Zurich University
of Applied Sciences

School of
Engineering
InIT Institute of Applied
Information Technology

Bachelor thesis (Computer Science)

Speaker Recognition with Few Training Samples

Authors	Alicia Lea Rüegg Arik Sidney Guggenheim
Main supervisor	Mark Cieliebak

Date	19.06.2020

Zurich University
of Applied Sciences

School of
Engineering

DECLARATION OF ORIGINALITY
Bachelor's Thesis at the School of Engineering

By submitting this Bachelor's thesis, the undersigned student confirms that this thesis is his/her own work and was written without the help of a third party. (Group works: the performance of the other group members are not considered as third party).

The student declares that all sources in the text (including Internet pages) and appendices have been correctly disclosed. This means that there has been no plagiarism, i.e. no sections of the Bachelor thesis have been partially or wholly taken from other texts and represented as the student's own work or included without being correctly referenced.

Any misconduct will be dealt with according to paragraphs 39 and 40 of the General Academic Regulations for Bachelor's and Master's Degree courses at the Zurich University of Applied Sciences (Rahmenprüfungsordnung ZHAW (RPO)) and subject to the provisions for disciplinary action stipulated in the University regulations.

City, Date:	**Name Student:**
Zurich, 17. June 2020	Alicia Lea Rüegg
Zurich, 17. June 2020	Arik Sidney Guggenheim

Abstract

Transcribing audio recordings of interviews or meetings manually is time-consuming. An automated system could therefore help to speed up the process. When creating a conversation transcription system, the task of speaker diarization (SD) is one of the most challenging ones.

In this thesis we present a solution, which uses a speaker identification (SI) system based on a convolutional neural network (CNN) developed by the Visual Geometry Group (VGG). In order to tailor the system for SD it was solely trained on audio data.

Short, a-priori voice samples of each speaker are used to adapt the system to a new conversation instantly. Self-designed heuristics support the system on how the results from the identification part should be interpreted. To assess the best fitting parameters, Verbmobil II, a speech corpus, was used. Verification of these parameters was done with the LibriSpeech corpus to enable comparison with other state-of-the art speaker recognition (SR) projects.

We determined that speech samples of 25 seconds are sufficient to minimize the diarization error rate (DER). On the LibriSpeech corpus, the system achieved a DER of 1.7% and an accuracy of 95.7% with two speakers. For five speakers, a DER of 4.8% and an accuracy of 91.8% was reached.

Zusammenfassung

Das Transkribieren von Interviews und Dialogen ist zeitaufwändig. Ein automatisiertes System könnte daher helfen, den Prozess zu beschleunigen. Bei der Entwicklung eines solchen Systems ist die Aufgabe der Sprechererkennung (”speaker diarization”) eine der grössten Herausforderungen.

In dieser Arbeit wird eine Lösung präsentiert, welche ein System zur Identifikation von Sprechern (”speaker identification”) verwendet. Die Basis bildet ein Convolutional Neural Network (CNN), welches auf Audiodaten trainiert wurde. Dieses neuronale Netz wurde von der Visual Geometry Group entwickelt und trägt daher den Namen VGG.

Kurze, a-priori Stimmproben jedes Sprechers wurden verwendet, um das System dynamisch an ein neues Gespräch anzupassen. Heuristiken unterstützen bei der Interpretation der Ergebnisse. Der Sprachkorpus Verbmobil II wurde verwendet, um die optimalen Parameter unseres Systems zu ermitteln. Um einen Vergleich mit anderen aktuellen Sprecher-Erkennungs-Systemen zu ermöglichen, wurde die Verifikation mit dem LibriSpeech Korpus durchgeführt.

Es wurde festgestellt, dass Sprecherproben von 25 Sekunden ausreichen, um die Fehlerrate der Sprechererkennung (”DER”) zu minimieren. Auf dem Korpus LibriSpeech erreichte das System mit zwei Sprechern eine DER von 1.7% und eine Genauigkeit (”accuracy”) von 95.7%. Bei fünf Sprechern wurde eine DER von 4.8% und eine Genauigkeit von 91.8% erreicht.

Preamble

We would like to give special thanks to:

- Prof. Dr. Mark Cieliebak for proposing the topic of this thesis. He was a great mentor and always pointed us in the right direction. We additionally thank him for his intensive support and expertise.
- Anna Ulasik for providing us well-prepared speech corpora.
- Philippe Schläpfer for the weekly support and for issuing us additional information about the evaluation of our system.
- Institute of Applied Information Technology (InIT) for providing us with computation resources on their GPU-cluster.
- Our employers mimacom ag and Digitec Galaxus AG for their flexibility and understanding while working on this project.
- Our friends and family for their constant support and their talent to always cover our backs in difficult situations.

