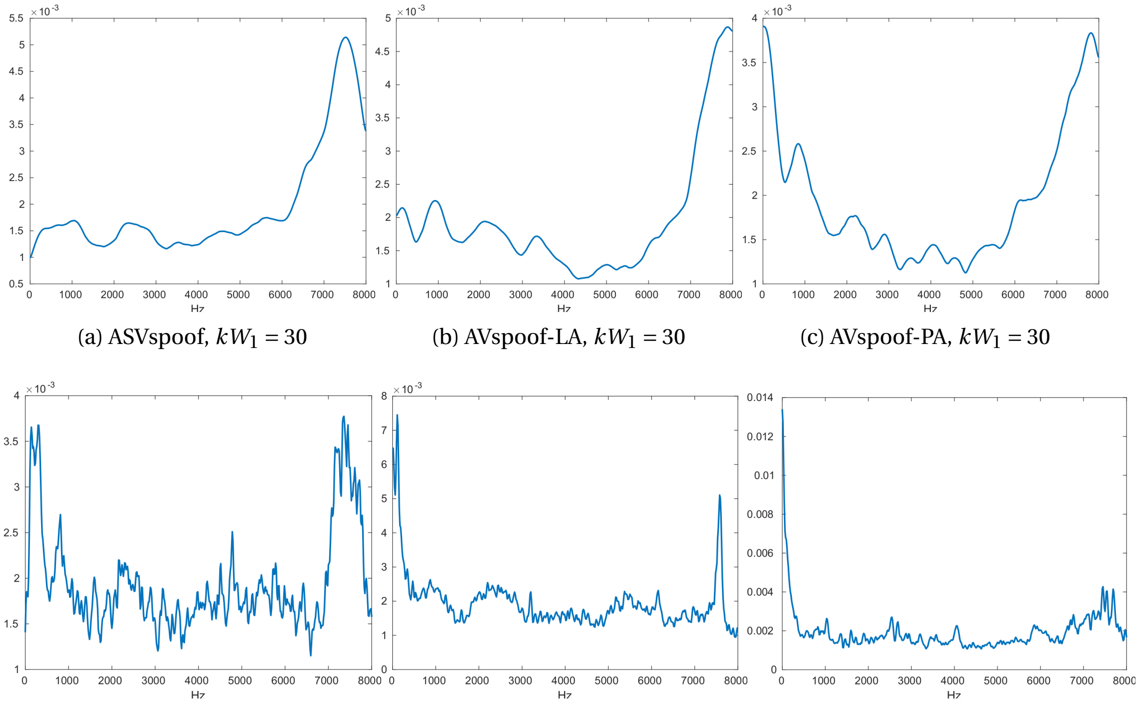
Table 4.13 – Impact of the mean and standard deviation features used alone and combined.

	AVspoof PA	**AVspoof LA**	**ASVspoof known / unknown**	**ASVspoof (Train) AVspoofLA (Eval)**	**AVspoofLA (Train) ASVspoof (Eval)**
μ	0.51	0.04	0.02 / 6.96	45.56	26.25
σ	2.03	4.65	4.10 / 19.46	55.42	45.15
$[\mu,\sigma]$	0.18	0.04	0.03 / 6.36	43.35	14.08

One explanation for the importance of the low frequency region for the detection of logical access attacks could be the following. Natural speech is primarily realized by movement of articulators that convert DC pressure variations created during respiration into AC pressure variations or speech sounds [Ohala, 1990]. Alternatively, there is an interaction between pulmonic and oral systems during speech production. In speech processing, including speech synthesis and voice conversion, the focus is primarily on glottal and oral cavity through source-system modeling. In the proposed LTSS-based approach, however, no such assumptions are being made. As a consequence, the proposed approach could be detecting logical access attacks on the basis of the effect of interaction between pulmonic and oral systems that exists in the natural speech but not in the synthetic or voice converted speech (due to source-system modeling and subsequent processing). It is understood that the interaction between pulmonary and oral cavity systems can create DC effects when producing sounds such as clicks, ejectives, implosives [Ohala, 1990]. Furthermore, human breath in the respiration process can reach the microphone and appear as “pop noise" [Shiota et al., 2015], which again manifests in the very low frequency region. Finally, it is worth mentioning that our observations are somewhat different than the observations made in [Paul et al., 2017, Sriskandaraja et al., 2016], where the authors have observed that high frequency regions were also helping in discriminating natural speech against synthetic speech. This difference can be due to the manner in which the signal is modeled and analyzed. In [Paul et al., 2017, Sriskandaraja et al., 2016], the analysis has been carried out with standard short-term speech processing, while in our case the analysis is carried out on statistics of log magnitude spectrum of 256 ms signal. So the importance of high frequency in standard short-term speech processing could be due to the differences in the spectral characteristics of specific speech sounds (e.g. fricatives) in bona fide speech and synthetic speech. In our case, the speech sound information is averaged out.

Analysis of convolution filters

The proposed CNN-based systems perform well on AVspoof-PA, AVspoof-LA and ASVspoof (except for the S10 attack). One of the question that arises is: what is being learned by the filters in the convolution layer? One way to understand the manner in which different parts of the spectrum are modeled is to observe the cumulative frequency response of the learned filters [Palaz et al., 2015, 2019], as we did previously in Section 3.4. We analyze the filters by computing the 512-points FFT of each filter in the CNN-based system and compute the cumulative frequency response by summing the magnitude spectra, as described in Eqn (3.2).



(a) ASVspoof, $kW_1 = 30$ (b) AVspoof-LA, $kW_1 = 30$ (c) AVspoof-PA, $kW_1 = 30$

(d) ASVspoof, $kW_1 = 300$ (e) AVspoof-LA, $kW_1 = 300$ (f) AVspoof-PA, $kW_1 = 300$

Figure 4.9 – Cumulative frequency response of the convolution filters learned on the ASVspoof, AVspoof-LA and AVspoof-PA databases with $kW_1 = 30$ or $kW_1 = 300$ samples.

We compare the frequency response of the sub-segmental CNNs ($kW_1 = 30$) and segmental CNNs ($kW_1 = 300$) with two convolutional layers in Figure 4.9. We first observe that the CNNs trained on ASVspoof and AVspoof-LA, i.e., on logical access attacks, extract similar information. The sub-segmental CNNs focus on high frequencies while the segmental CNNs focus both on high and low frequencies. On the other hand, the cumulative frequency responses of the CNNs trained on AVspoof-PA, i.e., on physical access attacks, are quite different. When $kW_1 = 30$ the first convolution layer focuses on low and high frequencies, while when $kW_1 = 300$ it focuses only on low frequencies, and especially on the DC component. The differences potentially explain the observations made in cross-database and cross-attack study, i.e., that these systems do not generalize well.

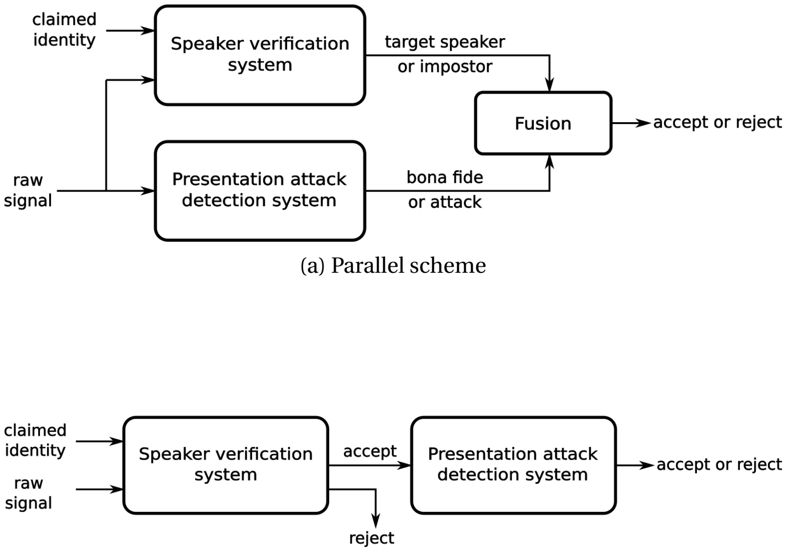
4.3 Fusion of speaker verification and presentation attack detection systems

In this section, we analyze the impact of fusing the speaker verification and presentation attack detection systems in terms of recognition performance and vulnerability.

4.3.1 Experimental protocol

The databases and evaluation protocol are the same as the ones used in Section 4.1. We selected a subset of the systems previously presented. Among the speaker verification systems we keep the two baseline systems as well as our proposed system "*r-vectors* $kW_1 = 300$" as this system yields the lowest EER (in the licit scenario) on the development set of both AVspoof and ASVspoof databases among the proposed systems. Among the presentation attack detection systems, we select the baseline system that yields the lowest HTER and a subset of the proposed systems.

4.3.2 Score-level fusion


claimed identity
Speaker verification system
target speaker or impostor
Fusion
accept or reject
raw signal
Presentation attack detection system
bona fide or attack
claimed identity
raw signal
Speaker verification system
accept
reject
Presentation attack detection system
accept or reject

(a) Parallel scheme

(b) Cascade scheme (the ordering of the systems is not important)

Figure 4.10 – Output-level fusion schemes of ASV and PAD systems.

There are two methods to combine scores of presentation attack detection systems and speaker verification systems at the *output*-level, illustrated in Figure 4.10: parallel and cascade fusion. In the parallel scenario, the scores output by the two systems are fused. The fusion can be very simple, e.g. an average, or more elaborated, e.g. training a classifier such as logistic regression [Sizov et al., 2015] or a neural network. In the cascade scenario, the utterance is first fed to one system. If the system classifies it as a positive sample (bona fide sample for presentation attack detection systems and genuine speaker for speaker verification systems), then it is fed to the second system. The order of the systems is not important as this is a commutative operation. This process is equivalent to a logical "AND" decision function, i.e., the utterance is accepted only if it is accepted by both systems. In [Korshunov and Marcel, 2017], the authors showed on the AVspoof database that the "AND" decision-level fusion, i.e., the cascading scheme, is more efficient than the parallel schemes. The main drawback of using such a decision-level fusion is that it is difficult to evaluate the trustworthy of the speaker verification system at different operation points. This would involve changing two operating points: one corresponding to the speaker verification system and the other to the presentation attack detection system. This is a non trivial task.

We propose a score-level fusion scheme that can be seen as a generalization of a logical "AND" decision-level function. $\mathscr{S}_{\text{ASV}}$ is the set of scores yielded by the speaker verification system (development and evaluation sets), $\mathscr{S}_{\text{PAD}}$ is the set of scores yielded by the presentation attack detection system (development and evaluation sets). The proposed score-level fusion consists in the following step:

1. Compute the mean and standard deviation of the scores of the development set of the speaker verification system (μ_{ASV} and σ_{ASV}) and of the presentation attack detection system (μ_{PAD} and σ_{PAD}).

2. Normalize the scores (development and evaluation sets):

$$\tilde{x}_{\text{ASV}} = \frac{x_{\text{ASV}} - \mu_{\text{ASV}}}{\sigma_{\text{ASV}}}, \forall x_{\text{ASV}} \in \mathscr{S}_{\text{ASV}}$$
$$\tilde{x}_{\text{PAD}} = \frac{x_{\text{PAD}} - \mu_{\text{PAD}}}{\sigma_{\text{PAD}}}, \forall x_{\text{PAD}} \in \mathscr{S}_{\text{PAD}}$$

3. Fix two independent thresholds on the development set such that both systems achieve an EER: τ_{ASV} and τ_{PAD}.

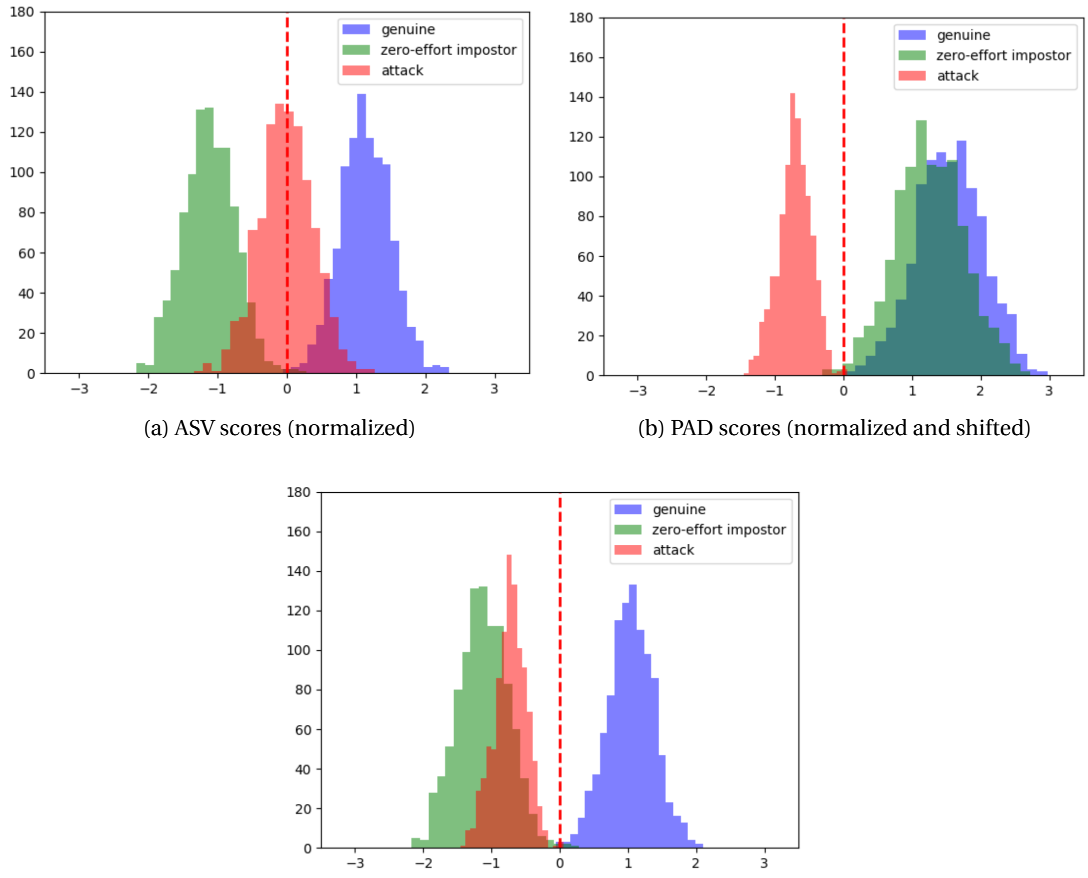
4. Align the scores (development and evaluation sets) of the two systems by shifting the scores of one system. We arbitrarily choose to shift the scores of the presentation attack detection system:

$$\tilde{\tilde{x}}_{\text{PAD}} = \tilde{x}_{\text{PAD}} - (\tau_{\text{PAD}} - \tau_{\text{ASV}})$$

5. For each sample (development and evaluation sets), compute the minimum between the scores of the systems and make the finaé decision by comparing it to a threshold Δ:

$$x_{\text{fused}} = \min(\tilde{x}_{\text{ASV}}, \tilde{\tilde{x}}_{\text{PAD}}) \lesseqgtr \Delta$$

This can be seen as a generalization of a logical "AND" and is equivalent when $\Delta = \tau_{\text{ASV}}$. The advantage of this approach is that the resulting trustworthy speaker verification system can be evaluated at different operating points by simply varying Δ and can be analyzed for example by using EPSC.



