Tampere University

Eetu Hyyryläinen

COMPARISON OF DIFFERENT FEATURES FOR NEURAL NETWORK-BASED MODELS IN SPEAKER IDENTIFICATION

Bachelor of Science thesis
Faculty of Information Technology and Communication Sciences
Examiner: M.Sc. Einari Vaaras
January 2023

ABSTRACT

Eetu Hyyryläinen: Comparison of different features for neural network-based models in speaker identification
Bachelor of Science thesis
Tampere University
Electrical engineering
January 2023

The general understanding in speech and audio processing has been that with larger datasets, task-specific features can be learned straight from a raw audio waveform, but with smaller datasets, traditional signal features, such as log-mel features, are used. The purpose of this work is to compare how different features of speech perform in the case example of the present study, neural network-based speaker identification, with datasets of different sizes. Speaker identification is the task of determining which voice from a group of voices best matches a speaker.

This work consists of two parts. The literature review briefly defines the functioning of a speaker identification system, as well as how speaker identification models have been typically created. In the experimental part of the present study, raw audio and log-mel features were compared with each other in multiple experimental conditions, which included different numbers of training samples and various noise levels.

The results of the present study show that, with no added noise, both features perform very reliably in the task of speaker identification. There was surprisingly not much disparity between the speaker classification accuracies of raw audio-based and log-mel spectrum-based models. Only when the training sample size was small, log-mel features beat raw audio clearly. When the training sample size was large, log-mel features beat raw audio only barely. However, when noise was introduced to the data, raw audio started to outperform log-mel features in the classification accuracy as the noise level increased. With the highest noise level used in the experiments, raw audio was clearly better than log-mel features in classification accuracy, especially with smaller training sampling sizes.

Keywords: speaker identification, speaker recognition, neural networks, log-mel features, raw audio

TIIVISTELMÄ

Eetu Hyyryläinen: Eri piirteiden vertailu neuroverkkopohjaisessa puhujantunnistuksessa
Kandidaatintyö
Tampereen yliopisto
Sähkötekniikka
Tammikuu 2023

Yleinen käsitys puheen- ja äänenkäsittelyssä on ollut, että suuremmilla tietoaineistoilla (*dataset*), tehtäväkohtaiset piirteet voidaan oppia suoraan puheen aaltomuodosta, mutta pienemmillä tietoaineistoilla käytetään perinteisiä signaalitason piirteitä, kuten log-mel -piirteitä. Tämän työn tarkoituksena on verrata puheen eri piirteiden suorituskykyä erikokoisilla tietoaineistoilla tutkimuksen tapausesimerkissä, neuroverkkopohjaisessa puhujantunnistuksessa (*speaker identification*). Puhujantunnistuksessa tehtävänä on määrittää mikä puheääni ryhmästä tunnettuja puheääniä vastaa puhujaa parhaiten.

Tämä työ jakautuu kahteen osaan. Kirjallisuuskatsauksessa määritellään lyhyesti puhujantunnistussysteemin toiminta, sekä kerrotaan miten puhujantunnistusmalleja ollaan tyypillisesti luotu. Tutkimuksen koeosuudessa puheen aaltomuotoa ja log-mel -piirteitä verrattiin keskenään useissa koeolosuhteissa, joihin kuului useampi opetusnäytemäärä, sekä erilaiset kohinatasot.

Tämän tutkimuksen tulokset osoittavat, että ilman lisättyä kohinaa molemmat piirteet toimivat erittäin luotettavasti puhujantunnistuksessa. Puheen aaltomuotoon ja log-mel -piirteisiin perustuvien mallien luokittelutarkkuuksien välillä ei ollut yllättäen selkeää eroa. Vain silloin, kun opetusdatan otoskoko oli pieni, log-mel -piirteet voittivat puheen aaltomuodon selkeästi. Kun otoskoot olivat suurempia, log-mel -piirteet voittivat vain niukasti. Kohinan lisäämisen jälkeen puheen aaltomuoto kuitenkin suoriutui paremmin, kuin log-mel -piirteet. Korkeimmalla kohinatasolla puheen aaltomuoto oli selkeästi parempi luokittelutarkkuudessa verrattuna log-mel -piirteisiin, varsinkin pienimmillä näytemäärillä.

Avainsanat: puhujan identifiointi, puhujantunnistus, neuroverkot, log-mel -piirteet, puheen aaltomuoto

PREFACE

This thesis was written at Tampere University for the bachelor's degree in electrical engineering. I would like to thank my supervisor M.Sc. Einari Vaaras for the valuable help and guidance, and for providing the interesting topic. This project in its interestingness was a further confirmation that I have chosen the right field for myself.