Contents

1 Introduction **1**

1.1 Related Work 2

1.2 Main Objective 4

1.3 Sub-Objectives 4

1.3.1 Host System on Own Infrastructure 4

1.3.2 Keep Error Rate as Low as Possible 5

1.4 Secondary Objectives 5

1.4.1 Create Voice Database 5

1.5 Thesis Overview 5

2 Theoretical Background **7**

2.1 Speaker Identification 7

2.1.1 Log-Mel Spectrogram 7

2.1.2 Convolutional Neural Network 8

2.1.3 Visual Geometry Group Model 8

2.1.4 VGGish 8

2.1.5 Classification 10

2.2 Evaluation of the System 11

2.2.1 Accuracy 11

2.2.2 Diarization Error Rate 12

3 Approach and Methods **13**

3.1 Data Preprocessing 13

3.1.1 Voice Activity Detection 14

3.1.2 Sliding Chunks Generation 15

3.1.3 Spectrogram Generation 16

3.2 Speaker Recognition 16

3.2.1 Feature Extraction 16

3.2.2 Classification 17

3.3 Post-processing 17

3.3.1 Apply Heuristics 18

3.3.2 Prepare Result 18

3.4 Experiments 19

3.4.1 Parameters 19

3.4.2 Training and Testing 20

4 Results and Discussion **23**

4.1 Choosing Appropriate Parameters 23

4.1.1 Choosing an Appropriate Classifier 23

4.1.2 Choosing an Appropriate Training Data Duration 26

4.1.3 Choosing an Appropriate Chunk Size 26

4.1.4 Choosing an Appropriate Skip Factor 27

4.1.5 Summary of the Findings on the Parameters . . 28

4.2 Evaluation on Different Settings 29

4.3 Evaluation of the Outliers 30

4.3.1 Evaluation of the Best Performers 30

4.3.2 Evaluation of the Worst Performers 31

4.4 Evaluation on a Different Corpus 33

5 Conclusion **35**

5.1 Comparison with Initial Objective 36

5.2 Significance of this Thesis 36

5.3 Questions Arising 37

5.3.1 Voice Activity Detection 37

5.4 Comparison to Other Systems 37

5.5 Directions for Future Research 38

6 Registers **40**

6.1 Bibliography 40

6.2 Glossary 45

6.3 List of Figures 47

6.4 List of Tables 48

6.5 Abbreviations 49

7 Appendix **51**

7.1 Technical Instructions 51

7.2 Result Structure 54

7.3 Additional Content 55

7.4 Initial Problem Description 56

7.5 Project Management 58

Chapter 1

Introduction

Nowadays, automation moves forward at an ever-faster pace. However, transcribing audio recordings is still often done by hand. Automated transcription systems are able to take over this task, but they face several problems during this procedure.

One challenge of automatic text transcription is to identify who is currently speaking. If no preliminary information about the speakers is present, a speaker diarization (SD) system is used to detect speaker changes within an audio file and to identify, when the same speaker resumes speaking.

A simplification of a SD system is a speaker identification (SI) system. A SI system finds the actual speaker among a list of predefined speakers. Therefore, initial voice recordings of each user have to be present. Speaker verification (SV) is a further simplification of SI and verifies whether a given utterance fits a claimed identity or not. Speaker recognition (SR) is the umbrella term for SV, SI and SD, with the terms set in order of increasing complexity.

On this background, it will be evaluated in this thesis, how a SD system can be simplified by a SI system, when short audio recordings

of each participant are available as a-priori information.

1.1 Related Work

If an audio stream with multiple speakers is divided into homogeneous segments (speaker change detection) and assigned to these speakers (speaker clustering), it is called SD. As shown by Tranter and Reynolds [1], SD systems have so far been used for different types of audio, which differ not only in their quality of recordings but also in the number of speakers, the duration and sequencing of speaker turns.

Modern SD solutions follow different methods. SD with unsupervised i-vector clustering proposed by Dehak et al. [2] has gained great attention in recent years due to its outperforming results. I-vector architectures recommend a process, where vectors are extracted from short excerpts of speech and are organized into speaker clusters [3]. These vectors have the characteristics of reducing the dimensions to a fixed-length feature vector while preserving the relevant information.

Neural networks are state-of-the-art in many classification problems, especially image classification. Deep neural networks (DNN) can be used for classification (direct method) or for feature extraction that is then used to train a secondary classifier (indirect method) [4]. The indirect method allows the DNN to transfer knowledge gained from one problem where a lot of labelled data is present and then reuse it to solve a similar problem. This technique is called transfer learning.

Nowadays, Neural Networks (NN) are not only used in standard image classification problems, they are also seen as an appropriate alternative for audio classification using i-vectors. To visualize audio

streams for DNNs, spectrograms are typically extracted from audio segments [4]. In-depth explanation on spectrograms will be discussed in section 2.1.1.

Convolutional Neural Networks (CNN), a subcategory of DNNs, are frequently used in SR systems. However, their application has been evaluated especially in image classification, where they are proven to be very effective [5]. CNNs like AlexNet [5], Inception [6], ResNet [7] and VGG [8] emerged from intense investigation. After the publication of large image repositories, such as ImageNet [9], they experienced an enormous upturn due to the now available amount of training data. Research has shown that CNNs, which are used for image classification, also perform well in audio classification tasks [10]. Similarly to image classification, the provision of large scale audio datasets like VoxCeleb [11] or LibriSpeech [12] have also given this research area a boost.

The use of VGG, a model developed by the Visual Geometry Group from the University of Oxford, for SI has already been evaluated in various forms and is also used in this thesis (section 2.1.3). In 2016, Eghbal-Zadeh et al. [13] participated in the DCASE2016 challenge, a challenge covering the detection and classification of acoustic scenes and events. They won the challenge by using a hybrid system with multi-channel i-vectors and VGG.

In 2018, Vélez, Rascon and Fuentes-Pineda [14] proposed a Siamese CNN architecture in the context of one-shot SI. The CNN takes over the task of extracting proper audio features and a classifier determines the similarity between these features. The VGG architecture, which was trained on VoxCeleb [10] or LibriSpeech [11], was among the top three performing models of their research.

1.2 Main Objective

The primary objective of this thesis is to implement a SI system which can be trained and adapted to new voices in near real-time. As few utterances as possible from each speaker should be used to tailor the system to the given conversation. One investigated solution will be implemented and applied to a chosen speech corpus to assess its feasibility and properties.

The initial problem description can be found in the appendix under section 7.4.

1.3 Sub-Objectives

In addition to the main goal described in the section above, there are several sub-objectives which will be addressed in this document.

1.3.1 Host System on Own Infrastructure

The designed solution would ideally be able to run on private infrastructure. There are several reasons why this can be important:

- No dependence on a third party provider is created. The system won't be affected by interruptions or other changes in an external system.
- No additional costs are incurred through the use of external services.
- Sensitive data that may be present in the conversation will not be passed to third parties.

1.3.2 Keep Error Rate as Low as Possible

The key metric to evaluate the system is the diarization error rate (DER). This measurement is described in a more detailed manner in section 2.2.2. The DER of 7.6% from Google's fully supervised speaker diarization approach by Zhang et al. [15] is taken as a reference for the system to develop.

1.4 Secondary Objectives

Depending on the remaining time and progress on the main objectives, a stretched objective is defined.

1.4.1 Create Voice Database

With the solution described above, the system can only use short audio fragments to learn the voice of a speaker. This secondary objective defines the next step to further lower the error rate and improve the model's overall accuracy by creating a voice database. This database contains verified voice chunks for each speaker from previous conversations. This data can be re-used in the case that the same person attends a further discussion.