(a) ASV scores (normalized)

(b) PAD scores (normalized and shifted)

(c) minimum of the ASV and PAD scores

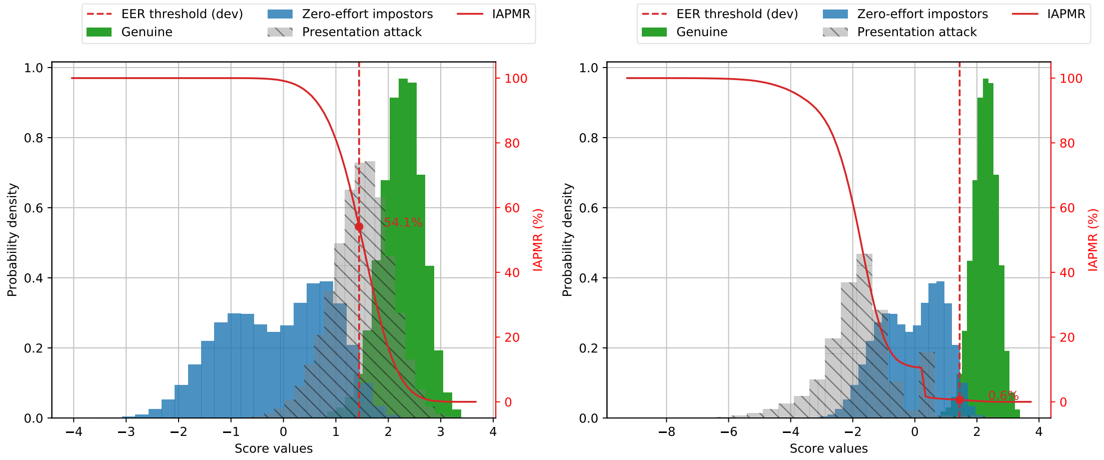
Figure 4.11 – Illustration of step (5) of the proposed score-level fusion scheme of ASV and PAD systems. The ASV and PADscores are synthetically generated.

The motivation for taking the minimum is illustrated in Figure 4.11, with synthetic scores normalized and aligned. If the scores of a sample yielded by the two systems are on the right side of threshold Δ, then their minimum will still be on the right side. On the other hand, if at least one of the two systems yields a score on the left side of the threshold Δ, then the minimum will be on the left side. Thus, emulating the "AND" logic.

4.3.3 Results

Figure 4.12 illustrates an example of a score distribution yielded by the proposed speaker verification system "*r-vectors* $kW_1 = 300$" on the ASVspoof database, with and without fusing it with a PAD system. We observe that the fusion with a PAD system is effective as it shifts the scores of the attacks to the left, i.e., below the threshold.

In Table 4.14, 4.15 and 4.16 we present the FNMR, FMR and IAPMR of the fused systems respectively on ASVspoof, AVspoof-LA and AVspoof-PA. We see that in all cases the fusion



(a) Without PAD

(b) Fused with a PAD system

Figure 4.12 – Scores histograms of the speaker verification system "*r-vectors* $kW_1 = 300$" with and without fusing it with a PAD system on the evaluation set of the ASVspoof database. The PAD system is "fusion{best CNN, LTSS MLP}".

Table 4.14 – Vulnerability analysis on the evaluation set of ASVspoof.

ASV system	**PAD system**	**FNMR (%)**	**FMR (%)**	**IAPMR(%)**
i-vectors	none	3.16	4.56	45.91
	CQCC	3.25	4.62	0.69
	LTSS MLP	3.24	4.58	**0.42**
	best CNN	3.18	4.56	6.28
	fusion{best CNN, LTSS MLP}	3.18	4.56	0.52
x-vectors	none	9.02	19.61	39.54
	CQCC	9.12	19.69	0.24
	LTSS MLP	9.06	19.63	0.25
	best CNN	9.05	19.60	1.35
	fusion{best CNN, LTSS MLP}	9.02	19.61	**0.22**
r-vectors, $kW_1 = 300$	none	3.02	3.66	54.12
	CQCC	3.11	3.80	0.84
	LTSS MLP	3.09	3.76	**0.42**
	best CNN	3.04	3.66	9.74
	fusion{best CNN, LTSS MLP}	3.03	3.68	0.61

with a presentation attack detection system do not degrade significantly the recognition performance in the licit scenario (FNMR and FMR), especially on AVspoof-LA and AVspoof-PA.

On the ASVspoof database, depending on the speaker verification system, the LTSS-based approach or the fusion of the LTSS and CNN based systems yields the lowest IAPMR. On AVspoof-LA, all PAD systems yield exactly the same performance with an IAPMR = 0. On AVspoof-PA, the baseline presentation attack detection system CQCC yields the lowest IAPMR.

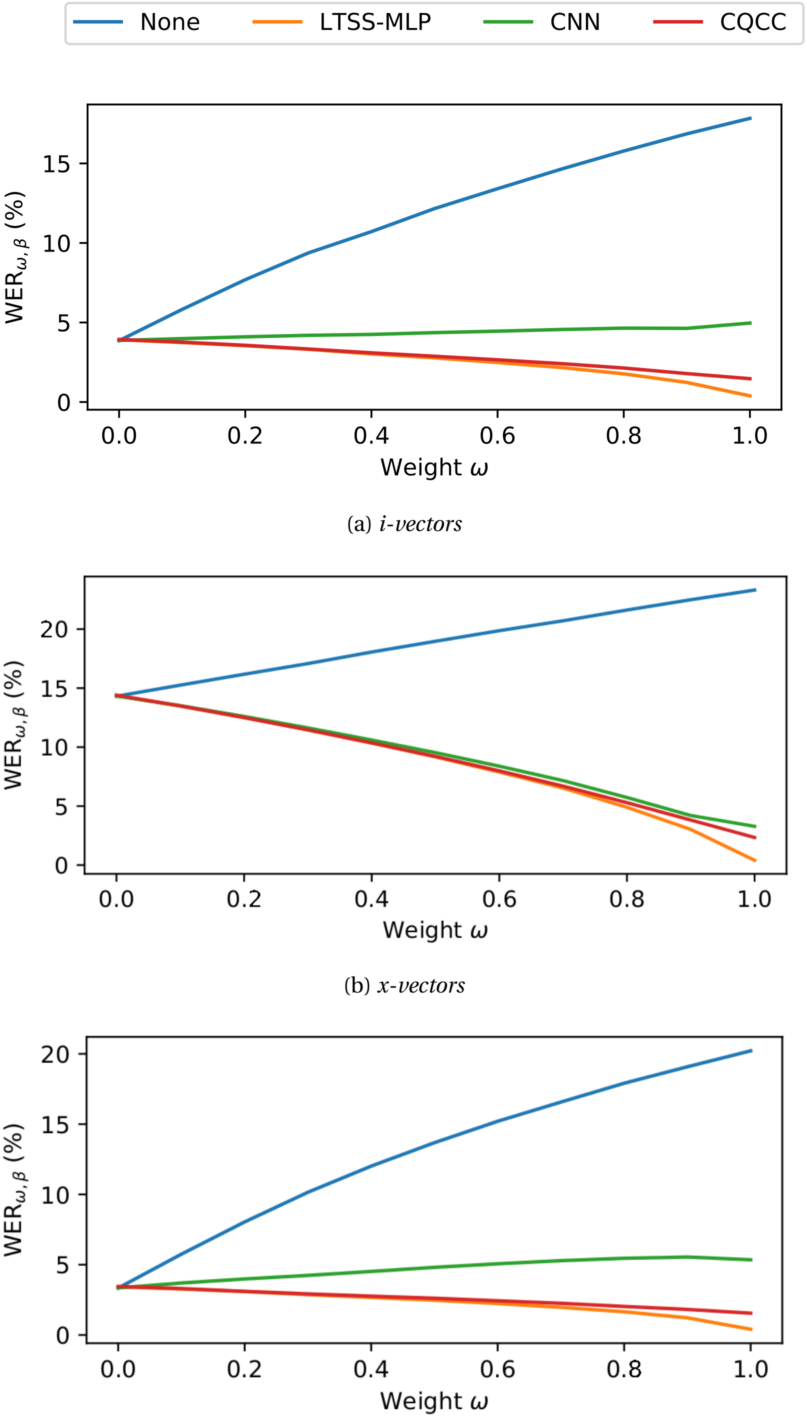
To get better insights of the differences in performance of the presentation attack detection systems on the ASVspoof database, we show the EPSC of the speaker verification systems in Figure 4.13 without and with presentation attack detection systems. We clearly observe the benefits of combining the speaker verification systems with presentation attack detection systems. In particular, in the case of ASVspoof 2015, we see that the proposed LTSS-MLP yields the lowest WER as soon as $\omega > 0$.

Table 4.15 – Vulnerability analysis on the evaluation set of AVspoof-LA.

ASV system	**PAD system**	**FNMR (%)**	**FMR (%)**	**IAPMR(%)**
i-vectors	none	4.61	7.91	99.31
	LFCC	4.61	7.91	0.00
	LTSS LDA	4.61	7.91	0.00
	best CNN	4.61	7.91	0.00
x-vectors	none	6.99	11.85	98.75
	LFCC	6.99	11.85	0.00
	LTSS LDA	6.99	11.85	0.00
	best CNN	6.99	11.85	0.00
r-vectors, $kW_1 = 300$	none	5.73	13.73	99.31
	LFCC	5.73	13.73	0.00
	LTSS LDA	5.73	13.73	0.00
	best CNN	5.73	13.73	0.00

Table 4.16 – Vulnerability analysis on the evaluation set of AVspoof-PA.

ASV system	**PAD system**	**FNMR (%)**	**FMR (%)**	**IAPMR(%)**
i-vectors	none	4.61	7.91	92.55
	CQCC	4.61	7.91	0.00
	LTSS MLP	4.61	7.92	0.05
	best CNN	4.61	7.91	0.01
x-vectors	none	6.99	11.85	88.68
	CQCC	6.99	11.85	0.00
	LTSS MLP	6.99	11.86	0.04
	best CNN	6.99	11.85	0.01
r-vectors, $kW_1 = 300$	none	5.73	13.73	98.77
	CQCC	5.82	13.70	0.00
	LTSS MLP	5.73	13.73	0.06
	best CNN	5.73	13.73	0.02



(a) *i-vectors*

(b) *x-vectors*

(c) *r-vectors* $kW_1 = 300$

Figure 4.13 – EPSC: weighted error rate of three speaker verification systems without and with PAD systems, on the evaluation set of the ASVspoof database. $\beta = 0.5$

4.4 Summary

This chapter first investigated the vulnerability of several speaker verification systems to presentation attacks. These investigations showed that both state-of-the-art *i-vectors* and *x-vectors* as well as the proposed *r-vectors* and end-to-end systems are vulnerable to such attacks. It was also found that system that yields the best performances in licit scenario, i.e. without attacks, tend to be more vulnerable. We then proposed two different approaches, which make minimal assumptions, to detect presentation attacks: one based on long-term spectral statistics and one based on CNNs trained on raw waveforms. Both these systems are competitive with the baseline systems. Finally, we showed that the score-level fusion of speaker verification and presentation attack detection systems with a scheme equivalent to an “AND” decision fusion keeps intact the performance on bona fide accesses and makes the systems robust to attacks.

5 Visualizing and understanding raw waveform-based neural networks

In the two previous chapters, we showed that training CNNs with raw waveforms performs better or comparably to state-of-the-art systems, based on conventional short-term spectral features. Similar observations have been made in the literature for other speech related tasks such as speech recognition [Palaz et al., 2013, Tüske et al., 2014, Sainath et al., 2015], voice activity detection [Zazo et al., 2016], emotion recognition [Trigeorgis et al., 2016], gender classification [Kabil et al., 2018] and speech enhancement[Fu et al., 2017, Pascual et al., 2017].

Speech signals contain a multitude of information: some related to the sound pronounced, such as formants, others related to speakers characteristics, such as fundamental frequency, as well as information related to the recording conditions such as background noise and quality of the voice. One interesting question is: what information is modelled by the neural networks trained on raw waveforms and is that information different depending on the task and architecture?

In that direction, in the previous chapters we performed an analysis of the first layer filters of the CNNs, trained respectively for speaker verification in Section 3.4 and for presentation attack detection in Section 4.2.6. Other works have also focused on the analysis of neural networks. In the context of speech recognition, in [Sainath et al., 2015] it was observed that the convolution filters, modeling 35ms of speech signal, tend to behave as a log-spaced frequency selective filter-bank. Whilst, in [Golik et al., 2015], some of the filters in the second convolution layer were found to behave like multi-resolution RASTA filters. In [Palaz et al., 2015, 2019] it was found that for speech recognition the first layer of the CNN models “sub-segmental" speech signal (signal of duration below one pitch period) and captures formant information.

These understandings are limited in the sense that they have been gained by analyzing the first or second convolution layers. They may not necessarily reveal the information the network as a whole is focusing on. The goal of this chapter is to investigate a gradient-based visualization method, which takes inspiration from the computer vision community, in order to analyze which information is modeled by neural networks trained on raw waveforms.

Section 5.1 presents the gradient-based visualization approach originally designed for images and its application to speech. Section 5.2 demonstrates its utility through a case study on phone recognition and speaker identification tasks. Section 5.3 focuses on applying the proposed visualization approach in a more systematic way by using a random Gaussian noise as input.