Tampere, 9th January 2023

Eetu Hyyryläinen

CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

CNN	Convolutional neural network
DFT	Discrete Fourier transform
ELU	Exponential linear unit
FLAC	Free lossless audio codec
GMM	Gaussian mixture model
MLP	Multilayer perceptron
ReLU	Rectified linear unit
RMS	Root mean square
SI	Speaker identification
SNR	Signal-to-noise ratio
SV	Speaker verification

1. INTRODUCTION

Speech is the primary way of communication between people, and thus it naturally contains a message in the form of words and utterances. Moreover, speech carries information considering the identity of a speaker, e.g. the anatomy of the vocal folds (Müller 2007). This is an example of a speaker-dependent characteristic, that differs from person to person. Such properties of speech are not essential for understanding the message conveyed in a conversation, but they enable us to recognise a friend over a phone for example. Nevertheless, the fields of speech recognition and speaker recognition contradictorily use the same features in their tasks (Beigi 2011, pp. 143-144). Speaker recognition refers to recognising a speaker based on the characteristics of their voice, but when the speaker is known to the listener or a technological application, this task is referred to as speaker identification (SI) (Beigi 2011).

Features refer to the parametric representations of speech signals. In speech and audio processing literature there is a general understanding considering the performance of certain features in recognition tasks with different dataset sizes. Traditional signal features, such as log-mel features, are thought to perform better with a smaller number of data samples, and with a larger number of data samples the best approach in terms of performance is considered to be learning task-specific features straight from raw audio waveform. The purpose of this thesis is to test this hypothesis by training neural network-based models with log-mel features and raw audio waveforms, and compare their performances with multiple experimental conditions by using SI as a case example.

This thesis is organized as follows. Chapter 2 first briefly defines SI, and then discusses how SI models are typically created. Chapter 3 presents an overview of the experiments in the present study. Additionally, the methods used in the SI task in the present experiments, such as log-mel features and the neural network-based classifiers, are presented. Chapter 4 presents the data used in the present experiments, and the experiments themselves in full detail. The results and observations considering the present experiments are gathered in Chapter 5. Chapter 6 concludes the thesis by drawing conclusions based on the results.

2. LITERATURE REVIEW

As introduced in Chapter 1, SI is the task of determining which voice from a group of known voices matches the speaker best. This is part of a bigger concept called speaker recognition. Speaker recognition is a generic term used for any procedure which involves knowledge of the identity of a person based on their voice (Beigi 2011, p. 5). Speaker recognition is divided into two groups, SI and speaker verification (SV), of which the former is the main focus of this work. SV differs from SI in a way that it performs the task of accepting or rejecting the identity claim. It is a binary classification task where the similarities of a claimed speaker model and a selected impostor model are compared. This is done with a "replay" or "spoofing" attack, where a test utterance from the claimed speaker is re-used in an attempt to fool the system (Müller 2007, pp. 218-219).

Traditionally, SI systems try to model speakers by creating separate speaker models for each speaker using a set of training utterances (Beigi 2011, p. 4). The speaker models are then used in an evaluation phase, where a test utterance is compared to all of the models, after which the closest match is returned. This process is illustrated in Figure 2.1. Depending on whether the test utterance is from the training set or not defines whether the identification task is closed-set or open-set. In closed-set identification, the test speaker

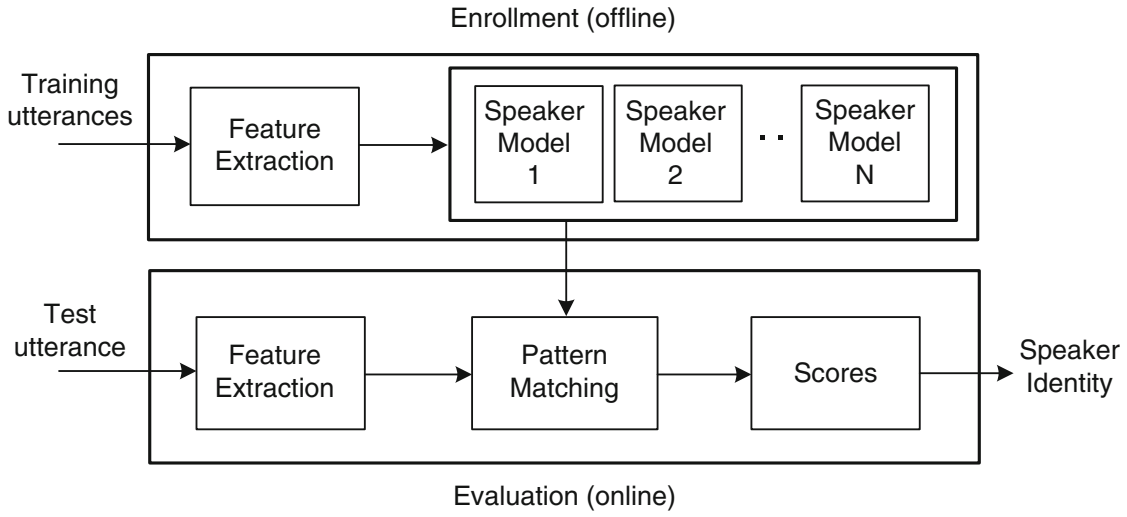

Figure 2.1. *The functioning of speaker identification system. Adapted from (Rao & Sarkar 2014, p. 5).*

is definitely in the training set (Beigi 2011, p. 548), but in open-set identification this is not necessarily the case. This literature review specifically discusses closed-set speaker identification, because the experiments of the present study are based on that.