1.5 Thesis Overview

This section provides a brief overview of this thesis.

Theoretical Background (chapter 2) This chapter explains the core concepts used for the chosen implementation.

Approach and Methods (chapter 3) This section covers the system architecture and implementation as well as details of how the system

was parameterized and how results were measured and validated.

Results and Discussion (chapter 4) This part shows and discusses the achieved results. Special attention is given to section 4.1.5 where the best parameters to achieve a low DER are evaluated.

Conclusion (chapter 5) In this section, the significance of the results is critically questioned. A reflection on what was achieved compared to the initial objective will also be presented. Additionally, possible improvements to the system and ideas will be proposed.

Chapter 2

Theoretical Background

This chapter provides the main theoretical background and context for SR and its evaluation, as well as the algorithms used in the thesis.

2.1 Speaker Identification

This section introduces the main techniques of the SI system used in this thesis.

2.1.1 Log-Mel Spectrogram

The Mel-scale was first introduced in the 1930s. The goal was to define a unit of pitch, such that distances that sounded alike to the listener were measured as equal [16]. A Mel spectrogram is the visual representation of sound on which the Mel-scale was applied. If the amplitude is additionally logarithmized, the spectrogram is referred to as Log-Mel spectrogram. The logarithmic compression of the Mel spectrograms has been established in neural network audio classifiers and are according to Kinnunen and Li [17] the preferred visual representation of audio.

2.1.2 Convolutional Neural Network

CNNs belong to the category of deep learning and are often used in the field of computer vision. A CNN can take an image as input and reduces it to a form which is simpler to process while capturing relevant features. The network is then able to differentiate one image from another [18]. Image classification is often based on CNNs for several reasons. On one hand, images can be entered directly into the system as input and therefore handcrafted feature extraction is no longer necessary. On the other hand it is, on the basis of spatial size reduction, scalable to a large amount of data [18].

2.1.3 Visual Geometry Group Model

VGG is one of various CNN architectures and one of the most used image recognition architectures [19]. In 2014, Simonyan and Zisserman demonstrated in [8] that the depth of a CNN is crucial for the model's accuracy. That's the reason why VGG pays particular attention to this aspect. Additionally it was shown that the combination of an increasing depth to 16-19 weight layers and the usage of very small (3x3) convolution filters improves the performance of a CNN remarkably [8].

2.1.4 VGGish

TensorFlow, an open source platform for end-to-end machine learning, provides an implementation of VGG, which was trained on the initial AudioSet [20], a large YouTube dataset. This model was introduced as VGGish [21].

AudioSet is a human labeled dataset for audio events released in 2017

by Google [20]. The aim of this large scale dataset is to close the research loop between image and audio [20]. The corpus comprises labeled YouTube segments of 10 seconds which belong to one or more class labels. The dataset has the following characteristics [20]:

- The categories have been chosen so that they describe audio from the real-world.
- The categories are named to indicate instantly to the listener what the sound is when it is heard.
- The categories are structured in a hierarchical way, so that, for example, the category ”dog sounds” includes sounds of ”growl” or ”howl” and ”dog sounds” belongs to ”domestic animals”, which in turn is a subcategory of ”animal sounds”.
- The audio should be assignable to a category without any context details or visual support.

VGGish can be used with different strategies:

- **As a feature extractor** With this approach the model can extract features from the input audio, which can be fed into a classification model afterwards. It takes over the task of the feature extractor and can be used for transfer learning as described in section 1.1.
- **As a classifier** VGGish can also directly be used for classification by adding additional layers on top of the provided model [21].

The architecture of VGGish is based on a configuration described in [8], which comprises eleven weight layers. However, TensorFlow made some adjustments in their reference implementation:

Input Image
conv2D-64
max pooling
conv2D-128
max pooling
conv2D-256 conv2D-256
max pooling
conv2D-512 conv2D-512
max pooling
global average pooling

Table 2.1: VGGish architecture implemented by TensorFlow

- The model uses an input size of 96 x 64 for log mel spectrogram audio inputs [21].
- Only four groups of convolution / maxpool layers have been used. For this purpose the last group of convolutional and max-pool layers has been dropped.

Reducing the architecture to four layers, outputs the structure as shown in table 2.1 in which all activation functions are rectified linear units (ReLU).

2.1.5 Classification

Classification is a subcategory of supervised learning. Given some input variables x, classification is the process of approaching a function f which maps the input to discrete output variables y. These are also called labels, categories or classes [22]. In the following the classifiers that are used in this thesis are briefly described:

- Support vector machines (SVM) are a set of supervised learning methods, which can be used for classification. For binary and

multi-class classification on a dataset, Support Vector Classification (SVC) and Linear SVC are possible characteristics of SVM. The latter characteristic contains the term linear because of its linear kernel [23].

- kNN is an acronym and stands for k-Nearest Neighbor. When an input *x* is received, it analyses the closest *k* instances (nearest neighbors) and takes the most common class, according to these nearest neighbors, as prediction [22].

Cross-validation (CV) can be applied to all of the classifiers mentioned above. It is used that the risk of overfitting, a modeling error when a function is too close to the given training data points, can be reduced [22]. Using the basic approach, the so called k-fold CV, the training data is partitioned into *k* smaller sets. The model is then trained on *k-1* of the folds as training data and the remaining part of the data is used as test set [24].

2.2 Evaluation of the System

After training the model, different evaluations can be conducted. Two evaluation possibilities are introduced in this section.

2.2.1 Accuracy

One metric for evaluating classification models is accuracy. It gives an indication about the quality of the model. Accuracy is calculated as follows [25]:

$$accuracy = \frac{number\ of\ correct\ predictions}{total\ number\ of\ predictions}$$

2.2.2 Diarization Error Rate

Another measurement possibility and the de facto standard metric for the quality of a SD system is the DER.

$$DER = \frac{false\ alarm + missed\ detection + confusion}{total\ speech}$$

The numerator is the sum of false alarm, missed detection and confusion. It represents the input signal that is incorrectly labeled.