5.1 Gradient-based visualization

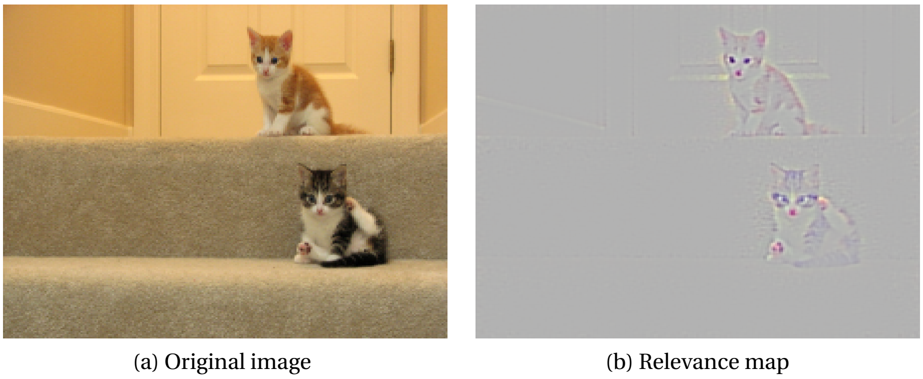
In computer vision research, it has been shown that gradient-based methods can help in visualizing the influence of each pixel in the input image on the prediction score via a relevance map [Simonyan et al., 2014, Springenberg et al., 2015, Zeiler and Fergus, 2014, Smilkov et al., 2017]. Inspired from that work, in this section we develop a gradient-based temporal and spectral relevance map extraction approach to understand the task-dependent information modeled by the CNN-based systems trained on raw waveforms.

5.1.1 Image processing

Visualization of what is captured by neural networks, especially by CNNs, is a very active field of research for image processing. Most visualization methods fall into three categories:

1. input perturbation-based methods, where the neural network is treated as a black box and the effect of altering the input image on the prediction score is measured, e.g., by occluding parts of the input [Zeiler and Fergus, 2014];
2. reconstruction-based methods [Erhan et al., 2009, Simonyan et al., 2014], where the idea is to synthesize or find among several images the input that maximizes the response of a unit of interest in the network;
3. gradient-based methods, which is the focus of this chapter.

In gradient-based methods, the gradient of a specific output unit, which is usually the one yielding the highest score, is computed with respect to each pixel of the input image. It measures how much a small variation of each pixel value will impact the prediction score. This corresponds to measuring the importance of each input value for the prediction. The result has the same size as the input image and is referred to as “relevance” map or “contribution” map. Several gradient-based methods have been proposed [Zeiler and Fergus, 2014, Simonyan et al., 2014, Springenberg et al., 2015], and essentially they only differ by how the gradient of rectified linear units (ReLU) is computed during backpropagation. In this work, we use the guided backpropagation method [Springenberg et al., 2015], as it has been shown to yield the sharpest results. In this method, the gradient at a ReLU layer is zero either if the gradient coming from above is negative or if the data value coming from below is negative. It is equivalent to computing the gradient of a ReLU function (as it is defined mathematically) but keeping only the gradients that have positive values, i.e., a positive impact on the score prediction. The motivation is twofold. First, some values will be canceled out when computing positive and negative gradients, which will create a noisy result. Second, we are only interested in visualizing what characterized a specific class, not what does not characterize it.


(a) Original image

(b) Relevance map

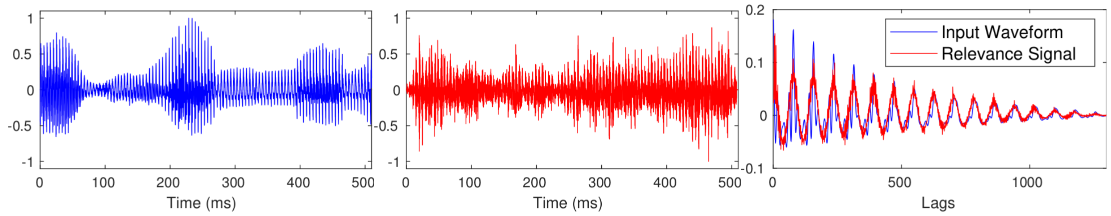
Figure 5.1 – Original image, taken from the imageNet database, and corresponding relevance map obtained with guided backpropagation.

Fig. 5.1 illustrates an example of such a visualization. The original image is taken from the imageNet database [Deng et al., 2009]. The relevance map, in Fig 5.1b, was obtained[1] with a VGG16 [Simonyan and Zisserman, 2015] trained on imageNet. The VGG16 is a deep CNN, composed of 13 convolution layers and 3 fully connected layers, with a small receptive field of size 3 × 3. It can be observed that the pixels that have a high impact on the classification results correspond to the two cats, while the pixels in the other parts of the image, e.g., the stairs, wall and door, are not important.

5.1.2 Extension to speech processing

We can apply the same approach on raw waveforms to obtain a relevance signal. An example of directly applying the guided backpropagation method in the case of raw waveforms is shown in 5.2b. This relevance signal was obtained with a CNN trained on the TIMIT database for speaker classification, which will be presented in Section 5.2. Unlike computer vision, where a human observer can visually interpret the information, the visualization of the time domain signal does not bring much insights into what important characteristics are extracted by the network because the results are difficult to interpret. Fig. 5.2c shows the auto-correlation of a short segment of the input waveform and its corresponding relevance signal. It can be observed that the relevance signal contains information related to the periodicity of the speech signal. This suggests that spectral level interpretation (obtained with conventional speech processing methods) could provide better insights. Indeed, such a relationship can be theoretically established.

[1] code used: https://github.com/ramprs/grad-cam.



(a) Input waveform (b) Relevance signal (c) Autocorrelation

Figure 5.2 – Analysis of the relevance signal obtained with guided backpropagation.

Let $\mathbf{x} = [x_0 \dots x_{N-1}]$ be a raw audio frame, belonging to class c, which is fed to a neural network. Next, discarding the softmax layer so as to remove influence from other classes, consider y^c the output unit corresponding to the class c. The gradient in the time domain with respect to input sample is defined as $f[n] = \frac{\partial y^c}{\partial x_n}$, $n = 0, \dots N-1$. We want to compute the gradient of the output unit y^c with respect to each frequency bin of the Fourier transform of the input waveform. That is, we want to visualize the impact of each frequency bin on the output. Thus, we want to compute $g[k] = \frac{\partial y^c}{\partial X_k}$ where $X_k = \sum_{n=0}^{N-1} x_n \exp(-i\frac{2\pi kn}{N})$. However, a real-valued non-constant function with complex-valued parameters does not fulfill the Cauchy-Riemann equations and is thus not differentiable. One can instead use the Wirtinger derivatives [Wirtinger, 1927] and apply the chain rule:

$$\begin{aligned}\frac{\partial y^c}{\partial X_k} &= \sum_{n=0}^{N-1} \frac{\partial y^c}{\partial x_n}\frac{\partial x_n}{\partial X_k}\\ &= \frac{1}{N}\sum_{n=0}^{N-1} \frac{\partial y^c}{\partial x_n}\frac{\partial \sum_{j=0}^{N-1} X_j e^{i\frac{2\pi jn}{N}}}{\partial X_k}\\ &= \frac{1}{N}\sum_{n=0}^{N-1} \frac{\partial y^c}{\partial x_n} e^{i\frac{2\pi kn}{N}}\\ &= \frac{1}{N}\sum_{n=0}^{N-1} f[n] e^{i\frac{2\pi kn}{N}}\end{aligned} \tag{5.1}$$

Thus,

$$g[k] = \text{DFT}^{-1}\{f[n]\}, \tag{5.2}$$

which is complex and symmetric. The derivation is simplified by dropping the complex conjugate part in the Wirtinger chain rule and by assuming that $\mathbf{x}$ and its DFT have the same dimension N. For a more rigorous derivation, the reader is referred to [Caracalla and Roebel, 2017].

The spectral relevance map can be visualized by plotting the amplitude of the first half of the signal, i.e. $|g[k]|$, for $k = 0, \dots, \lceil\frac{N}{2}\rceil - 1$. The derived result is valid for any linear transformation, invertible with respect to $\mathbf{x}$. In other words, if $\mathbf{X} = M\mathbf{x}$ and M is invertible, then $\frac{\partial y^c}{\partial \mathbf{X}} = M^{-1}\frac{\partial y^c}{\partial \mathbf{x}}$. Thus, other transforms could also be investigated.

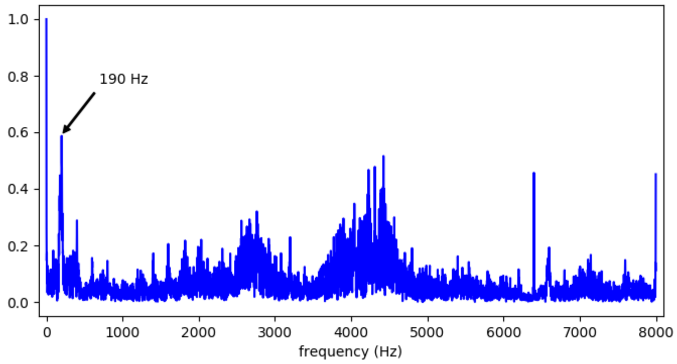


Figure 5.3 – Example of a spectral relevance map.

Figure 5.3 illustrates an example of such a spectral relevance map, obtained with a CNN trained for speaker identification on the TIMIT database and presented in Section 5.2. This spectral relevance map reveals interesting information such as a peak around the fundamental frequency. However, the input signal spans 510 ms and cannot be assumed stationary. Thus, it is difficult to interpret the spectral relevance signal. This is the case for all systems as the input signal **x** usually spans more than 250 ms. Thus, instead of computing the inverse DFT of $f[n]$ in Eqn (5.2) we used short-term analysis methods such as short-time Fourier transform. Building on that, in the next section we present two case studies.

5.2 Case studies: phone classification and speaker identification

We present two case studies on phone classification and speaker identification to demonstrate the utility of analyzing the spectral relevance signals to understand what is modeled by the CNNs. The two CNNs are trained on the same database and process the input in a sub-segmental manner with a kernel width in the first convolution layer $kW_1 = 30$ samples. The goal of this analysis is to see if the two CNNs trained for different tasks learn different information from the same raw waveforms. This is a particularly interesting case as conventional phone recognition and speaker identification systems both rely on MFCC features.

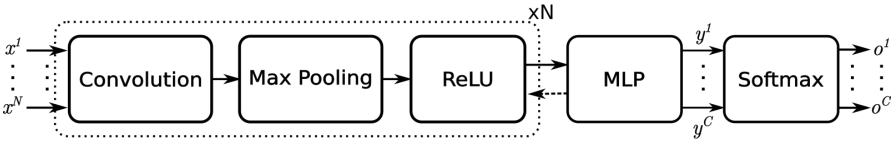


Figure 5.4 – Architecture of the raw waveform based CNN system.

The general architecture of the CNNs, illustrated in Figure 5.4, is the same as the ones used in Section 3.1 and 4.2.2. We use the same notations: w_{seq} is the length of the input, kW_i and dW_i are respectively the kernel width and kernel shift in each convolution layer $i = 1 \dots N$, which decides the block processing applied on the signal, n_f denotes the number of filters in the convolution layer.

5.2.1 Phone classification

We first describe the CNN-based phone classification system that is analyzed. We then present the analysis of the relevance signal. Finally, a study quantifying the observations is presented.

System description

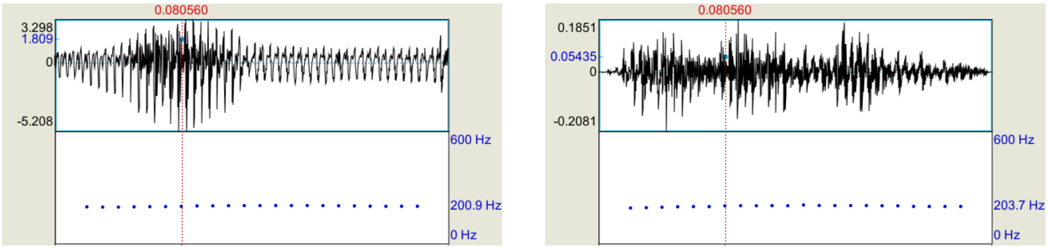
We trained a phone classifier on the TIMIT database following the protocol that is used to benchmark phone recognition systems. We chose the hyper-parameters of the system with one hidden layer from the existing work in [Palaz et al., 2019]. The hyper-parameters are presented in Table 5.1. The input to the network is of length 250ms. The CNN is composed of three convolutional layers, followed by one fully connected layer. Each convolution is followed by a max pooling with a kernel width and shift of 3 samples and by a ReLU activation function. In the original study the hyper-parameters were obtained through cross validation on the development set. The system yields phone error rate of 22.8% on the development set, and 23.6% on the test set.

Table 5.1 – Hyper-parameters of the phone classification system. n_f denotes the number of filters in the convolution layer. n_{hu} denotes the number of hidden units in the fully connected layer. kW and dW denote kernel width and kernel shift (stride).