An essential part of speaker recognition systems is extracting vectors of features from time-domain sampled acoustic waveforms. The features can be categorized as 'low-level', which convey physiological information, such as the size of the vocal folds of a given speaker. The features can also be 'high-level', which in turn reflect behavioral aspects of the speaker, such as temperament or accent (Rao & Sarkar 2014, p. 3). Feature extraction begins with waveforms being divided into overlapping segments called frames, typically 20 ms to 30 ms in duration (Togneri & Pullella, 2011). Next, the frames are multiplied by a window function, which performs tapering at the beginning and end edges of the frames. This prepares the frame for the next stage, where feature transformations are usually applied. Typically, a Fourier transform is applied to the audio frames to produce spectral-level features, as SI-related phenomena are spectral in nature.

SI can be described as a multiclass classification problem, and many methods have been used for accomplishing this task. Very widely used approaches have been nearest neighbor, vector quantization, neural networks, and binary trees (Müller 2007, p. 219). Reynolds and Rose (1995) compared the use of Gaussian mixture models (GMMs) in text-independent SI to other techniques, such as unimodal gaussian classifier and veqtor quantization codebook, and found out that GMM outperforms all of them. According to Chakroun and Mondher (2020), most state-of-the-art speaker recognition applications have used GMMs. However, in case of limited training data and short test utterances, GMMs do not achieve high accuracy. This has lead to developement of new techniques such as the i-vector, which is a feature extraction method specifically designed for speaker identification (Chakroun & Mondher, 2020). The current state-of-the-art method in SI utilizies i-vector based approach using a GMM based universal background model (Tiwari et al. 2019). Deep learning-based models have also been used for this task. Bunrit et al. (2019) compared a convolutional neural network (CNN) trained with spectrogram images to a support vector machine classifier which used MFCC features. The study resulted in the CNN being the better classifier in text-independent speaker identification with a 95.83% accuracy.

3. METHODS

In this chapter, all the methods that were used in the present SI experiments are presented. The objective of the present study was to compare different audio features for neural network-based models by using SI. The used features were raw audio and log-mel features, which are discussed more in-depth in Sections 3.1 and 3.2, respectively. A deep learning-based approach was chosen for the SI system of the present experiments; as seen in an overview of the two models in Figure 3.1, multilayer perceptrons (MLPs) and CNNs were used. These methods are discussed further in Sections 3.3 and 3.4, respectively. The data used in training of these models is labeled, and thus their learning processes can be called supervised learning (Alpaydin & Bach 2014, p. 21).

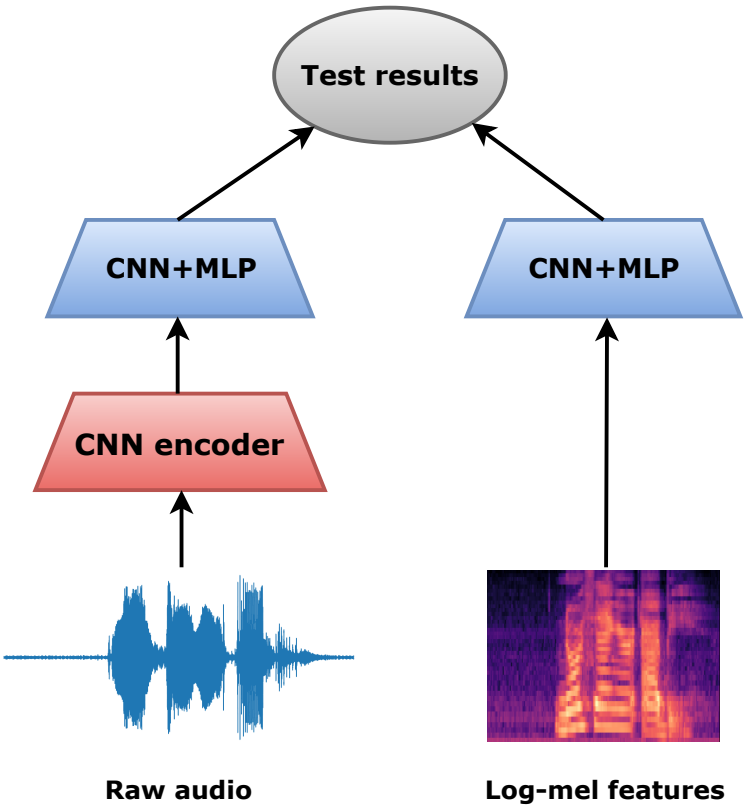

Figure 3.1. *The general idea behind the present experiments. A CNN encoder changes dimensionality of digital acoustic waveform from 1D to 2D. To see which feature performs better in the present task, a classifier consisting of a CNN and a MLP is trained for both cases, using either encoded raw audio or log-mel features.*

In order to make raw audio and log-mel features comparable, the raw audio is first converted into a similar dimensionality as the log-mel features using a CNN encoder. Then, these converted raw audio features and the log-mel features are used as an input to a classifier consisting of CNN and MLP layers in order to determine which features perform better in the SI task.