- **False alarm** is the length of non-speech that was wrongly classified as speech.
- **Missed detection** is in fact the length of speech incorrectly classified as non-speech.
- **Confusion** represents the length of speech that was assigned to the wrong speaker [26]. This error emerges for example through a speaker change which is not detected [27].

Chapter 3

Approach and Methods

This thesis covers a self-hostable system, implemented in Python, which is described in section 1.3.1. Other approaches, such as using an established speech service, have been evaluated. For reasons described in section 1.3.1 these services were discarded from further evaluation. The implemented solution is solely based on open source libraries which greatly reduces costs of running the system.

The core system consists of three main parts. Figure 3.1 shows these parts, as well as the inner modules. Each of these parts and its modules will be explained in depth in the following sections.

3.1 Data Preprocessing

The first part of the application is responsible for preprocessing the audio file which is given as input. After this step, the data can then be submitted to the speaker recognition part.

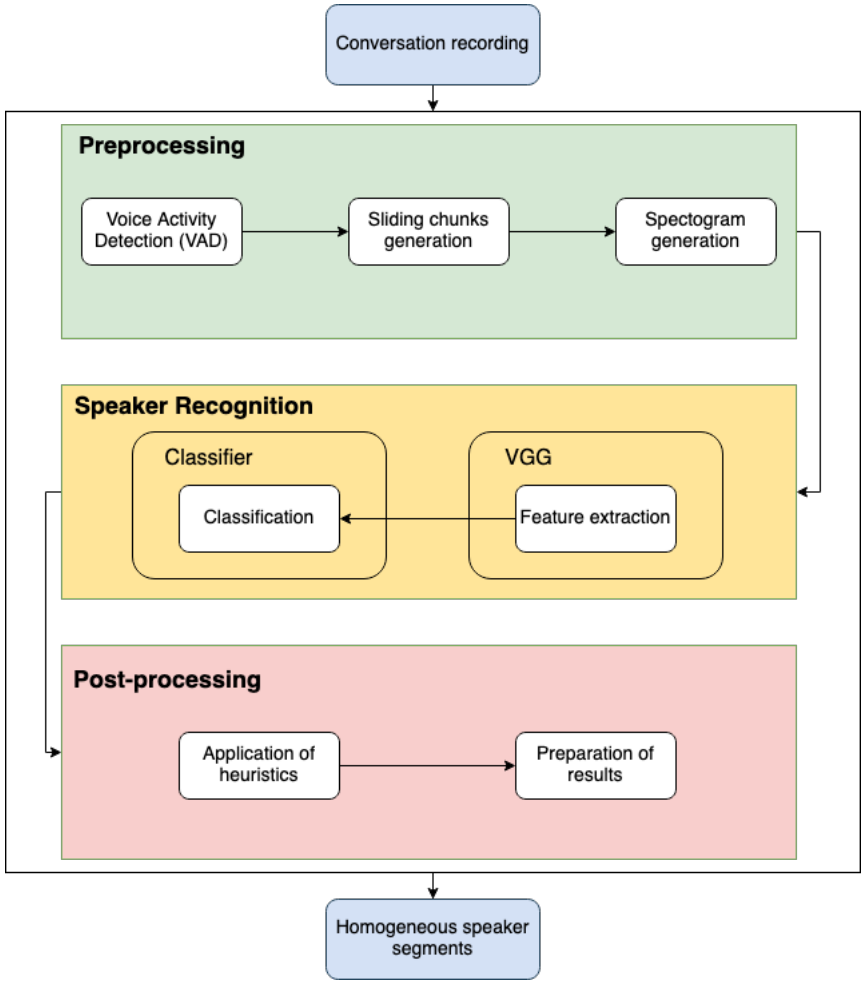

Figure 3.1: System architecture and data flow overview

3.1.1 Voice Activity Detection

Voice activity detection (VAD) is used in a wide range of different speech technologies [28]. Since silence is not assignable to a speaker, it must be removed from the conversation. This module analyses the audio file and discards parts which do not include speech.

The application built for this thesis relies on Google's VAD implementation which was developped for the Web Real-Time Communication (WebRTC) [29], an open source application programming interface (API) for browsers.

Chunk No.	Text
1	Grüss Gott Frau Müller
2	Frau Müller ich
3	ich komm vorbei
4	komm vorbei wegen
5	bei wegen unserer
6	unserer Geschä
7	Geschäftsrei
8	ftsreise

Table 3.1: An example of the division of an initial sentence into chunks

3.1.2 Sliding Chunks Generation

A sliding and overlapping window algorithm, as shown in figure 3.2, was implemented to generate equally long audio files from the output of the VAD. The choice to overlap the chunks is based on the fact that the advantages of big and small chunks can be combined. Bigger audio fragments contain more speech which is beneficial to the developed system. Smaller chunks on the other hand reduce the risk of more than one speaker in a segment.

Since there is no gold standard on how long these chunks and the overlap between them should be, this was parameterized. Further details on chosen parameters can be found in section 3.4.1.

Table 3.1 illustrates how a sentence of 3375 ms with the content ”Grüss Gott Frau Müller, ich komm vorbei wegen unserer Geschäftsreise” is divided with a chunk size of 750 ms. It is apparent that such a short chunk size can already incorporate several words and that the grammatical correctness is not crucial.

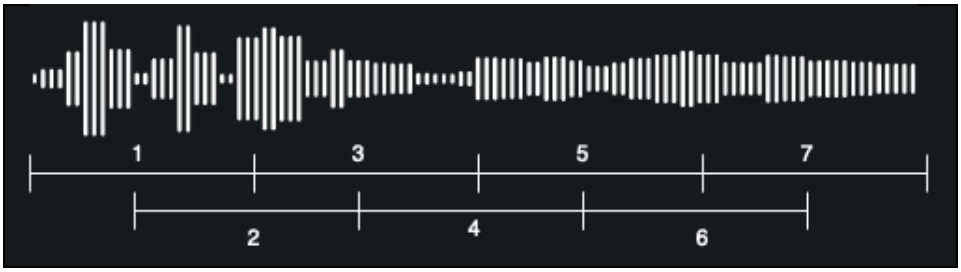

Figure 3.2: Visualization of the sliding window algorithm used to create audio chunks

3.1.3 Spectrogram Generation

Since VGG is commonly used for image classification (as described in section 1.1), the system needs to visualize the audio files in order to feed them into the neural network. TensorFlow's reference implementation for VGGish uses Log-Mel frequency spectrograms [21] which is explained in depth in section 2.1.1. Figure 3.3 shows what the generated Mel spectrogram and Log-Mel spectrogram of four 750 ms audio chunks seperated by 250 ms silence look like. Clearly visible is the applied sliding window algorithm, as the second half of a preceding chunk is the same as the first half of the next one.