Layer	kW	dW	n_f/n_{hu}
Conv1	30	10	80
Conv2	7	1	60
Conv3	7	1	60
MLP	-	-	1024

Visualization and analysis of relevance signals

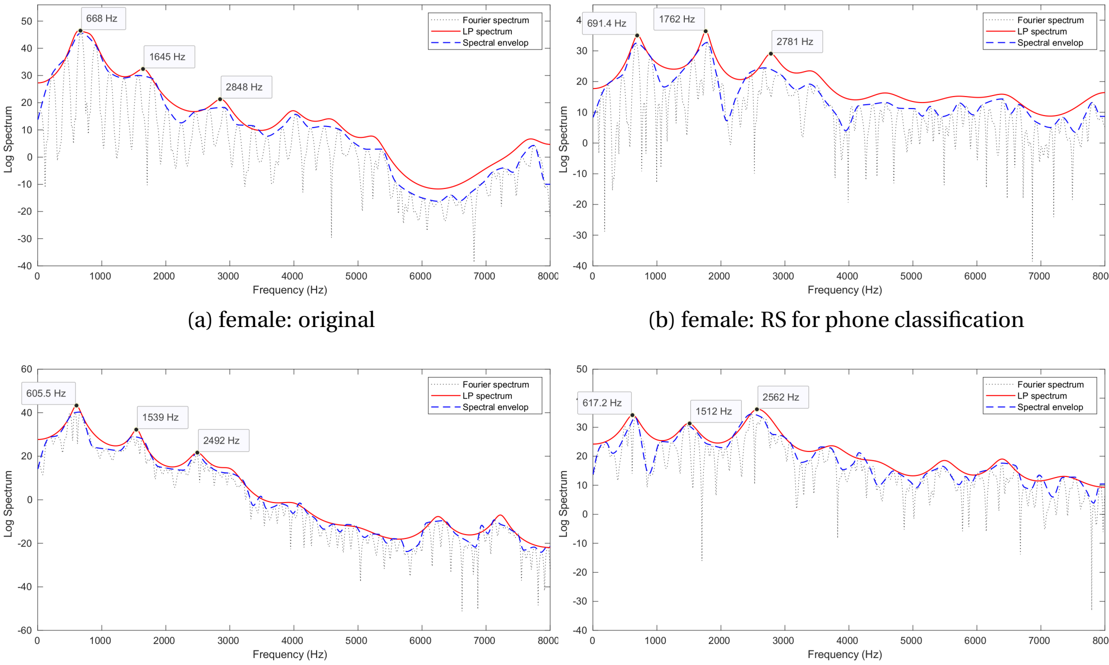
Fig. 5.5 shows the original waveform and the relevance signal corresponding to the phone /ah/ along with the pitch frequency F0 contours for the two signals obtained using Praat [Boersma, 2001]. We observe that the two signals are different in the temporal domain, however the F0 contours are similar. Fig. 5.6a and 5.6c show the short-term spectrum of the sound /ah/ produced by a male and a female speaker in exactly the same phonetic context (i.e., speaking the same text) in the TIMIT corpus. Fig. 5.6b and 5.6d show the short-term spectrum of the corresponding spectral relevance signals. The analysis window size used was of length 25 ms. We observed that, although the original signal and relevance signal differ in temporal domain, the harmonic structure and the envelop structure are similar. In particular, the first and second formants.



(a) Original

(b) Relevance signal

Figure 5.5 – F0 contours of an example waveform and corresponding temporal relevance map obtained for the phone classification system.



(a) female: original

(b) female: RS for phone classification

(c) male: original

(d) male: RS for phone classification

Figure 5.6 – Example of original and relevance signals (RS) for vowel $/ah/$, overlaid with spectral envelop (dashed:blue) and LP spectra (solid:red). Phone classification CNN trained on TIMIT.

Quantitative analysis

In order to ascertain that the relevance signal contains indeed fundamental frequency and formant information, we performed a quantitative study on the American English Vowels (AEV) dataset [Hillenbrand et al., 1995]. We chose this database because the steady state durations, fundamental frequencies and formant information are available. The analysis is done for 48 female and 45 male speakers following the standard protocol. In the steady state region, we computed the fundamental frequencies (F0) and first two formants (F1 and F2).

The formants were computed using 16th order linear prediction analysis and are averaged over a context of 10 frames around the central frame in the steady state region. We consider that the F0 and formant values are correct if it is within the range F±Δ, where F is the F0, F1 or F2 value and Δ is the respective standard deviation as specified in AEV dataset. Table 5.2 shows the average percentage accuracy of F0, F1 and F2 values for different phonemes. As it can be seen, the F0, F1 and F2 estimated from the relevance signal match well the estimates provided in the AEV dataset. This shows that, despite the CNN modeling sub-segmental speech signal (about 2ms) at the input layer, the network as a whole is capturing both fundamental frequencies and formant information.

Table 5.2 – Average accuracy in (%) of fundamental frequencies(F0) and formant frequencies (F1 and F2) of vowels produced by 45 male and 48 female speakers, estimated from relevance signal of AEV dataset.

		/ah/	/eh/	/iy/	/oa/	/uw/
F0	F	93	91	91	94	92
	M	92	90	89	93	90
F1	F	90	92	93	91	93
	M	88	92	92	89	93
F2	F	94	94	94	95	94
	M	94	93	94	94	93

5.2.2 Speaker identification

System description

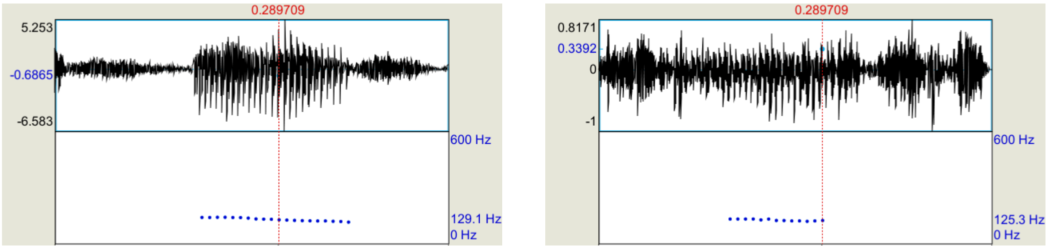
We train a CNN-based speaker identification system with the architecture and hyper-parameters used in Section 3.1, which are detailed in Table 5.3. The CNN is trained to classify the 462 speakers in the training set of the TIMIT phone recognition setup. For each speaker, 9 utterances were used for training the CNN and 1 utterance is used for validation. The utterance-level accuracy obtained on the validation set is 98.3%.

Table 5.3 – Hyper-parameters of the speaker identification system. The input to the network is of length 510ms. Definition of notations can be found in Table 5.1.

Layer	kW	dW	n_f/n_{hu}
Conv1	30	10	80
Conv2	10	1	80
MLP	-	-	100

Visualization and analysis of relevance signal

Fig. 5.7 presents an example speech signal and the corresponding relevance signal. Below each of the signal we also show F0 contours using Praat. We observe the same as for phone classification: the two signals are very different in the time domain, however the F0 contours

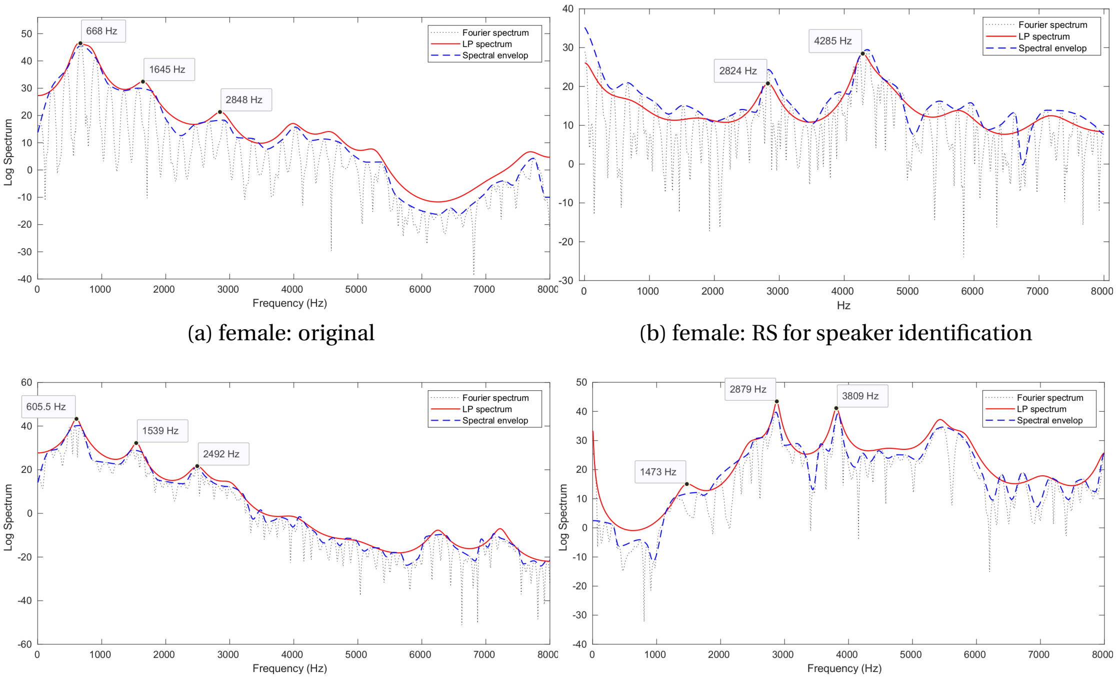


(a) Original

(b) Relevance signal

Figure 5.7 – F0 contours of an example waveform and corresponding relevance signal obtained for the speaker identification system.

are similar.



(a) female: original

(b) female: RS for speaker identification

(c) male: original

(d) male: RS for speaker identification

Figure 5.8 – Example of original and relevance signals (RS) for vowel $/ah/$, overlaid with spectral envelop (dashed:blue) and LP spectra (solid:red). Speaker identification CNN trained on TIMIT.

Fig. 5.8a and 5.8c show the short-term spectrum of the sound /ah/ produced by a male and a female speaker in exactly the same phonetic context (i.e., speaking the same text) in the TIMIT corpus. Fig. 5.8b and 5.8d show the short-term spectrum of the corresponding spectral relevance signals. The observations on these two plots are consistent with what we found on many examples belonging to different speakers and are the following. First, there is a peak in the low frequencies. Secondly, there are two high frequency regions that are emphasized.

A first region between 2000 and 3500 Hz and between 3500 and 5000 Hz. This is consistent with other studies [Kinnunen, 2003, Gallardo et al., 2014, Orman and Arslan, 2001], where authors performed an analysis of which frequency sub-bands are the most useful for speaker discrimination on the TIMIT database using either F-ratio measure [Kinnunen, 2003, Gallardo et al., 2014, Orman and Arslan, 2001] or vector ranking method [Orman and Arslan, 2001]. They also found that mid/high frequencies were discriminative: respectively between 2500Hz and 4000Hz [Kinnunen, 2003], between 2000Hz and 4000Hz [Gallardo et al., 2014] and between 3000Hz and 4500Hz [Orman and Arslan, 2001].

Quantitative analysis

In order to verify that the relevance signals contain F0 information, we conducted a quantitative study on TIMIT database by extracting and comparing the F0 contours of input speech waveforms and the F0 contours of the relevance signals for all the ten utterances from 462 speakers. We performed the analysis only for the voiced frames in the original speech signal. The result is quantified in terms of the frame level F0 value deviation between F0 contour of the relevance signal with respect to the F0 contour of the input speech waveform. Approximately, 20% of the frames with F0 value zero in the F0 contour of the relevance signal are not considered in the calculation. The mean F0 deviation was 15Hz.

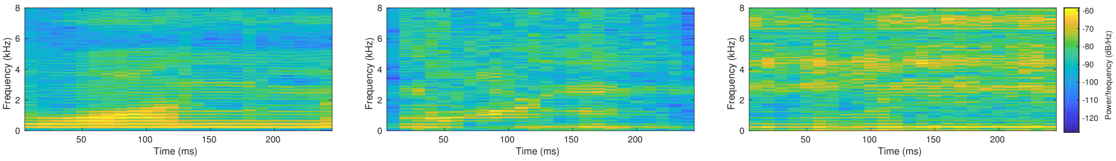
5.2.3 Phone classification versus speaker identification

The CNNs trained for speaker identification and for phone classification apply the same block processing on the raw waveforms, i.e, they both process 30 samples with a 10 samples shift. A question that arises: do the two systems focus on the same kind of spectral information?

Fig. 5.9 illustrates the difference in the information captured by the phone classification CNN and speaker verification CNN for /ah/ uttered by a TIMIT speaker. It can be observed that the phone classification CNN relevance signal retains well information related to the first two formants (around 1000 Hz) when compared to the speaker identification CNN relevance signal. We have performed informal listening tests on the relevance signals obtained with the two CNNs on a few TIMIT utterances. We have found that the relevance signals obtained with phone classification CNN are “intelligible", while the relevance signals of the speaker identification CNN are not.

5.2.4 Sub-segmental versus segmental speaker identification CNN

In Section 3.4, it was found that the first layer filters of the CNNs capture different information depending on whether the input is processed in a sub-segmental ($kW_1 = 30$) or in a segmental ($kW_1 = 300$) manner. The analysis in Section 5.2.2 focused on sub-segmental processing. To analyze the impact of processing the input in a sub-segmental or segmental manner, we train a segmental CNN with exactly the same architecture as the one presented in Table 5.3 except

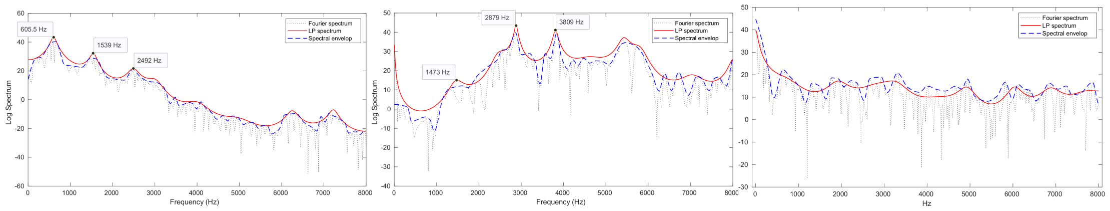


(a) Original (b) Phone Classification (c) Speaker Identification

Figure 5.9 – Spectrograms of an example waveform and corresponding spectral relevance maps obtained for phone classification CNN and speaker identification CNN.

that $kW_1 = 300$ instead of 30. The CNN is trained for a speaker identification task on the same data. We observed that the segmental CNN models mostly very low frequency bands, instead of higher frequency regions as it was the case for the sub-segmental CNN. An example is shown in Figure 5.10. This indicates that the information captured by the CNNs is not only task-dependent but also architecture-dependent.

Running the same quantitative analysis as in Section 5.2.2 to estimate F0 values yields a mean F0 deviation of 4Hz. This shows that this CNN models fundamental frequency information.



(a) original (b) RS, $kW_1 = 30$ (c) RS, $kW_1 = 300$

Figure 5.10 – Example of original and relevance signals (RS) for vowel $/ah/$, overlaid with spectral envelop (dashed:blue) and LP spectra (solid:red) of a male speaker. Speaker identification CNNs trained on TIMIT with $kW_1 = 30$ and $kW_1 = 300$.

5.3 Gradient-based visualization from random noise

In the previous section, we have presented the proposed gradient-based visualization approach and have applied it on two systems, trained on the TIMIT database for two tasks: phone classification and speaker identification. This study showed that the two CNNs model different information from the raw waveforms. The observations match either knowledge on speech characteristics (formants relate more to sounds and fundamental frequency to speakers characteristics) or observations made by other authors [Kinnunen, 2003, Gallardo et al., 2014, Orman and Arslan, 2001] on the same database (speaker characteristics also lie in higher frequency regions). This shows that the proposed gradient-based analysis method is indeed useful to analyze what information is captured by neural networks.