3.1 Acoustic waveform

Speech is a non-stationary acoustic signal, which means its parameters change over time (Beigi 2011, p. 76). Speech signal is observed as a continuous function $s(t)$ where t is a continuous variable representing time. In Figure 3.2 is a digital representation of an acoustic waveform of this kind. To analyze a continuous-time waveform in digital systems, the waveform has to be sampled with some sampling rate, which transforms the waveform to discrete-time signal. After this the waveform is a sequence of numbers $s(n)$, where n is an index indicating time. The final step in digitizing the signal is quantization, which is done by an analog-to-digital converter (Tohyama & Tsunehiko, 1998). This is the format in which audio files, such as the FLAC files used in the present experiments are in, and they are referred to as raw audio.

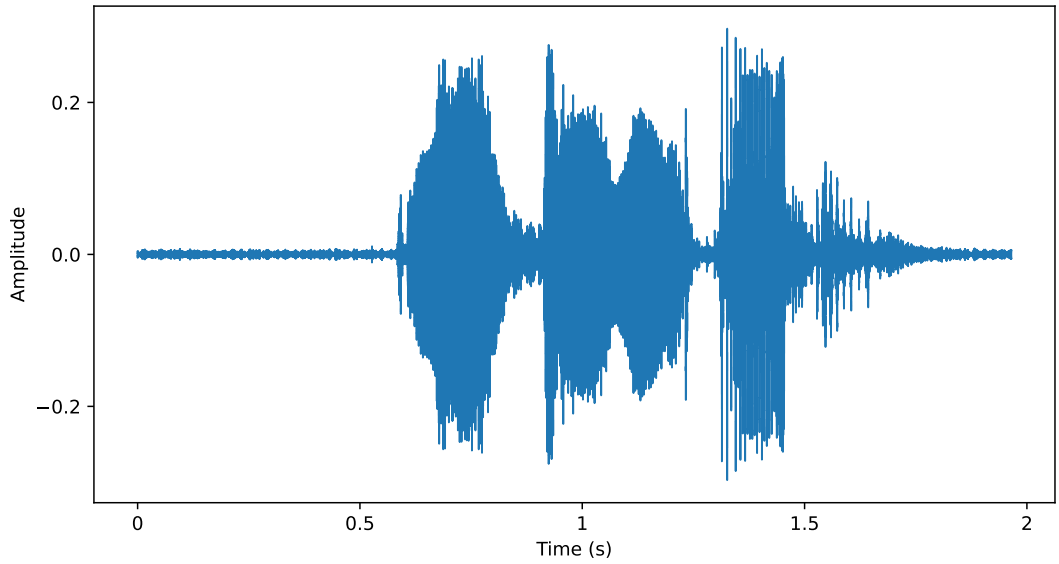

Figure 3.2. *A digital representation of an utterance containing the words "Northanger Abbey".*

3.2 Log-mel spectrum

Mel scale (abbreviation from the word melody) is a way to model how humans perceive sound in a non-linear fashion: in case of equally spaced frequencies, higher frequencies are perceived closer to each other than lower frequencies (Beigi 2011, p. 147). To obtain a mel spectrogram, windowed discrete Fourier transformation (DFT) is calculated first for

each frame:

$$F_k = \sum_{n=0}^{N-1} w_n s_n e^{-\frac{2\pi i n k}{N}} \quad , \tag{3.1}$$

where $k = 0, ..., N-1$. Next, the absolute value of the frequency-domain signal F_k is transformed to mel scale by applying a set of triangular band pass filters seen in Figure 3.3. The center frequencies of these filters are equally spaced on the mel scale, whose transformation function is defined as

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

where f is frequency in hertz. The triangle shaped filters in Figure 3.3 start and end at the center frequencies of their neighbours. A common choice in speech recognition are 24 of these filters (Müller 2007, pp. 229-230). In the present study, the chosen number for the filters was 40.