3.2 Speaker Recognition

The second block is intended for the SR part. An indirect approach, as described in section 1.1, has been chosen for the implementation.

3.2.1 Feature Extraction

VGGish, as described in section 2.1.4, is used for feature extraction. Through this process, the prepared spectrograms are converted to 128 dimensional embeddings which can then be fed into the next module.

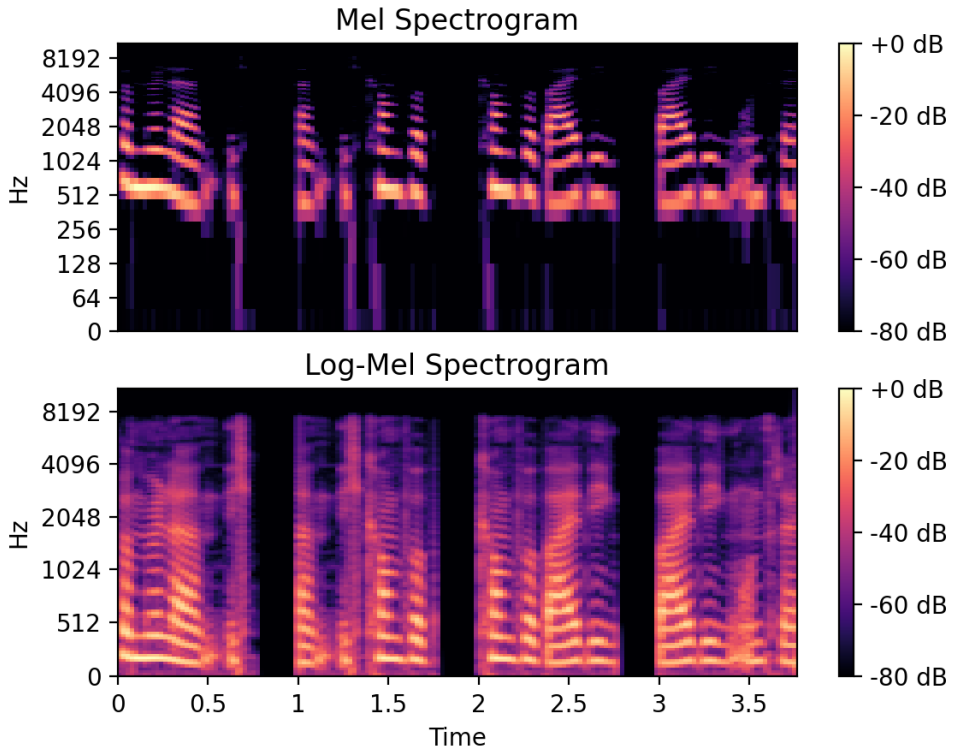

Figure 3.3: Four 750 ms long audio chunks visualized as Mel spectrogram and Log-Mel spectrogram

3.2.2 Classification

The classification maps the features extracted by the VGGish to the speaker. This results in the confidence of each speaker for the given audio segment. Since there are different types of classifiers (section 3.4.1), the best one according to section 4.1.1 was chosen.

3.3 Post-processing

The SR block provided the confidences for each speaker in each audio chunk. This part allows the modules to concatenate the speech fragments as precisely as possible.

3.3.1 Apply Heuristics

This module incorporates self-developed heuristics which define on how the segments should be concatenated back to a whole audio file. Two heuristics were implemented in this thesis and are used together:

1. Given three audio chunks ($chunk_{x-1}$, $chunk_x$ and $chunk_{x+1}$):
   If $chunk_{x-1}$ and $chunk_{x+1}$ are assigned the same speaker but $chunk_x$ is assigned a different one, then assign $chunk_x$ to the same speaker as $chunk_{x-1}$.

2. Given x audio chunks ($chunk_i$, $chunk_{i+1}$, .. , $chunk_{i+x}$) of length c which are consecutive and assigned to the same speaker but surrounded by chunks assigned to other speakers. Further there is a factor a (described in section 3.4.1) and a function *len*, which calculates the length of a chunk in milliseconds.
   If
   $$len(chunk_i + .. + chunk_{i+x}) <= c * a$$
   then assign the chunks ($chunk_i$, $chunk_{i+1}$, .. , $chunk_{i+x}$) to the speaker of $chunk_{i-1}$.

3.3.2 Prepare Result

This module prepares the final result. In order to generate the ultimate result, the chunks were merged to homogeneous speaker segments by reverting the sliding window algorithm. Since each chunk contains half of the next one, the system needs to append the first half of each segment to the summary of the preceding ones. Additionally, the non-speech parts, which were removed by the VAD in section 3.1.1, must be reapplied. The outputs of this module are con-

Parameter	Values
Training data duration in ms	$\{ x + 250 \mid 250 \leq x \leq 2000 \}$
Chunk length in ms	$\{ x + 5000 \mid 5000 \leq x \leq 60000 \}$
Classifier	{ kNN, SVC (linear&RBF kernel) }
Skip factor *a*	{ 1.2, 1.8 }

Table 3.2: Parameters applied to the system during the experimental phase

sistent segments with the corresponding speakers and timestamps of the original audio file.

3.4 Experiments

This section describes the used corpus as well as the parameters that have been applied in order to build the best possible SD system with respect to the sub-objectives described in section 1.3.1.