The main drawback of this approach is that the value of the gradient depends on the input sample that is fed to the neural network. This induces that in order to understand what information the network models as a whole we need to visualize the spectral gradients obtained on many inputs and try to find common patterns, i.e., to qualitatively assess which frequency bands are amplified or attenuated in most cases. This process is time consuming and not reliable. In order to palliate to this problem, we want to use a single input that contains all frequencies uniformly. To do so, we generate a Gaussian white noise, which is a stationary signal with a flat spectrum. We feed this noise, which has a size w_{seq} (this value varies from 510 ms to 2.41 seconds depending on the CNN), to the CNN. We compute the relevance signal, i.e. the gradient, with guided backpropagation and take its inverse Fourier transform, as in Eqn (5.1), computed over the *whole* signal since the input is stationary. The resulting spectral relevance signal shows which frequency regions are more important than others for the classification, independently of the input signal. This method can be linked to the work in [Pinto et al., 2008], which treats neural networks as a non-linear blackbox system and aims to analyze its transfer function. In that paper, white noise is fed to the neural network and the reverse transfer function is estimated with the reverse correlation method [Klein et al., 2000].

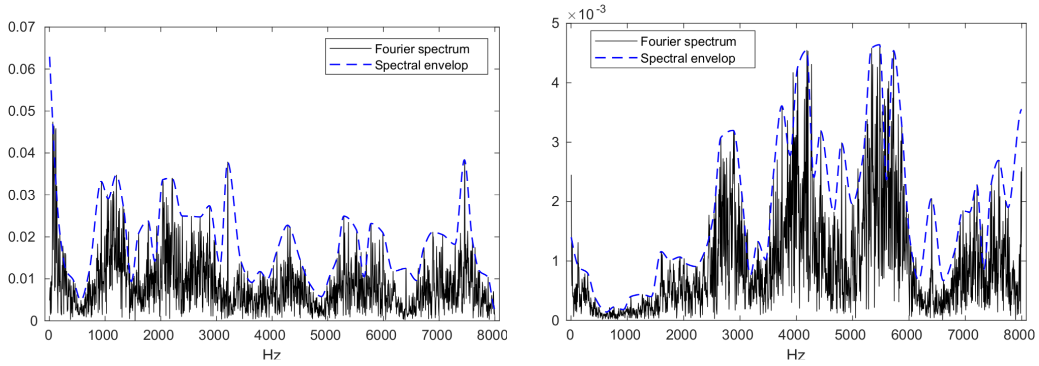
In this section, we first use this method on the CNNs from Section 5.2 trained on the TIMIT database in order to show that we reach similar conclusions. We then apply it on the CNNs previously proposed in Section 3.2.2 and 3.3.2 for speaker verification. Finally, we compare this method with one of our previous first layer analysis on CNNs trained for presentation attack detection in Section 4.2.4.

5.3.1 Case studies: phone classification and speaker identification

In Figure 5.11, we show the direct application of this method on the two case studies presented in Section 5.2. Both CNNs are trained on the TIMIT database, one for phone classification and the second one for speaker identification. Both CNNs focus on low frequencies, below 500 Hz, which corresponds to the fundamental frequencies. The phone classification CNN focuses on first formant regions among other regions. On the other hand, the speaker identification CNN models higher frequency regions between 2000 and 6000 Hz. These observations match the observations made in the previous section with speech signals as input.

5.3.2 Application to proposed speaker verification systems

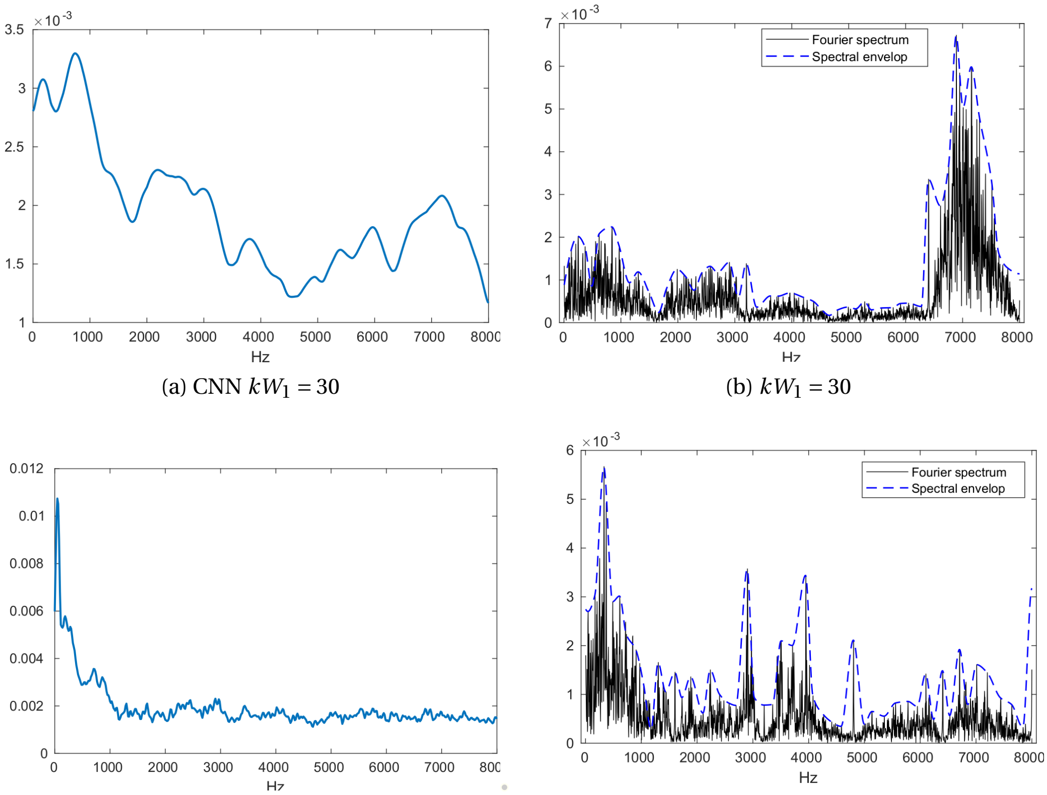
In this section, we analyze the raw waveform-based CNNs trained on two different databases: Voxforge and VoxCeleb. The CNNs trained on Voxforge are the ones described in Section 3.2.2 and are composed of two convolution layers followed by a fully connected layer. The convolution in the first layer is either applied with a short kernel width $kW_1 = 30$ (≈ 2ms) or with a long kernel width $kW_1 = 300$ (≈ 20ms). In Section 3.4 we conducted an analysis of the convolution filters of the first layer. We showed that while the frequency responses are different when $kW_1 = 30$ and when $kW_1 = 300$, they both focus on low frequencies. By conducting further analysis we had found that in both cases the filters model formant information and that when $kW_1 = 300$ the filters model fundamental frequencies.



(a) CNN trained for phone classification.

(b) CNN trained for speaker identification.

Figure 5.11 – Spectral relevance signal computed with guided backpropagation from random Gaussian noise. Both CNNs are trained on the TIMIT database.



(a) CNN $kW_1 = 30$

(b) $kW_1 = 30$

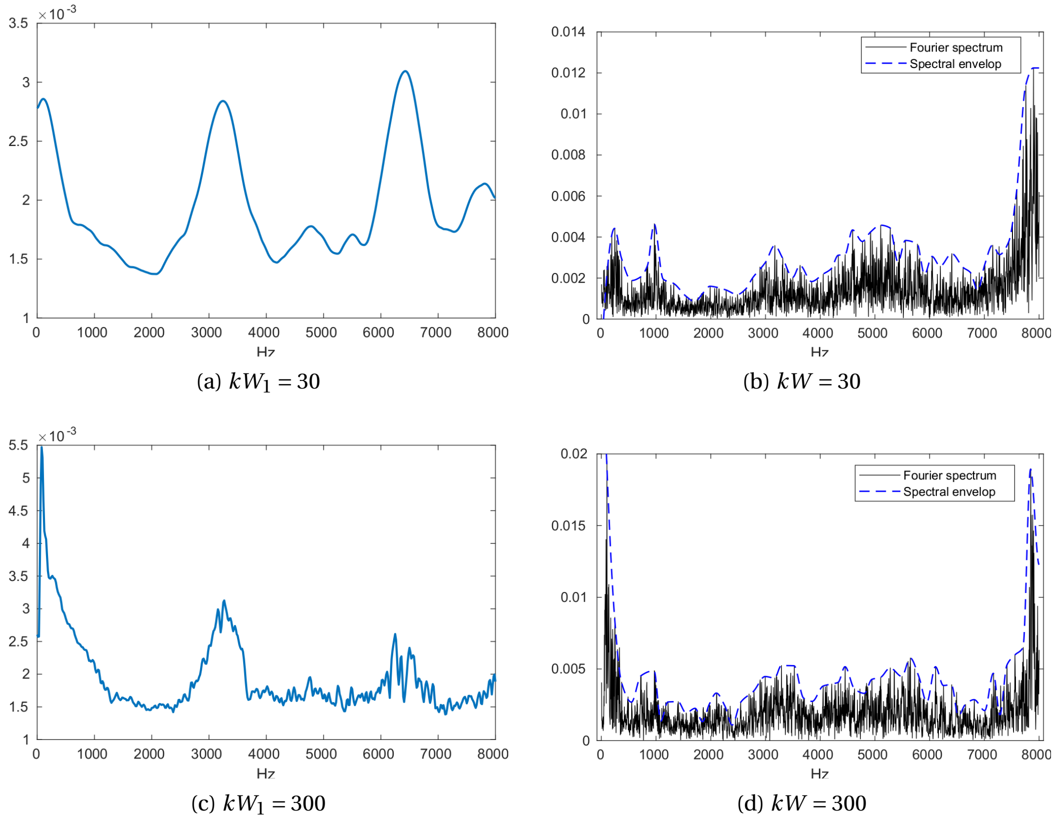
(c) $kW_1 = 300$

(d) $kW_1 = 300$

Figure 5.12 – Comparison of first layer analysis (cumulative frequency response of the convolution filters) and spectral relevance signal computed with guided backpropagation from random Gaussian noise. CNNs with 2 convolution layers trained on Voxforge database, with $kW_1 = 30$ or $kW_1 = 300$.

In Figure 5.12, we compare the cumulative frequency response of the first layer filters, computed according to Eqn (3.2) and already presented in Figure 3.5, to the spectral relevance signal obtained from random Gaussian noise. When $kW_1 = 30$, we observe that the two curves are similar below 6000 Hz. However the peak in $[6500, 8000]$ Hz is highly amplified in the spectral relevance signal. This suggests that while the high frequencies do not appear to be important in the first layer, it is actually amplified in the next layers. When $kW_1 = 300$, both the cumulative frequency response of the first layer filters and the spectral relevance signal have their main peak in the low frequencies.

In Figure 5.13 we show the same visualizations of CNNs trained on the VoxCeleb database, described in Section 3.3.2. These CNNs contain 6 convolution layers, followed by a global statistical pooling layer and by a fully connected layer. As before, the convolution in the first layer is either applied with a short kernel width $kW_1 = 30$ ($\approx$ 2ms) or with a long kernel $kW_1 = 300$ ($\approx$ 20ms). We observe that the spectral relevance signals are quite different from the cumulative frequency responses of the first layer. In particular, we observe that high frequencies are important for speaker discrimination.



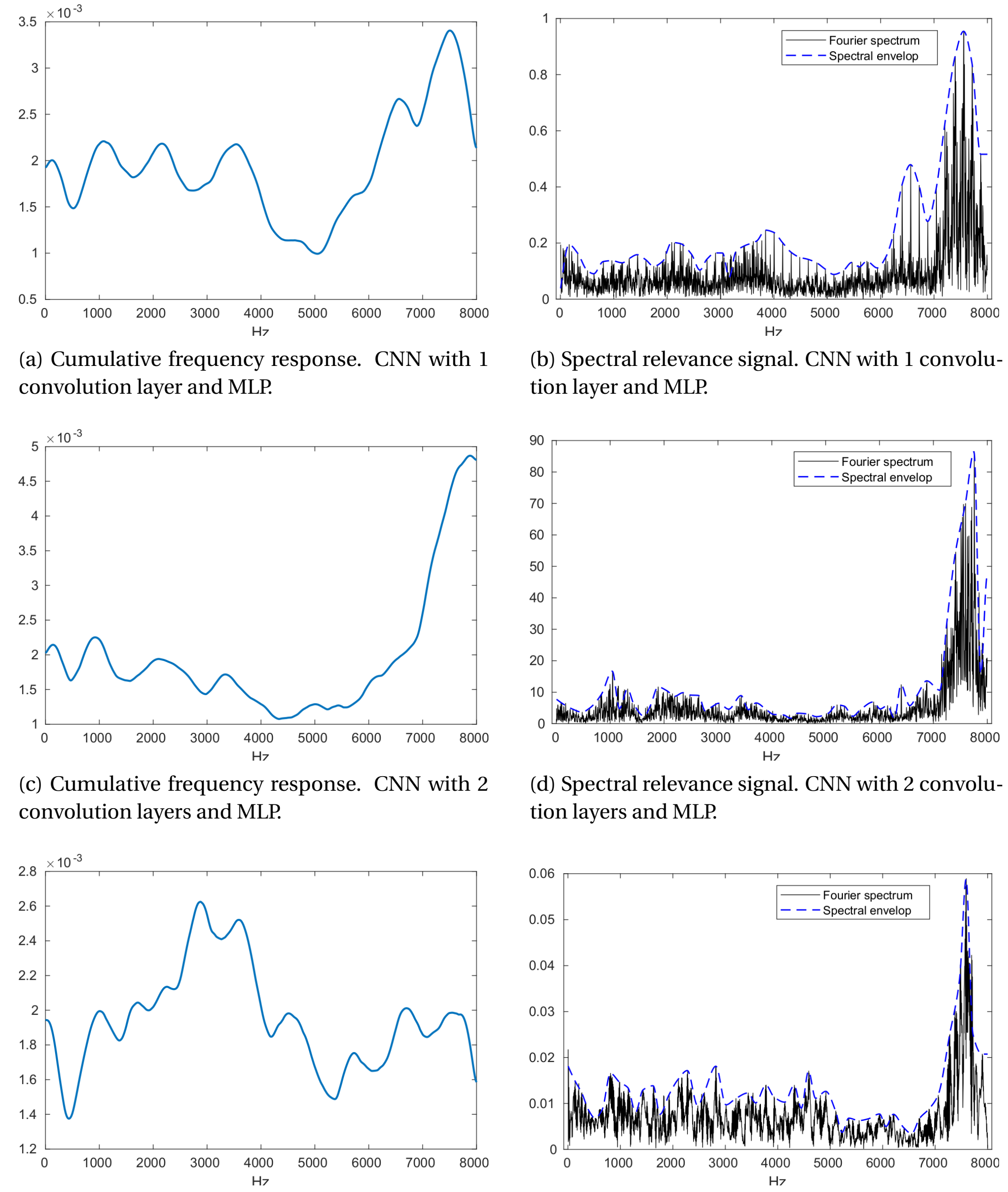
(a) $kW_1 = 30$

(b) $kW = 30$

(c) $kW_1 = 300$

(d) $kW = 300$

Figure 5.13 – Comparison of first layer analysis (cumulative frequency response of the convolution filters) and spectral relevance signal computed with guided backpropagation from random Gaussian noise. CNN with 6 convolution layers trained on VoxCeleb database.