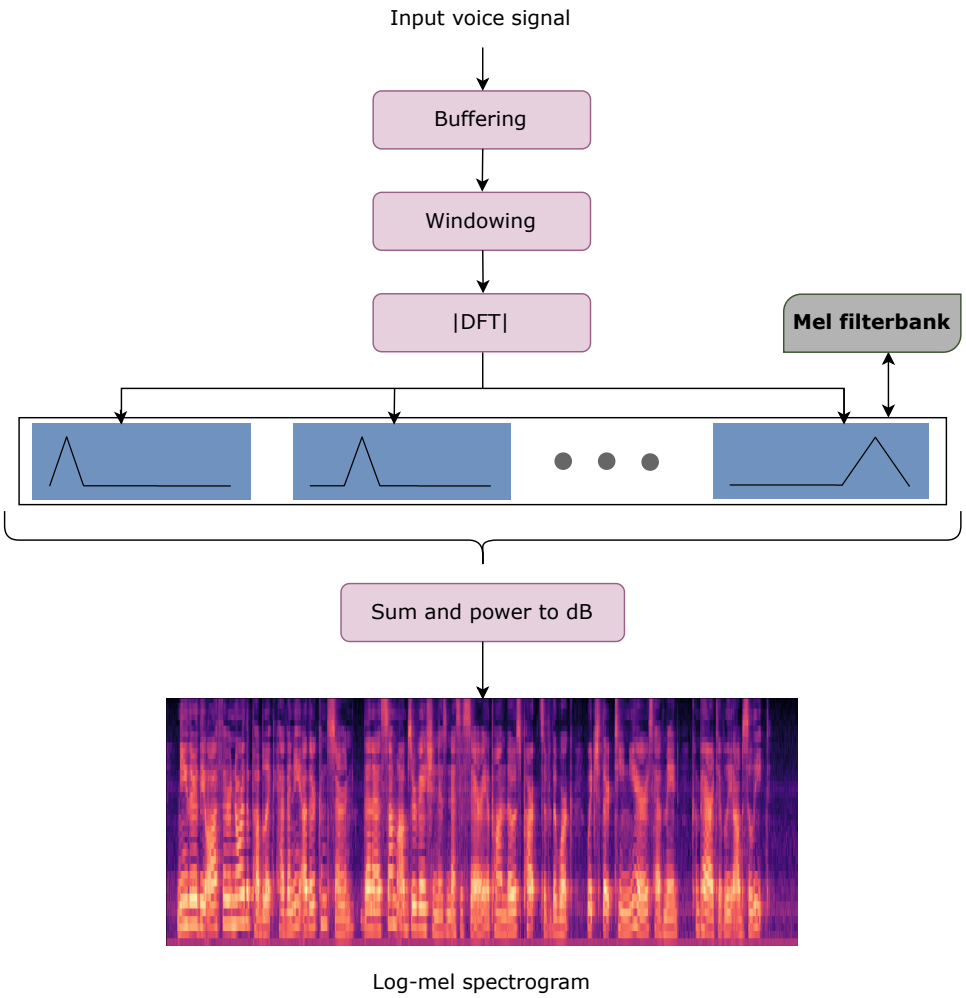

Figure 3.3. *The process of log-mel spectrogram feature extraction. After the DFT, a set of band-pass filters that make up the mel filterbank are applied to obtain the mel spectrum.*

After the frequencies in hertz are transformed to mel frequencies in mels, the signal can be represented as a two-dimensional log-mel spectrogram. As its name suggests, it is a logarithmic mel spectogram. Every step involved in the extraction of the log-mel spectrogram from the signal representation of speech is depicted in Figure 3.3. Log-mel features have been found to be very practical in deep learning-based audio tasks, since they represent audio in a two-dimensional form. This suits well with e.g. CNNs that have been found to work especially well with two-dimensional inputs, such as images (Palanisamy et al. 2020).

3.3 Multilayer perceptron

A multilayer perceptron is an artificial neural network model, which consists of an input layer, at least one hidden layer and an output layer. Its neurons of layers are either fully or partially connected to the ones in the next layers, depending on the architecture (Shanmuganathan & Samarasinghe 2016, p. 6). This type of neural network is used for regression and classification tasks (Alpaydin & Bach 2014, p. 267). An MLP is composed of a number of parallel perceptrons in a number of layers depending on the number of hidden layers. A perceptron is its basic processing element. The output of a perceptron is a weighted sum of its inputs, which may come from other neurons, or from the input layer. A single perceptron defines a hyperplane. When there are K parallel perceptrons, a single output, o_i, is defined as a matrix multiplication

$$o_i = \mathbf{w}_i^T \mathbf{x} \quad , \tag{3.3}$$

where $\mathbf{w}$ is a vector of weights, $\mathbf{x}$ is a vector of inputs and $i = 1, ..., K$ (Alpaydin & Bach 2014, pp. 271-273). Usually this output is then scaled with some activation function. Some of the most used activation functions include sigmoid, hyperbolic tangent and rectified linear unit (ReLU) activation function (Michelucci 2018). The activation function used in this study for MLP is exponential linear unit function (ELU), which is defined as

$$f(x) = \begin{cases} \alpha(e^x - 1), & \text{when } x < 0 \\ x, & \text{when } x \geq 0 \end{cases} \quad , \tag{3.4}$$

where α should be a positive number. Usually in the output layer, a softmax function is used. This function takes every output from the previous layer as parameters, and calculates the estimated class probabilities for them (Alpaydin & Bach 2014, p. 273).

3.4 Convolutional neural network

Convolutional neural networks (CNNs) have been inspired by the visual cortex of animals (Géron 2018). They are neural networks that use convolution in place of matrix multiplication. Convolutions are often applied to two-dimensional data, such as images. For a two-dimensional image I, the operation is defined as follows:

$$(I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n) \quad , \tag{3.5}$$

where K is a two-dimensional kernel (Goodfellow et al. 2016). A kernel is a window with a set of weights, which are learned for every convolutional layer in a CNN during training. The kernel slides across the input image, where it is a receptive field of a neuron in the next convolutional layer. Kernels can have different stride values, which reduces the dimensionality of the next feature map. As can be seen in Figure 3.4, convolutional layers consist of multiple stacked feature maps of equal sizes, which makes detecting multiple features possible in a single convolutional layer, as each feature map is responsible for detecting some specific feature. Every convolutional layer is generally followed by an activation function layer, where ReLU is usually used (Géron 2018).