3.4.1 Parameters

As described in sections 1.2 and 1.3.1, the goal of the thesis is to implement a SI system with a DER as low as possible by only having few utterances given by each speaker. But how short can these segments be without worsening the DER? And what is generally the perfect balance between the length of the utterances and the DER? To answer these questions, the system was trained on the parameters as shown in table 3.2 to provide the best possible setting.

- **Training data duration:** This parameter can take values from 0.25 to two seconds, with an increase step of 250 milliseconds. This parameter defines the total length of the training time per speaker.

- **Chunk length:** This parameter can take values from five seconds to one minute, with an increase step of five seconds. This parameter defines the length of a chunk that is used for training purposes.
- **Classifier:** According to section 2.1.5, two different classifiers are tested. The SVC classifier is further evaluated with diverse parameters, such as a linear or radial basis (RBF) kernel. The best performing constellation is in addition tuned with CV. The used regularization parameter *C* [30] can take the following values: 1, 10, 100, 1000. The regularization parameters are determined anew for each conversation.
- **Skip factor a:** To get the lowest possible DER, heuristics are applied before the chunks are merged back into one audio file. This factor determines how long the smallest segment may be. The factor can represent the values 1.2 and 1.8 according to table 3.2.

Based on the training time and the length of the chunks, the amount of training data can be calculated. The training time is therefore taken as dividend and the chunk length is chosen as divisor. Applying the division algorithm, the quotient yields to the number of training samples which are then given as input into the proposed system of this thesis.

3.4.2 Training and Testing

Speech Corpus

Testing is crucial to measure the performance of the system. To asses the developed software, a part of Verbmobil II [31], a speech corpus

Metric	Value
No. of used conversations	68
Avg. conversation duration	194.6 s
Female conversations	49 (72.0%)
Male conversations	3 (4.4%)
Mixed conversations	16 (23.6%)
Discarded conversations	12

Table 3.3: Key metrics of the used Verbmobil II speech corpus

developed by the University of Munich, was used. Table 3.3 describes key metrics of this dataset. All used conversations which are less than two minutes are discarded.

Different recording settings can be assigned to each conversation:

- Speakers are on a mobile or analog phone.
- Speakers are in the same room.
- Speakers are in a closed, separate room. Recording is then done either by a headset, a neckband microphone or a clip microphone.

Phone recordings are subject to more white noise and have a sample rate of 8 kHz, while the other recording settings have a 16 kHz sample rate.

All used conversations in Verbmobil II are limited to two persons without overlapping speech. Each speaker fragment is an own audio file, thus it is possible to get the exact timestamps of all speaker changes.

Training Data

To create training data, a defined number of chunks of length c (as described in section 3.4.1) from each speaker is taken randomly from

a conversation. This data is used for training the classifier.

Testing Data

The test dataset for the use case in this thesis is different when calculating the accuracy and the DER.

- **Accuracy** To calculate the accuracy, the remaining chunks after extracting the training set is taken as testing data.
- **DER** For the DER the whole conversation, including the training data, is taken as test set. The reason for this is that the verified voice samples of the speakers in our use case are taken from the same conversation which needs to be transcribed.

Chapter 4

Results and Discussion

In this chapter different experiments are described, evaluated and discussed on the corpus described in section 3.4.2. For comparison purposes LibriSpeech was used in the last experiment.

4.1 Choosing Appropriate Parameters

This section explains the most suitable choice of parameters.

4.1.1 Choosing an Appropriate Classifier

As a first step, a fitting classifier is chosen from the various classifiers described in section 3.4.1. All classifiers have been applied with the default parameters [32].

Figure 4.1 shows the predefined classifier's performance according to different training data durations. The performance metrics are the average DER and accuracy (sections 2.2.2 and 2.2.1) per training duration across all conversations, skip factors and chunk sizes described in section 3.4.1.

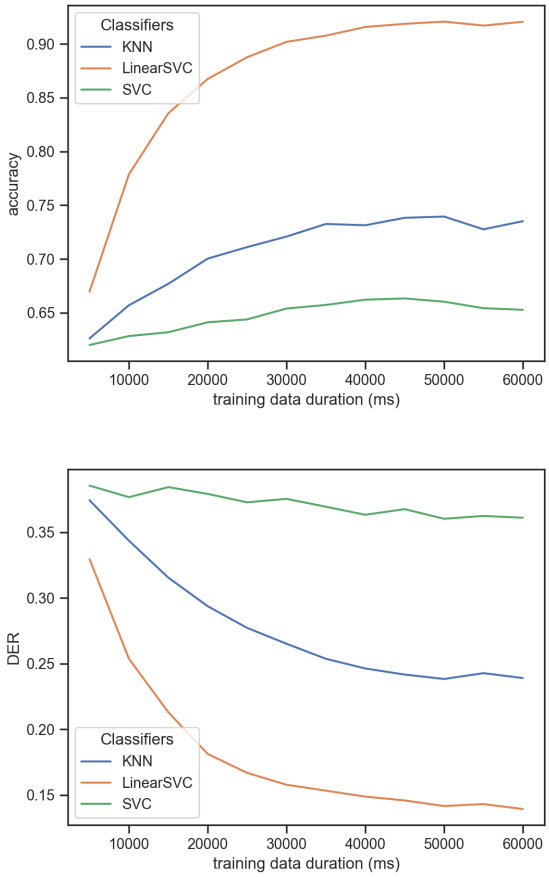

Figure 4.1: Performance of various classifiers on a range of different training data durations measured by accuracy and DER

As is clearly visible in figure 4.1, SVC with a linear kernel (referred to as linearSVC) performs better from the very beginning than the kNN or SVC with a RBF (referred to as SVC). With a training data duration of only 30 seconds, linearSVC achieves an accuracy of over 90%. For the same length of training data duration, kNN only achieves an accuracy of approximately 72% and SVC even below 63%.

Looking at the classifier's DER, the same trend emerges. LinearSVC achieves an average DER of less than 20% at 20 seconds of training data. With the same training data duration, kNN achieves a DER of about 30% and SVC is even worse at 38%.