(a) Cumulative frequency response. CNN with 1 convolution layer and MLP.

(b) Spectral relevance signal. CNN with 1 convolution layer and MLP.

(c) Cumulative frequency response. CNN with 2 convolution layers and MLP.

(d) Spectral relevance signal. CNN with 2 convolution layers and MLP.

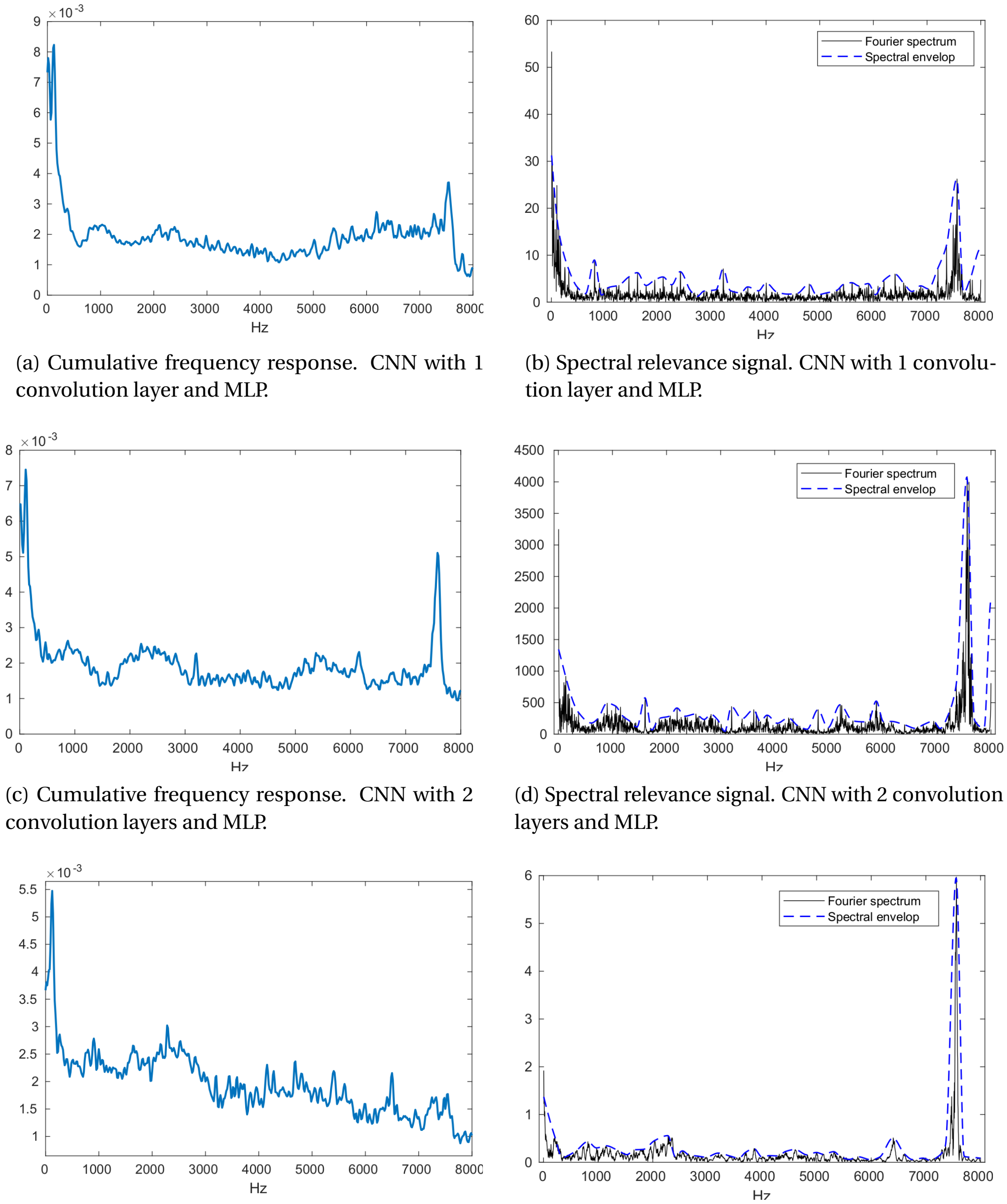
(e) Cumulative frequency response. CNN with 6 convolution layers and MLP.

(f) Spectral relevance signal. CNN with 6 convolution layers and MLP.

Figure 5.14 – Comparison of first layer analysis (cumulative frequency response of the convolution filters) and spectral relevance signal computed with guided backpropagation from random Gaussian noise. AVspoof-LA, $kW = 30$.

5.3.3 Application to proposed presentation attack detection systems: influence of depth

We want to compare our proposed gradient-based approach to the cumulative frequency response of the first layer convolution filters, described in Eqn (3.2), and previously used in



(a) Cumulative frequency response. CNN with 1 convolution layer and MLP.

(b) Spectral relevance signal. CNN with 1 convolution layer and MLP.

(c) Cumulative frequency response. CNN with 2 convolution layers and MLP.

(d) Spectral relevance signal. CNN with 2 convolution layers and MLP.

(e) Cumulative frequency response. CNN with 6 convolution layers and MLP.

(f) Spectral relevance signal. CNN with 6 convolution layers and MLP.

Figure 5.15 – Comparison of first layer analysis (cumulative frequency response of the convolution filters) and spectral relevance signal computed with guided backpropagation from random Gaussian noise. AVspoof-LA, $kW = 300$.

Sections 3.4 and 4.2.6. The goal is to both conduct a sanity check of the gradient-based method and show its advantages compared to methods that focus only on the analysis of the first layer.

In particular, we analyze the influence of the number of convolution layers in the CNN on the visualization obtained with these two methods. We run this analysis on different CNNs trained on AVspoof-LA to detect presentation attacks. The CNNs have respectively 1, 2 and 6 convolution layers and have a kernel width in the first convolution layer $kW_1 = 30$ or $kW_1 = 300$. More details can be found in Section 4.2.4.

In Figure 5.14 and 5.15 we show the resulting plots of the CNNs, using respectively a short kernel $kW_1 = 30$ and a long kernel $kW_1 = 300$ in the first convolution layer. When the CNNs contain 1 or 2 convolution layers, whether $kW_1 = 30$ or $kW_1 = 300$, the two visualization methods yield very similar results. When $kW_1 = 30$ the CNNs focus on the higher frequencies, with the highest peak centered around ≈ 7500 Hz. When $kW_1 = 300$, there is also a peak in the low frequencies, culminating at 0Hz in addition to the same peak centered around ≈ 7500 Hz. However, when the CNN contains 6 convolution layers, the two methods yield different visualization. In both cases ($kW_1 = 30$ and $kW_1 = 300$), the gradient-based visualization is similar to what is observed when the CNNs contain only 1 or 2 convolution layers. However, the cumulative frequency response is completely different. This indicates that, as expected, analyzing only the first layer might not be sufficient when the networks are deeper.

5.4 Summary

Inspired from computer vision research, this chapter proposed a gradient-based visualization approach for understanding the information modeled by CNN-based systems trained on raw waveforms. Through case studies on phone classification and speaker identification tasks, we first showed that the relevance signal obtained through guided backpropagation can be analyzed using conventional speech signal processing techniques to gain insight into the information modeled by the whole neural network. While this visualization is useful, it depends on the input values and it can be time consuming and difficult to find common trends for different inputs. We then proposed to use instead an input with a flat spectrum, such as a Gaussian random process. We used this method to analyze the information modeled by the CNNs proposed in the previous chapters and trained for speaker recognition and for presentation attack detection.

6 Conclusions and future directions

6.1 Conclusions

The focus of this thesis was on developing speaker verification systems robust to presentation attacks with minimal prior knowledge. Traditional systems are based on short-term spectral processing of speech signals. This thesis investigated alternative approaches for speaker verification and presentation attack detection using convolutional neural networks (CNNs) that take raw speech as input. We validated these approaches on different corpora against state-of-the-art systems based on standard short-term spectral features. We studied the vulnerability of the developed speaker verification systems without and with presentation attack detection. Furthermore, we investigated methods to analyze the information modeled by raw waveform-based CNNs.

In **Chapter 3**, we showed that modeling raw waveforms with neural networks yields competitive systems to the state of the art. We proposed two approaches, both based on training CNNs on raw waveforms: the first one relies on the extraction of embeddings, referred to as *r-vectors* while the second one relies on a proposed end-to-end speaker specific adaptation method. In clean conditions, the end-to-end approach outperforms the one based on *r-vectors* and yields the lowest error rate compared to state-of-the-art systems. On the other hand, this approach performs poorly in challenging conditions while *r-vectors* perform well. In both scenarios, it was found that sub-segmental and segmental CNNs capture different information and are complementary, i.e., improve the overall performance when combined. In challenging conditions, *r-vectors* yield competitive systems compared to the other embeddings based systems (*i-vectors* and *x-vectors*). Moreover, the combination of *r-vectors* with either *i-vectors* or *x-vectors* leads to significant improvement. In particular, the score-level fusion of *i-vectors*- and *r-vectors*-based systems decreases the EER by 23% compared to using *i-vectors* alone and achieves to the best of our knowledge the lowest EER reported in the literature. This indicates that the embeddings learned from raw waveforms are complementary to the ones based on standard short term spectral features. Until now, state-of-the-art speaker verification systems have mainly been focusing on vocal tract information as no efficient method was found to incorporate voice source-related information such as fundamental frequency. Through an analysis of the first layer convolution filters, as well as the analysis performed in Chapter 5, it was found that the raw waveform-based CNNs are able to capture both voice source and vocal tract system information.

In **Chapter 4**, we showed that state-of-the-art *i-vectors* and *x-vectors* systems as well as our proposed CNN-based approaches are all vulnerable to presentation attacks. In particular, we found that systems that perform well in a licit scenario, i.e., when there is no attack, tend to be more vulnerable to presentation attacks. This is possibly due to the fact that they handle better variability, such as recording quality and background noise, and thus “accept” the artefacts contained in the attack samples. We then developed two countermeasures that use minimal prior knowledge. The first method is based on utterance-level first and second order spectral statistics. We found that these statistics classified with a linear classifier (linear discriminant analysis) yield competitive performance but need a non-linear classifier to detect concatenative speech synthesis attacks (S10 attack of ASVspoof 2015 database). The second method is similar to the approach proposed for speaker verification in Chapter 3 and consists in training a binary CNN on raw waveforms. These methods, either fused or as standalone systems, yield comparable or better performance than state-of-the-art systems. The analysis of the linear discriminant analysis weights and the first layer of the CNNs showed that these two approaches focus on different information depending on whether the attacks are performed through physical or logical accesses. Cross-database and cross-attack studies suggest that the proposed approaches do not generalize well. The cross-attack aspect is understandable given that in both approaches the systems model different information for physical access attacks and logical access attacks. Our studies also show that none of the approaches based on standard short-term spectral processing truly generalize across databases. We finally showed that both state-of-the-art and proposed presentation attack detection systems make the speaker verification systems robust to presentation attacks while not degrading the speaker verification performances.

In **Chapter 5**, we focused on developing a method to analyze what information neural networks capture as a whole from the raw waveforms. Towards that, we adapted gradient-based visualization methods used in the computer vision community. We demonstrated that standard short-term speech analysis techniques can be employed to analyze the relevance signals obtained through guided backpropagation. We also found that gradient visualization obtained by feeding white noise as input could provide insights about what the neural networks are focusing on as a whole irrespective of the output class and of the input.

6.2 Future directions

Following the work proposed in this thesis, several directions of research can be considered:

1. Presentation attack detection systems, both state-of-the-art and the systems proposed

in this thesis, can achieve a high performance when trained and tested on the same type of attacks and in the same recording environments. However, as we have observed in Section 4.2.5, they do not perform well in cross-attack and cross-database scenarios. Investigating new methods that generalize well to unseen data is of paramount interest, especially as new attacks can easily be forged. Furthermore, recent advances in speech synthesis using neural networks, such as the Wavenet system [Oord et al., 2016], are leading to speech signals that are closer to natural speech. Detecting attacks generated by such speech synthesis systems is open for further research.

2. The work presented in Chapter 3 was a first step towards using neural networks trained on raw waveforms for speaker verification. We found that a simple cross-entropy training leads to performance comparable to state-of-the-art systems. One direction of research would be to investigate other architectures and loss functions, such as the center loss [Wen et al., 2016]. In our studies, we found that multiple speaker embeddings can be extracted by modeling raw waveforms and combined to improve performance of speaker verification systems. We also observe that the proposed *r-vectors* embeddings are complementary to *i-vectors* and *x-vectors*. Another direction of research can be to investigate extraction of complementary embeddings as opposed to seeking a single perfect speaker embedding. Finally, further investigations are needed to ascertain the generalization capability of raw waveform-based neural networks in varying conditions.

3. The gradient-based visualization method presented in Chapter 5 is a usefool tool to understand what information is learned by neural networks. However, we focused on one particular method. There is a plethora of different visualization methods developed in the computer vision community, which could be adapted to speech. Evaluating which method is the most appropriate is a challenging research topic.

Bibliography

Md Jahangir Alam, Patrick Kenny, Gautam Bhattacharya, and Themos Stafylakis. Development of CRIM system for the automatic speaker verification spoofing and countermeasures challenge 2015. In *Proc. of Interspeech*, 2015.

Federico Alegre, Asmaa Amehraye, and Nicholas Evans. A one-class classification approach to generalised speaker verification spoofing countermeasures using local binary patterns. In *Proc. of IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, 2013a.