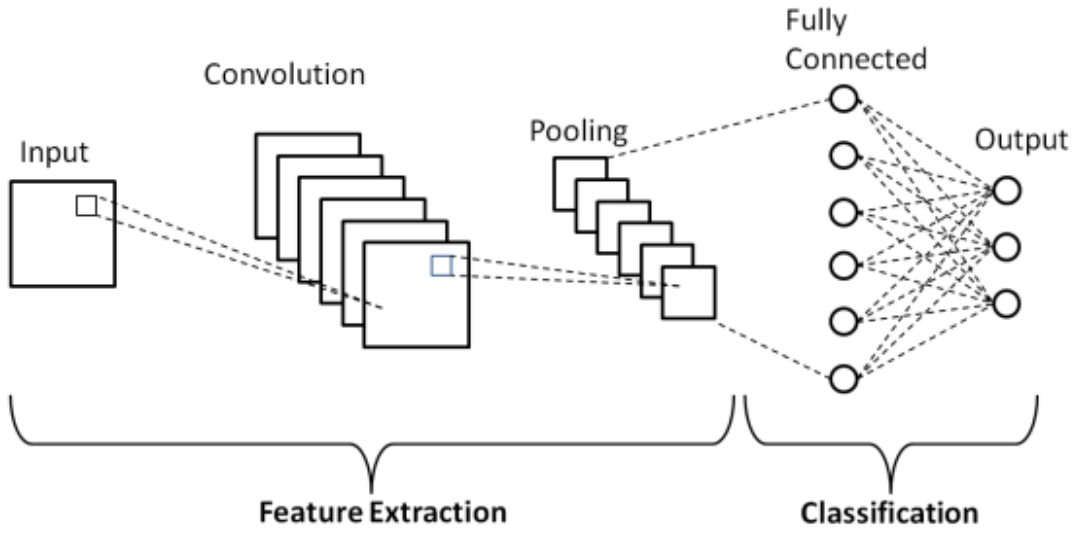

Figure 3.4. *A typical CNN architecture. Adapted from (Phung & Rhee 2019).*

Pooling layers are very similar to convolutional layers, but their goal is to subsample the input. One reason for this is to limit the risk of overfitting. Instead of weights, a pooling neuron uses an aggregation function, such as max, which picks the maximum value inside a pooling kernel (Géron 2018).

Typical CNN architectures use one or more convolutional layers with ReLU layers in between them, followed by a pooling layer, followed by more layers with a similar structure. This pattern of layers goes on an arbitrary number of times, depending on the complexity

of the architecture. Finally, one or more fully-connected layers (Section 3.3) are added after the convolutional part of the network (Géron 2018). A typical CNN architecture is depicted in Figure 3.4 in a simplified manner. This type of architecture, where the final few layers are fully connected, is used in this study.

4. EXPERIMENTS

In this chapter, the neural network-based SI experiments of the present study are presented in full detail. This includes e.g. the data used in the present experiments and the details of the neural network-based classifiers.

4.1 LibriSpeech dataset

The data used in the SI experiments in this study is from an automatic speech recognition corpus of read English speech called LibriSpeech (Panayotov et al. 2015). It was specifically created for training and evaluating speech recognition systems, and it contains over 1000 hours of speech sampled at 16 kHz. The speech is derived from audiobooks, whose speakers are partitioned by using speaker diarization.

In the dataset, the speech is in a form of digital audio as short FLAC files, some longer than others. Each of these files contains an utterance from a specific speaker. The dataset also has corresponding text for each utterance to be used by text dependent speech recognition tasks. LibriSpeech dataset is divided into different-sized subsets. For this study, a subset called train-clean-100 was used, which contains 100 hours of gender-balanced speech. In more detail, this subset has 251 total speakers, 29 124 utterances, and average utterance length is approximately 12 seconds. Furthermore, the maximum and minimum utterance lengths are approximately 24.5 seconds and 1.4 seconds, respectively.

4.2 Experimental setup

The first step before the experiments was to discard some speakers from the dataset. These were the speakers who in total had less than 20 minutes of speech. This was done so that each speaker would have approximately the same amount of speech, as most speakers had 20–25 minutes of speech. As a result, the dataset size decreased to 231 speakers and 27 445 utterances.

The SI system illustrated in Figure 3.1 was implemented using PyTorch, which is a deep learning library for Python programs (Paszke et al. 2020). The code used in this project is publicly available in GitHub: (https://github.com/eetuhoo/pytorch-speaker-identification).

The data was randomly split into train, validation and test sets in a ratio of 60:20:20. The validation set was used to stop the training if the loss value of the validation set did not improve for 30 epochs. Next, the data was extracted to vectors of raw audio or log-mel features, depending on the experiment. These features were extracted from a randomly taken segment of 1.4 seconds length from every utterance in the dataset. This segment length was chosen so that every utterance could be included in the present experiments, as the shortest utterance of the dataset was of this length. Furthermore, the segment length was defined as 1.4 seconds since this can be considered as a reasonable utterance length for SI systems.