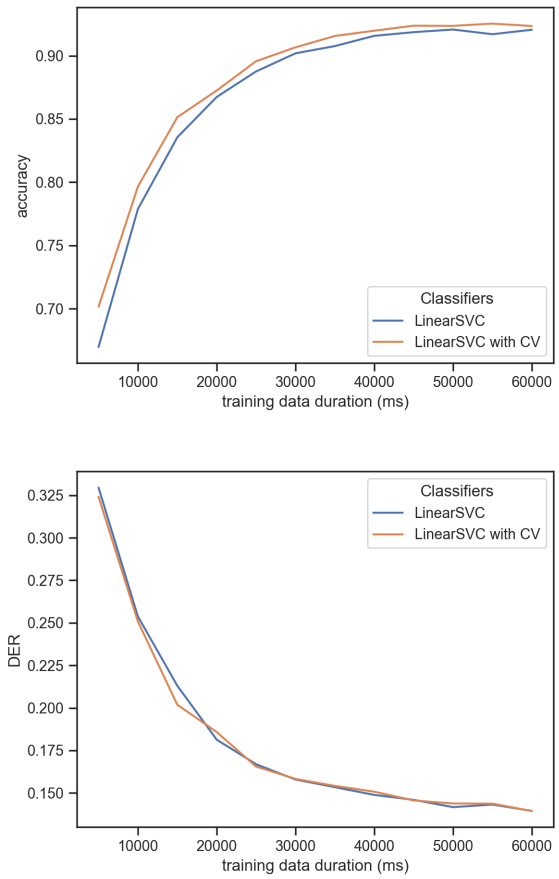

Figure 4.2: Performance of linearSVC with and without cross-validation on a range of different training data lengths measured by accuracy and DER

According to figure 4.1 it is assumed that linearSVC is best suited for the setting.

With the help of CV, overfitting can be prevented. This could lead to a performance increase of the classifier. CV was therefore applied on linearSVC with the parameters described in section 3.4.1. The comparison of linearSVC and cross-validated linearSVC was again carried out on the basis of accuracy and DER. The results achieved are shown in figure 4.2.

Comparing linearSVC and linearSVC with CV, the discrepancy be-

tween them is no longer that striking. LinearSVC with CV always performs better than without, if accuracy is taken as the measure. Comparing the DER, no clear winner can be determined.

Since the two classifiers have similar performances and the one with CV has achieved better accuracy, the following experiments are performed with cross-validated linearSVC.

4.1.2 Choosing an Appropriate Training Data Duration

Given the selected classifier, a suitable training data duration can be selected next. As visible in figure 4.2, there is a kink in the graph at 25 seconds. From that moment, DER decreases only minimally in the range of around 2%. For this reason, the training data duration of 25 seconds is assumed to be the most suitable.

4.1.3 Choosing an Appropriate Chunk Size

After the classifier and the training data duration were fixed, the right choice of chunk length is being evaluated. A scoring algorithm as follows was chosen:

For each conversation

1. Sort the chunk sizes ascending by their achieved DER
2. Score the chunk size with the lowest DER three points, the chunk size with the second lowest DER two points and the chunk size with the third lowest DER one point. All other chunk sizes score zero points.

Figure 4.3 shows the points that each chunk size has achieved. The 750 milliseconds chunk size was the best, with 95 points, followed by 1000 milliseconds with significantly fewer points.

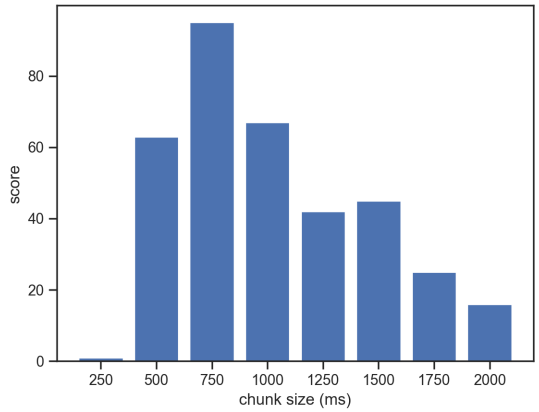

Figure 4.3: Evaluation of the selected chunk sizes based on their achieved scores

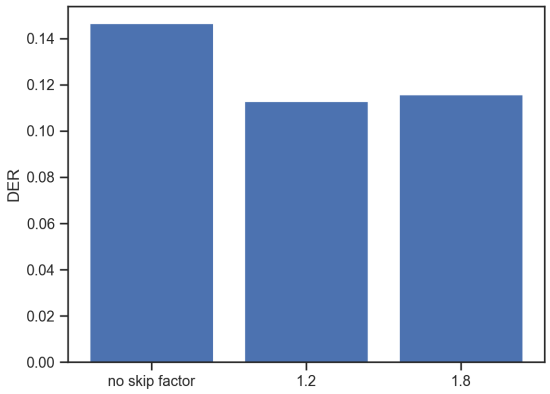

Figure 4.4: Evaluation of the skip factors based on the DER

The strong differentiation of the scores shows that the appropriate choice of chunk size is very important for the performance of the system.

4.1.4 Choosing an Appropriate Skip Factor

The last parameter to be evaluated is the skip factor *a* (section 3.4.1) which determines the smallest segment.

Figure 4.4 shows the result of the comparison between the three fac-

tors which were chosen to merge the chunks:

- No skip factor applied for merging the chunks
- A skip factor of 1.2 (section 3.4.1)
- A skip factor of 1.8 (section 3.4.1)

As shown in figure 4.4, there is not much difference between the three skip factors. In particular the skip factor of 1.2 and 1.8 differ in a DER of less than one percent. It is therefore assumed that the skip factor has no great influence on the DER.

4.1.5 Summary of the Findings on the Parameters

According to the above sub-experiments, it can be concluded that the correct choices of a classifier and the length of the chunks are key criteria for success, because they significantly influence the DER. In addition to these two parameters it is also important to have a certain amount of training time available.

In contrast to these three important factors, the skip factor does not play a major role. It is debatable whether further skip factors will lead to a significant improvement.