Federico Alegre, Ravichander Vipperla, Asmaa Amehraye, and Nicholas Evans. A new speaker verification spoofing countermeasure based on local binary patterns. In *Proc. of Interspeech*, 2013b.

Frédéric Bimbot, Jean-François Bonastre, Corinne Fredouille, Guillaume Gravier, Ivan Magrin-Chagnolleau, Sylvain Meignier, Teva Merlin, Javier Ortega-García, Dijana Petrovska-Delacrétaz, and Douglas A Reynolds. A tutorial on text-independent speaker verification. *EURASIP Journal on Advances in Signal Processing*, 2004(4):430–451, 2004.

Paul Boersma. Praat, a system for doing phonetics by computer. *Glot International*, 5(9/10): 341–345, 2001.

Bruce Bogert, M. Healy, and J. Tukey. The quefrency alanysis of time series for echoes: Cepstrum, Pseudo-Autocovariance, Cross-Cepstrum and Saphe Cracking. In *Proc. of Symposium on Time Series Analysis*, 1963.

João P. Cabral, Steve Renals, Korin Richmond, and Junichi Yamagishi. Towards an improved modeling of the glottal source in statistical parametric speech synthesis. In *Proc. of Workshop on Speech Synthesis*, 2007.

William M Campbell, Douglas E Sturim, and Douglas A Reynolds. Support vector machines using GMM supervectors for speaker verification. *IEEE signal processing letters*, 13(5): 308–311, 2006.

Hugo Caracalla and Axel Roebel. Gradient conversion between time and frequency domains using wirtinger calculus. In *Proc. of International Conference on Digital Audio Effects*, 2017.

Ke Chen and Ahmad Salman. Learning speaker-specific characteristics with a deep neural architecture. *IEEE Transactions on Neural Networks*, 22(11):1744–1756, 2011.

Nanxin Chen, Yanmin Qian, Heinrich Dinkel, Bo Chen, and Kai Yu. Robust deep feature for spoofing detection - the SJTU system for ASVspoof 2015 challenge. In *Proc. of Interspeech*, 2015.

Ivana Chingovska, André Anjos, and Sébastien Marcel. Biometrics evaluation under spoofing attacks. *IEEE Transactions on Information Forensics and Security*, 9(12):2264–2276, 2014.

Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. Torch7: A Matlab-like Environment for Machine Learning. In *BigLearn, NIPS Workshop*, 2011.

Steven Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, 1980.

Phillip L De Leon, Michael Pucher, Junichi Yamagishi, Inma Hernaez, and Ibon Saratxaga. Evaluation of speaker verification security and detection of HMM-based synthetic speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(8):2280–2290, 2012a.

Phillip L De Leon, Bryan Stewart, and Junichi Yamagishi. Synthetic speech discrimination using pitch pattern statistics derived from image analysis. In *Proc. of Interspeech*, 2012b.

Najim Dehak. *Discriminative and generative approaches for long-and short-term speaker characteristics modeling: application to speaker verification.* PhD thesis, École de technologie supérieure, 2009.

Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2011.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. of Computer Vision and Pattern Recognition*, 2009.

Heinrich Dinkel, Nanxin Chen, Yanmin Qian, and Kai Yu. End-to-end spoofing detection with raw waveform CLDNNS. In *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

Thomas Drugman and Tuomo Raitio. Excitation modeling for HMM-based speech synthesis: Breaking down the impact of periodic and aperiodic components. In *Proc. of ICASSP*, 2014.

Pavankumar Dubagunta, Bogdan Vlasenko, and Mathew Magimai-Doss. Learning voice source related information for depression detection. In *Proc. of ICASSP*, 2019.

H. K. Dunn and S. D. White. Statistical measurements on conversational speech. *Journal of the Acoustical Society of America*, 11:278–288, 1940.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. Technical Report 1341, University of Montreal, June 2009.

N. R. French and J. C. Steinberg. Factors governing the intelligibility of speech sounds. *Journal of the Acoustical Society of America*, 19(1):90–119, 1947.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics, 2001.

Szu-Wei Fu, Yu Tsao, Xugang Lu, and Hisashi Kawai. Raw waveform-based speech enhancement by fully convolutional networks. In *Proc. of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2017.

Toshiaki Fukada, Keiichi Tokuda, Takao Kobayashi, and Satoshi Imai. An adaptive algorithm for mel-cepstral analysis of speech. In *Proc. of ICASSP*, 1992.

Sadaoki Furui. Cepstral analysis technique for automatic speaker verification. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 29(2):254–272, 1981.

Laura Fernández Gallardo, Michael Wagner, and Sebastian Möller. Spectral sub-band analysis of speaker verification employing narrowband and wideband speech. In *Proc. of Odyssey*, 2014.

Jakub Gałka, Marcin Grzywacz, and Rafał Samborski. Playback attack detection for text-dependent speaker verification over telephone channels. *Speech Communication*, 67:143 – 153, 2015.

Pavel Golik, Zoltán Tüske, Ralf Schlüter, and Hermann Ney. Convolutional neural networks for acoustic modeling of raw time signal in LVCSR. In *Proc. of Interspeech*, 2015.

John HL Hansen and Taufiq Hasan. Speaker recognition by machines and humans: A tutorial review. *IEEE Signal Processing Magazine*, 32(6):74–99, 2015.

Andrew O Hatch, Sachin Kajarekar, and Andreas Stolcke. Within-class covariance normalization for SVM-based speaker recognition. In *Proc. of International Conference on Spoken Language Processing*, 2006.

Georg Heigold, Ignacio Lopez Moreno, Samy Bengio, and Noam Shazeer. End-to-end text-dependent speaker verification. In *Proc. of ICASSP*, 2016.

Hynek Hermansky. Perceptual linear predictive (PLP) analysis of speech. *The Journal of the Acoustical Society of America*, 87(4):1738–1752, 1990.

James Hillenbrand, Laura A. Getty, Michael J. Clark, and Kimberlee Wheeler. Acoustic characteristics of american english vowels. *The Journal of the Acoustical society of America*, 97(5): 3099–3111, 1995. http://homepages.wmich.edu/~hillenbr/voweldata.html.

Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon. *Spoken language processing: A guide to theory, algorithm, and system development*, pages 517–519. Prentice Hall PTR, 2001.

ISO/IEC JTC 1/SC 37 Biometrics. DIS 30107-1, information technology – biometrics presentation attack detection. American National Standards Institute, January 2016a.

ISO/IEC JTC 1/SC 37 Biometrics. DIS 30107-3, information technology – biometric presentation attack detection – part 3: Testing and reporting. American National Standards Institute, September 2016b.

Zhe Ji, Zhi-Yi Li, Peng Li, Maobo An, Shengxiang Gao, Dan Wu, and Faru Zhao. Ensemble learning for countermeasure of audio replay spoofing attack in asvspoof2017. In *Proc. of Interspeech*, 2017.

Jee-weon Jung, Hee-soo Heo, IL-ho Yang, Hye-jin Shim, and Ha-jin Yu. Avoiding speaker overfitting in end-to-end DNNs using raw waveform for text-independent speaker verification. In *Proc. of Interspeech*, 2018.

Selen Hande Kabil, Hannah Muckenhirn, and Mathew Magimai.-Doss. On learning to identify genders from raw speech signal using CNNs. In *Proc. of Interspeech*, 2018.

Hideki Kawahara, Ikuyo Masuda-Katsuse, and Alain De Cheveigne. Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds. *Speech communication*, 27(3):187–207, 1999.

Patrick Kenny. Bayesian speaker verification with heavy-tailed priors. In *Proc. of Odyssey*, 2010.

Patrick Kenny, Gilles Boulianne, Pierre Ouellet, and Pierre Dumouchel. Joint factor analysis versus eigenchannels in speaker recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(4):1435–1447, 2007.

Patrick Kenny, Vishwa Gupta, Themos Stafylakis, Pierre Ouellet, and Jahangir Alam. Deep neural networks for extracting baum-welch statistics for speaker recognition. In *Proc. of Odyssey*, 2014.

Tomi Kinnunen. Spectral features for automatic text-independent speaker recognition. *Licentiate's Thesis, University of Joensuu*, 2003.

Tomi Kinnunen and Haizhou Li. An overview of text-independent speaker recognition: From features to supervectors. *Speech communication*, 52(1):12–40, 2010.

Tomi Kinnunen, Ville Hautamäki, and Pasi Fränti. On the use of long-term average spectrum in automatic speaker recognition. In *Proc. of International Symposium on Chinese Spoken Language Processing*, 2006.

Tomi Kinnunen, Md Sahidullah, Héctor Delgado, Massimiliano Todisco, Nicholas Evans, Junichi Yamagishi, and Kong Aik Lee. The asvspoof 2017 challenge: Assessing the limits of replay spoofing attack detection. In *Proc. of Interspeech*, 2017.

David J Klein, Didier A Depireux, Jonathan Z. Simon, and Shihab A. Shamma. Robust spectrotemporal reverse correlation for the auditory system: optimizing stimulus design. *Journal of computational neuroscience*, 9(1):85–111, 2000.

Pavel Korshunov and Sébastien Marcel. Cross-database evaluation of audio-based spoofing detection systems. In *Proc. of Interspeech*, 2016.

Pavel Korshunov and Sébastien Marcel. Impact of score fusion on voice biometrics and presentation attack detection in cross-database evaluations. *IEEE Journal of Selected Topics in Signal Processing*, 11(4):695 – 705, 2017.

Pavel Korshunov, Sébastien Marcel, Hannah Muckenhirn, A. R. Gonçalves, A. G. Souza Mello, R. P. Velloso Violato, F. O. Simões, M. U. Neto, M. de Assis Angeloni, J. A. Stuchi, H. Dinkel, N. Chen, Y. Qian, D. Paul, G. Saha, and Md Sahidullah. Overview of BTAS 2016 speaker anti-spoofing competition. In *Proc. of BTAS*, 2016.

Serife Kucur Ergunay, Elie Khoury, Alexandros Lazaridis, and Sébastien Marcel. On the vulnerability of speaker verification to realistic voice spoofing. In *Proc. of BTAS*, 2015.

Galina Lavrentyeva, Sergey Novoselov, Egor Malykh, Alexander Kozlov, Oleg Kudashev, and Vadim Shchemelinin. Audio replay attack detection with deep learning frameworks. In *Proc. of Interspeech*, 2017.

Yun Lei, Nicolas Scheffer, Luciana Ferrer, and Mitchell McLaren. A novel scheme for speaker recognition using a phonetically-aware deep neural network. In *Proc. of ICASSP*, 2014.

Timo Leino. Long-term average spectrum study on speaking voice quality in male actors. In *Proc. of Stockholm Music Acoustics Conference*, 1993.

Sue Ellen Linville and Jennifer Rens. Vocal tract resonance analysis of aging voice using long-term average spectra. *Journal of Voice*, 15(3):323–330, 2001.

Yi Liu, Yao Tian, Liang He, Jia Liu, and Michael T Johnson. Simultaneous utilization of spectral magnitude and phase information to extract supervectors for speaker verification anti-spoofing. *Proc. of Interspeech*, 2015.

Anders Löfqvist. The long-time-average spectrum as a tool in voice research. *Journal of Phonetics*, 14:471–475, 1986.

Yui Man Lui, David Bolme, P Jonathon Phillips, J Ross Beveridge, and Bruce A Draper. Preliminary studies on the good, the bad, and the ugly face recognition challenge problem. In *Proc. of Computer Vision and Pattern Recognition Workshops*, 2012.

Ivan Magrin-Chagnolleau, Guillaume Gravier, and Raphaël Blouet. Overview of the 2000-2001 elisa consortium research activities. In *Proc. of Odyssey*, 2001.

Stéphane Mallat. Understanding deep convolutional networks. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150203, 2016.

Johnny Mariéthoz and Samy Bengio. Can a professional imitator fool a GMM-based speaker verification system? Technical Report Idiap-RR-61-2005, Idiap Research Institute, 2005.

Alvin Martin and Mark Przybocki. The nist 1999 speaker recognition evaluation—an overview. *Digital signal processing*, 10(1-3):1–18, 2000.

Suely Master, Noemi de Biase, Vanessa Pedrosa, and Brasília Maria Chiari. The long-term average spectrum in research and in the clinical practice of speech therapists. *Pró-Fono Revista de Atualização Científica*, 18(1):111–120, 2006.

Elvira Mendoza, Nieves Valencia, Juana Muñoz, and Humberto Trujillo. Differences in voice quality between men and women: Use of the long-term average spectrum (LTAS). *Journal of Voice*, 10(1):59–66, 1997.

Hannah Muckenhirn, Pavel Korshunov, Mathew Magimai.-Doss, and Sébastien Marcel. Long-term spectral statistics for voice presentation attack detection. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 25(11):2098–2111, 2017.

Hannah Muckenhirn, Mathew Magimai.-Doss, and Sébastien Marcel. Towards directly modeling raw speech signal for speaker verification using CNNs. In *Proc. of ICASSP*, 2018.

Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: a large-scale speaker identification dataset. In *Proc. of Interspeech*, 2017.

Manish Narwaria, Weisi Lin, Ian Vince McLoughlin, Sabu Emmanuel, and Liang-Tien Chia. Nonintrusive quality assessment of noise suppressed speech with mel-filtered energies and support vector regression. *IEEE Transactions on Audio, Speech and Language Processing*, 20 (4):1217–1232, 2012.

Sergey Novoselov, Alexandr Kozlov, Galina Lavrentyeva, Konstantin Simonchik, and Vadim Shchemelinin. STC anti-spoofing systems for the ASVspoof 2015 challenge. In *Proc. of ICASSP*, 2016.

Akio Ogihara, UNNO Hitoshi, and Akira Shiozaki. Discrimination method of synthetic speech using pitch frequency against synthetic speech falsification. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 88(1):280–286, 2005.

John J. Ohala. Respiratory activity in speech. In W. J. Hardcastle and A. Marchal, editors, *Speech Production and Speech Modeling*. Kluwer Academic Publishers, 1990.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Alan V. Oppenheim and Ronald W. Schafer. From frequency to quefrency: A history of the cepstrum. *IEEE Signal Processing Magazine*, 21(5):95–106, 2004.

Özgür Devrim Orman and Levent M Arslan. Frequency analysis of speaker identification. In *Proc. of Odyssey*, 2001.

Dimitri Palaz. *Towards End-to-End Speech Recognition*. PhD thesis, Ecole polytechnique Fédérale de Lausanne, 2016. Thèse EPFL n° 7054.