For raw audio data, a CNN-based encoder was used to transform the raw audio into two-dimensional features. The architecture of this CNN encoder was the same five-layer CNN as the popular audio encoder used in e.g. van den Oord et al. (2019). The CNN classifier model was identical for log-mel features and the encoded raw audio, consisting of three 2D-convolution layers followed by three fully-connected layers. The convolutional layers had a square kernel of shape 3 and output channels {64, 64, 64}, each layer followed by batch normalization, a ReLU nonlinearity, maxpooling with kernel sizes {(4,2), (5,2), (7,2)}, and a 10% dropout. The output of the last convolutional layer was then flattened and fed to three fully-connected layers with 320, 256 and 231 units. Each fully-connected layer was followed by an ELU nonlinearity, and every fully-connected layer had a 10% dropout. In order for the model to output a probability for each one of the 231 speakers, a softmax function was applied after the last model layer.

The models were trained with three different data sampling rates: 100%, 50% and 25%. The data sampling was performed randomly and its purpose was to simulate different-sized datasets being used for training the models. For the training process in every experiment, a batch size of 32 was used, and cross-entropy loss was used as the loss function. Adam optimizer was used to optimize the parameters of the neural networks. After training the models, the performance of the models was tested using the test set. It is worth noting that when extracting features from the utterances of the test set, the 1.4 second segment was taken from a specific location. This was done to make the results comparable with each other.

The experiments were also conducted with different levels of noise to simulate the performance of the different features in noisy conditions. In this second part of the experiments, noise was added at three signal-to-noise ratios (SNR): low (20 dB), medium (10 dB) and high (0 dB). For each noise level, the same three data sampling sizes were used as in the first part. The used noise model was additive white Gaussian noise model, and so the noise was randomly sampled from a zero mean Gaussian distribution. Noise was added

to the audio files using the following equation for SNR:

$$SNR = 10 \cdot log_{10}\left(\frac{RMS_{signal}^2}{RMS_{noise}^2}\right) \quad , \tag{4.1}$$

where RMS_{signal} and RMS_{noise} are the root mean square values of signal and noise, respectively. This equation can be presented with respect to the RMS_{noise} value:

$$RMS_{noise} = \sqrt{\frac{RMS_{signal}^2}{10^{SNR/10}}} \quad . \tag{4.2}$$

As the RMS value of the noise is equal to standard deviation with mean zero, Equation 4.2 can be used to sample noise at different SNR values (Sherman & Butler 2007).

5. RESULTS

The testing accuracies of the classification experiments are compiled in Table 5.1. The results are ordered according to the noise levels, which in turn are ordered from no noise to high noise level.

Table 5.1. *Classification accuracies for the two features in the experiments. Three different data sampling rates were used for utterances with three different added noise levels, and for utterances with no noise at all. SNR is 20 dB for low noise, 10 dB for medium noise and 0 dB for high noise level.*

Data sampling	Noise (SNR)	Accuracy (Raw audio)	Accuracy (Log-mel)
100%	None	0.9806	0.9923
50%	None	0.9581	0.9864
25%	None	0.8501	0.9705
100%	20 dB	0.9624	0.9666
50%	20 dB	0.9285	0.9304
25%	20 dB	0.8755	0.8954
100%	10 dB	0.9314	0.9248
50%	10 dB	0.8704	0.8653
25%	10 dB	0.8243	0.8164
100%	0 dB	0.8922	0.8443
50%	0 dB	0.8652	0.7044
25%	0 dB	0.7583	0.6087

Starting from the clean utterances with no added noise, log-mel features performed better than raw audio in all the cases with different sampling sizes. Nevertheless, raw audio performed very reliably as well. Both features achieved almost 100% classification accuracy using 100% sampling size. Going down in sampling size, raw audio drops faster in accuracy than log-mel features, whose performance stays relatively consistent with all sampling sizes. When using sampling size of 25%, raw audio drops relatively low in accuracy with $\sim$85% compared to log-mel features. This goes along with the hypothesis, that with smaller data sampling sizes traditional signal features, such as log-mel features, perform better.

A similar effect can be seen with different data sampling sizes, when using utterances with added noise; as sampling size rises, so does classification accuracy. Log-mel features also beat raw audio, when noise level was low, but only slightly. The turning point, where raw audio started to beat log-mel features was when the noise level was medium. Here the difference is still subtle, but when the noise level was high, the accuracy with raw audio was clearly better than with log-mel features. This can be seen especially with 50% and 25% data sampling sizes.

Training durations also varied in the present experiments. Generally, training with raw audio was slower when measuring in the amount of training epochs. For models trained with raw audio, the training process took approximately 30–70 training epochs longer than training models with log-mel features. The only exception for this was when using data sampling size of 25%, where the training stopped earlier for raw audio. This was because the neural network was unable to get the loss value under 0.4, as opposed to training with log-mel features, where the loss value was approximately 0.09. For 50% and 100% sampling sizes, the loss difference between the two features was in range of 0.03–0.1.

6. CONCLUSION

In speech and audio processing, the size of a dataset is considered to play a role in the performance of different signal features. Traditional signal features have been thought to outperform raw audio waveform with smaller datasets, and vice versa with larger datasets. However, it is unclear when one starts to beat the other. This thesis was made to compare different features of speech for neural network-based classifiers, using speaker identification as a case example. In this thesis, the log-mel spectrum was chosen as a traditional signal feature, to be compared against raw audio waveform.

This study showed that in speaker identification identification, both the log-mel features and the raw audio performed very reliably with data sampling sizes of 100%, 50% and 25%. A surprising result of the study was that log-mel features beat raw audio also with large sampling sizes. Only when simulating noisy conditions, not only the classification accuracy started to naturally drop, but raw audio started to perform better in relation to log-mel features.

With a bigger dataset, raw audio could possibly overtake log-mel features in the classification accuracy with large data sampling sizes. However, as seen from the results, the accuracies for both features are very close to 100% already, so the difference would be minimal. Another element, that can possibly affect the classification results is the dimensionality of the encoded raw audio. The dimensionality was purposefully set to be equal to the dimensionality of log-mel features, so that the results could be as comparable as possible. However, the encoded raw audio could as well have a higher dimensionality than the log-mel features. Furthermore, the encoder that was used to change the dimensionality of raw audio increases the computing capacity of the architecture used for raw audio. The log-mel features did not have a separate encoder, and therefore the computing capacity for the log-mel features was also smaller. This can be significant when adding fixed noise levels to the signal, as the encoder can learn to eliminate noise, and this in turn can explain the better performance of raw audio relative to log-mel features with noisy signals. Furthermore, the experiments conducted with noisy signals could have been more realistic if the noise levels would have alternated during the training and testing processes, since in realistic scenarios the level of noise is very rarely fixed.

REFERENCES

Alpaydin E. and Bach F. (2014). *Introduction to machine learning*. MIT Press, Cambridge, Massachusetts.

Beigi H. (2011). *Fundamentals of speaker recognition*. Springer US, Yorktown Heights.

Bunrit S., Inkian T., Kerdprasop N. and Kerdprasop K. (2019). Text-independent speaker identification using deep learning model of convolution neural network. *International journal of machine learning and computing*, Vol.9 (2), pp. 143-148.

Chakroun R. and Mondher F. (2020). Robust features for text-independent speaker recognition with short utterances. *Neural computing & applications*, Vol.32 (17), pp. 13863-13883.

Géron A. (2018). *Neural networks and deep learning*. O'Reilly Media, Inc, Beijing.

Goodfellow I., Bengio Y. and Courville A. (2016). *Deep Learning*. MIT Press, London.

Michelucci U. (2018). A Single Neuron in *Applied Deep Learning: A Case-Based Approach to Understanding Deep Neural Networks*. Apress, Berkeley, CA.

Müller C. (2007). *Speaker classification I: fundamentals, features and methods*. Springer, New York.

Palanisamy K., Singhania D., Yao A. (2020). *Rethinking CNN Models for Audio Classification*. arXiv preprint arXiv:2007.11154, Cornell University Library, Ithaca.

Panayotov V., Chen G., Povey D. and Khudanpur S. (2015). Librispeech: An ASR corpus based on public domain audio books. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206-5210.

Paszke A., Gross S., Massa F., Lerer A., Bradbury J., Chanan G., Killeen T., Lin Z., Gimelshein N., Antiga L., Desmaison A., Köpf A., Yang E., DeVito Z., Raison M., Tejani A., Chilamkurthy S., Steiner B., Lu F., Bai J. and Chintala S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*, Vol.32.

Phung V. H. and Rhee E. J. (2019). A High-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets. *Applied sciences*, Vol.9, p. 4500.

Rao K. S. and Sarkar S. (2014). *Robust Speaker Recognition in Noisy Environments*. Springer International Publishing AG, Cham.

Reynolds D. A. and Rose R. C. (1995). Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE transactions on speech and audio processing*, Vol.3 (1), pp. 72-83.

Sherman C. H. and Butler J. L. (2007). *Transducers and Arrays for Underwater Sound*. Springer, New York.

Tiwari V., Hashmi M. F., Keskar A. and Shivaprakash N. C. (2019). Speaker identification using multi-modal i-vector approach for varying length speech in voice interactive systems. *Cognitive systems research*, Vol.57, pp. 66-77.

Togneri R. and Pullella D. (2011). An Overview of Speaker Identification: Accuracy and Robustness Issues. *IEEE Circuits and Systems Magazine*, vol.11 (2), pp. 23-61.

Tohyama M. and Tsunehiko K. (1998). Discrete Representation of Signals in *Fundamentals of acoustic signal processing*. Academic Press, United States.

van den Oord A., Li Y. and Vinyals O. (2019). *Representation Learning with Contrastive Predictive Coding*. arXiv preprint arXiv:1807.03748, Cornell University Library, Ithaca.