Dimitri Palaz, Ronan Collobert, and Mathew Magimai.-Doss. Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks. In *Proc. of Interspeech*, 2013.

Dimitri Palaz, Mathew Magimai.-Doss, and Ronan Collobert. Analysis of CNN-based speech recognition system using raw speech as input. In *Proc. of Interspeech*, 2015.

Dimitri Palaz, Mathew Magimai.-Doss, and Ronan Collobert. End-to-end acoustic modeling using convolutional neural networks for HMM-based automatic speech recognition. *Speech Communication*, 108:15–32, 2019.

Vardan Papyan, Yaniv Romano, and Michael Elad. Convolutional neural networks analyzed via convolutional sparse coding. *Journal of Machine Learning Research*, 18(83):1–52, 2017.

Santiago Pascual, Antonio Bonafonte, and Joan Serrà. Segan: Speech enhancement generative adversarial network. In *Proc. of Interspeech*, 2017.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *Autodiff, NIPS Workshop*, 2017.

Tanvina B Patel and Hemant A Patil. Combining evidences from mel cepstral, cochlear filter cepstral and instantaneous frequency features for detection of natural vs. spoofed speech. In *Proc. of Interspeech*, 2015.

Dipjyoti Paul, Monisankha Pal, and Goutam Saha. Spectral features for synthetic speech detection. *IEEE Journal of Selected Topics in Signal Processing*, 11(4):605–617, 2017.

Joel Pinto, Garimella SVS Sivaram, and Hynek Hermansky. Reverse correlation for analyzing mlp posterior features in asr. In *Proc. of International Conference on Text, Speech and Dialogue*, 2008.

Fabrice Plante, Georg F. Meyer, and William A. Ainsworth. A pitch extraction reference database. In *Proc. of EuroSpeech*, 1995.

Simon JD Prince and James H Elder. Probabilistic linear discriminant analysis for inferences about identity. In *Proc. of International Conference on Computer Vision*, 2007.

Yanmin Qian, Nanxin Chen, and Kai Yu. Deep features for automatic spoofing detection. *Speech Communication*, 85:43–52, 2016.

Nalini K. Ratha, Jonathan H. Connell, and Ruud M. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM Systems Journal*, 40(3):614–634, 2001.

Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. Speaker verification using adapted Gaussian mixture models. *Digital signal processing*, 10(1):19–41, 2000.

Md Sahidullah, Tomi Kinnunen, and Cemal Hanilçi. A comparison of features for synthetic speech detection. In *Proc. of Interspeech*, 2015.

Tara N Sainath, Ron J Weiss, Andrew Senior, Kevin W Wilson, and Oriol Vinyals. Learning the speech front-end with raw waveform CLDNNs. In *Proc. of Interspeech*, 2015.

Marvin R. Sambur. Selection of acoustic features for speaker identification. *IEEE Transactions on Audio, Speech, and Signal Processing*, 23(2):176–182, 1975.

Sayaka Shiota, Fernando Villavicencio, Junichi Yamagishi, Nobutaka Ono, Isao Echizen, and Tomoko Matsui. Voice liveness detection algorithms based on pop noise caused by human breath for automatic speaker verification. In *Proc. of Interspeech*, 2015.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. of International Conference on Learning Representations*, 2015.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Proc. of International Conference on Learning Representations*, 2014.

Aleksandr Sizov, Elie Khoury, Tomi Kinnunen, Zhizheng Wu, and Sébastien Marcel. Joint speaker verification and anti-spoofing in the i-vector space. *IEEE Transactions on Information Forensics and Security*, 10(4):821–832, 2015.

Kåre Sjölander and Jonas Beskow. Wavesurfer - an open source speech tool. In *Proc. of International Conference on Spoken Language Processing*, 2000.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smooth-Grad: removing noise by adding noise. In *ICML workshop on visualization for deep learning*, 2017.

Lindsey K Smith and Alexander M Goberman. Long-time average spectrum in individuals with parkinson disease. *NeuroRehabilitation*, 35(1):77–88, 2014.

David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust dnn embeddings for speaker recognition. In *Proc. of ICASSP*, 2018.

Meet H Soni and Hemant A Patil. Non-intrusive quality assessment of synthesized speech using spectral features and support vector regression. In *Proc. of Speech Synthesis Workshop*, 2016.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. In *Proc. of International Conference on Learning Representations*, 2015.

Kaavya Sriskandaraja, Vidhyasaharan Sethu, Phu Ngoc Le, and Eliathamby Ambikairajah. Investigation of sub-band discriminative information between spoofed and genuine speech. In *Proc. of Interspeech*, 2016.

Johan Sundberg. Perception of singing. *The psychology of music*, 1999:171–214, 1999.

Kristine Tanner, Nelson Roy, Andrea Ash, and Eugene H Buder. Spectral moments of the long-term average spectrum: Sensitive indices of voice change after therapy? *Journal of Voice*, 19(2):211–222, 2005.

Xiaohai Tian, Zhizheng Wu, Xiong Xiao, Eng Siong Chng, and Haizhou Li. Spoofing detection from a feature representation perspective. In *Proc. of ICASSP*, 2016.

Massimiliano Todisco, Héctor Delgado, and Nicholas Evans. A new feature for automatic speaker verification anti-spoofing: Constant Q cepstral coefficients. In *Proc. of Odyssey*, 2016.

George Trigeorgis, Fabien Ringeval, Raymond Brueckner, Erik Marchi, Mihalis A. Nicolaou, Björn W. Schuller, and Stefanos Zafeiriou. Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network. In *Proc. of ICASSP*, 2016.

Zoltán Tüske, Pavel Golik, Ralf Schlüter, and Hermann Ney. Acoustic modeling with deep neural networks using raw time signal for LVCSR. In *Proc. of Interspeech*, 2014.

Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez. Deep neural networks for small footprint text-dependent speaker verification. In *Proc. of ICASSP*, 2014.

Jesús Villalba and Eduardo Lleida. Detecting replay attacks from far-field recordings on speaker verification systems. In *Biometrics and ID Management*, pages 274–285. Springer, 2011.

Jesús Villalba, Antonio Miguel, Alfonso Ortega, and Eduardo Lleida. Spoofing detection with DNN and one-class SVM for the ASVspoof 2015 challenge. In *Proc. of Interspeech*, 2015.

Bogdan Vlasenko, Jilt Sebastian, D S Pavan Kumar, and Mathew Magimai.-Doss. Implementing fusion techniques for the classification of paralinguistic information. In *Proc. of Interspeech*, 2018.

Robbie Vogt and Sridha Sridharan. Explicit modelling of session variability for speaker verification. *Computer Speech & Language*, 22(1):17–38, 2008.

Longbiao Wang, Yohei Yoshida, Yuta Kawakami, and Seiichi Nakagawa. Relative phase information for detecting human speech and spoofed speech. In *Proc. of Interspeech*, 2015.

Shuai Wang, Yexin Yang, Tianzhe Wang, Yanmin Qian, and Kai Yu. Knowledge distillation for small foot-print deep speaker embedding. In *Proc. of ICASSP*, 2019.

Zhi-Feng Wang, Gang Wei, and Qian-Hua He. Channel pattern noise based playback attack detection algorithm for speaker recognition. In *Proc. of International Conference on Machine Learning and Cybernetics (ICMLC)*, 2011.

Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *Proc. of European Conference on Computer Vision*, 2016.

Wilhelm Wirtinger. Zur formalen theorie der funktionen von mehr komplexen veränderlichen. *Mathematische Annalen*, 97(1):357–375, 1927.

Jared J. Wolf. Efficient acoustic parameters for speaker recognition. *The Journal of the Acoustical Society of America*, 51(6B):2044–2056, 1972.

Zhizheng Wu, Chng Eng Siong, and Haizhou Li. Detecting converted speech and natural speech for anti-spoofing attack in speaker recognition. In *Proc. of Interspeech*, 2012.

Zhizheng Wu, Xiong Xiao, Eng Siong Chng, and Haizhou Li. Synthetic speech detection using temporal modulation feature. In *Proc. of ICASSP*, 2013.

Zhizheng Wu, Nicholas Evans, Tomi Kinnunen, Junichi Yamagishi, Federico Alegre, and Haizhou Li. Spoofing and countermeasures for speaker verification: a survey. *Speech Communication*, 66:130–153, 2015a.

Zhizheng Wu, Tomi Kinnunen, Nicholas Evans, Junichi Yamagishi, Cemal Hanilçi, Md Sahidullah, and Aleksandr Sizov. Asvspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge. In *Proc. of Interspeech*, 2015b.

Zhizheng Wu, Tomi Kinnunen, Nicholas W. D. Evans, Junichi Yamagishi, Cemal Hanilçi, Md. Sahidullah, and Aleksandr Sizov. ASVspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge. In *Proc. of Interspeech*, 2015c.

Zhizheng Wu, Junichi Yamagishi, Tomi Kinnunen, Cemal Hanilci, Mohammed Sahidullah, Aleksandr Sizov, Nicholas Evans, Massimiliano Todisco, and Hector Delgado. ASVspoof: the automatic speaker verification spoofing and countermeasures challenge. *IEEE Journal of Selected Topics in Signal Processing*, 11(4):588–604, 2017.

Xiong Xiao, Xiaohai Tian, Steven Du, Haihua Xu, Eng Siong Chng, and Haizhou Li. Spoofing speech detection using high dimensional magnitude and phase features: The NTU approach for asvspoof 2015 challenge. In *Proc. of Interspeech*, 2015.

Sarthak Yadav and Atul Rai. Learning discriminative features for speaker identification and verification. In *Proc. of Interspeech*, 2018.

Bayya Yegnanarayana, Sharat Reddy, and Prahallad Kishore. Source and system features for speaker recognition using aann models. In *Proc. of ICASSP*, 2001.

Rubén Zazo, Tara N. Sainath, Gabor Simko, and Carolina Parada. Feature learning with raw-waveform CLDNNs for voice activity detection. In *Proc. of Interspeech*, 2016.

Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Proc. of European Conference on Computer Vision*, 2014.

Chunlei Zhang and Kazuhito Koishida. End-to-end text-independent speaker verification with triplet loss on short utterances. In *Proc. of Interspeech*, 2017.

Chunlei Zhang, Chengzhu Yu, and John HL Hansen. An investigation of deep learning frameworks for speaker verification anti-spoofing. *IEEE Journal of Selected Topics in Signal Processing*, 11(4):684–694, 2017.

MUCKENHIRN Hannah

hannah.muckenhirn@idiap.ch

EDUCATIONAL BACKGROUND

2015 – 2019 **Ecole Polytechnique Fédérale de Lausanne – Lausanne, Switzerland**: PhD in Electrical Engineering.

2011 – 2014 **Ecole Polytechnique Fédérale de Lausanne – Lausanne, Switzerland**: Master in Communication Systems.
Distinction: Research Scholars MSc Program (`http://ic.epfl.ch/ResearchScholars`)

WORK EXPERIENCE

2015 – 2019 **Idiap Research Institute – Martigny, Switzerland**: Research Assistant.
Conducting doctoral research on "Trustworthy speaker recognition with minimal prior knowledge using neural networks". The main goal is to develop speaker recognition systems robust to presentation attacks by jointly learning relevant features and classifier from the raw speech signal with deep learning approaches, as well as understand the discriminative information learned by such systems.

2018 (3 months) **Google, Speech team – New York, USA**: Software Engineer Intern.
Conducted research on and implemented a deep neural network-based speaker-conditioned voice filtering system.

2015 (6 months) **Starclay, Data Science start-up – Paris, France**: Data Scientist.
Implemented machine learning algorithms for natural language processing and smart meter data analysis.

2014 (4 months) **Universidad Politecnica de Madrid, Image Processing Group – Madrid, Spain**: Intern.
Designed and implemented image processing algorithms for on-road vehicle detection.

2013 – 2014 (6 months) **IBM Research – Zurich, Switzerland**: Intern (Master thesis).
Modeled, simulated and analyzed the signal processing chain of radio telescopes in the context of the *Square Kilometre Array* project.

2012 – 2013 (1 year) **EPFL, Laboratory of Security and Cryptography – Lausanne, Switzerland**: Research Assistant.
Implemented automated verification of distance-bounding security protocols.

PUBLICATIONS

Journal

• **H. Muckenhirn**, P. Korshunov, M. Magimai.-Doss and S. Marcel, "Long-Term Spectral Statistics for Voice Presentation Attack Detection", *IEEE/ACM Transactions on Audio, Speech and Language Processing, 25(11):2098-2111*, 2017.

Conferences

- Q. Wang*, **H. Muckenhirn***, K. Wilson, P. Sridhar, Z. Wu, J. Hershey, R. A. Saurous, R. J. Weiss, Y. Jia, I. Lopez Moreno, “Voicefilter: Targeted voice separation by speaker-conditioned spectrogram masking”, *Interspeech*, 2019.
- **H. Muckenhirn**, V. Abrol, M. Magimai.-Doss and S. Marcel, “Understanding and Visualizing Raw Waveform-based CNNs ”, *Interspeech*, 2019.
- **H. Muckenhirn**, M. Magimai.-Doss and S. Marcel, “On Learning Vocal Tract System Related Speaker Discriminative Information from Raw Signal Using CNNs”, *Interspeech*, 2018.
- S. H. Kabil, **H. Muckenhirn** and M. Magimai.-Doss, “On Learning to Identify Genders from Raw Speech Signal Using CNNs”, *Interspeech*, 2018.
- **H. Muckenhirn**, M. Magimai.-Doss and S. Marcel, “Towards directly modeling raw speech signal for speaker verification using CNNs”, *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018.
- **H. Muckenhirn**, M. Magimai.-Doss and S. Marcel, “End-to-End Convolutional Neural Network-based Voice Presentation Attack Detection”, *IEEE/IAPR International Joint Conference on Biometrics*, 2017.
- P. Korshunov, S. Marcel, **H. Muckenhirn** et al., ”Overview of BTAS 2016 Speaker Anti-spoofing Competition”, International Conference on Biometrics: Theory, Applications and Systems, 2016.
- **H. Muckenhirn**, M. Magimai.-Doss and S. Marcel, “Presentation Attack Detection Using Long-Term Spectral Statistics for Trustworthy Speaker Verification”, *International Conference of the Biometrics Special Interest Group*, 2016.

SKILLS

Programming	Python, Java, C++, SQL, Torch/PyTorch, TensorFlow
Languages	French (Mother tongue), English (Fluent)