For the further experiments the parameters were set as follows:

- **Classifier:** Cross-validated SVC with linear kernel
- **Training data duration:** 25000 milliseconds
- **Chunk size:** 750 milliseconds
- **Skip factor:** 1.2

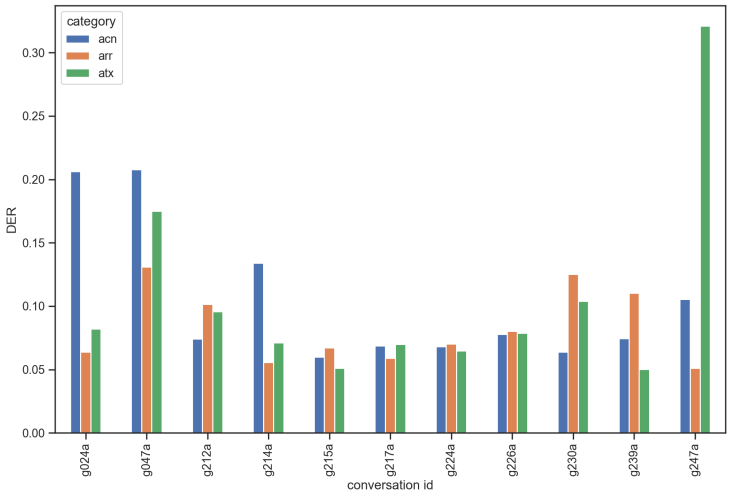

Figure 4.5: Evaluation of the different conversation settings based on the DER

4.2 Evaluation on Different Settings

As described in section 3.4.2, there are different settings for a conversation. Given the parameters chosen in section 4.1.5, it is analysed, how the DER vary by settings.

Figure 4.5 shows the conversations from the Verbmobil II corpus, which were recorded in 3 different settings. The settings can be described as follows:

- **acn:** the setting acn is an acronym and stands for the scenario *main* (**a**), the technical definition of the recording *close* (**c**) and the detailed description *neckband microphone* (**n**). The audio recordings in this setting have a sample rate of 16 kHz.
- **arr:** the setting arr is an acronym and stands for the scenario *main* (**a**), the technical definition of the recording *room* (**r**) and the detailed description *room* (**r**). The audio recordings in this setting have a sample rate of 16 kHz.

ID	Dialogue	Setting	Gender	Conversation Length	DER
1	g222a	atx	f, m	310580 ms	2.1%
2	g222a	acn	f, m	290820 ms	2.7%
3	g236a	acn	f, f	120880 ms	3.7%
4	g239a	atx	f, f	148020 ms	5.0%
5	g215a	atx	f, m	314580 ms	5.1%

Table 4.1: Details from the five best settings of the Verbmobil II corpus according to the DER

- **atx:** the setting atx is an acronym and stands for the scenario *main* (**a**) and the technical definition of the recording *telephone* (**t**). The **x** stands either for *mobile* or *analog phone* and is a detailed description of the recording. The audio recordings in this setting have a sample rate of 8 kHz.

As can be seen in figure 4.5, there is no setting that consistently shows the lowest DER. Accordingly, there is no direct correlation between the sample rate and the DER, although a higher sample rate of an audio recording leads to a greater audio resolution. Therefore, it can be deduced that the system proposed in this thesis can also handle different settings.

4.3 Evaluation of the Outliers

The aim of this experiment is to evaluate the dialogues that scored particularly well and particularly badly in relation to the DER. It is the goal to analyse the outliers and to find connections between them.

4.3.1 Evaluation of the Best Performers

By analyzing the data shown in table 4.1, no conclusions can be drawn from the details. However, it is interesting that the speakers in conver-

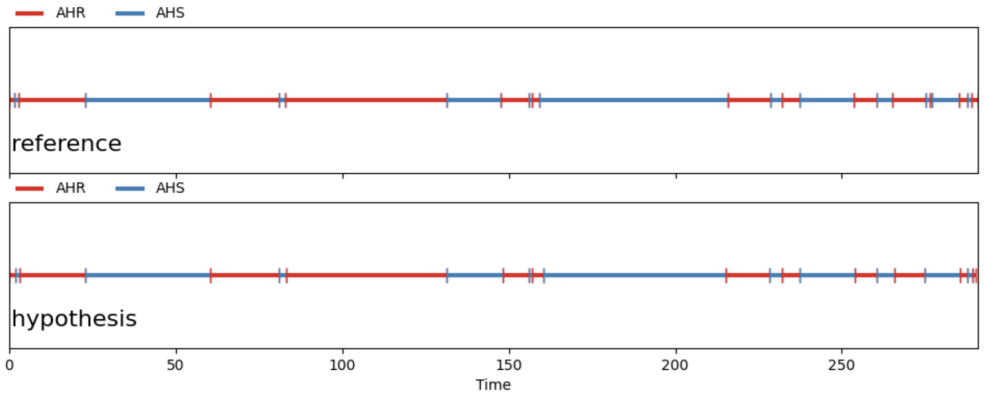

Figure 4.6: Comparison between the calculated and reference speaker changes of an audio recording. The figure shows homogeneous speaker segments based on the best performing conversation of the used corpus.

sation g222a (ID 1 & 2) and the speakers in conversation g236a (ID 3) respectively g239a (ID 4) are the same. It seems that there are speaker pairs which are especially favorable for the setting. It is also apparent that no male pair has made it into the top five.

Figure 4.6 visualizes the best conversation (ID 1) evaluated in table 4.1 by comparing the hypothesis calculated by the system to the reference speaker changes. The audio stream is divided into homogeneous segments of the different speakers whereas each speaker is assigned a specific colour. It can be clearly seen that reference and hypothesis speaker changes are almost identical, Which is an indication that the system performs well for the conversation.

4.3.2 Evaluation of the Worst Performers

If one looks at the five worst performers in table 4.2 in relation to DER, it is striking that in all five settings the genders of both speakers are the same. The combination of two women is particularly difficult for the system to distinguish.

ID	Dialogue	Setting	Gender	Conversation Length	DER
1	g525a	ach	f, f	195600 ms	25.6%
2	g539a	ach	f, f	156380 ms	26.1%
3	g538a	ach	f, f	214360 ms	26.9%
4	g247a	atx	m, m	188460 ms	32.1%
5	g522a	ach	f, f	317380 ms	35.7%

Table 4.2: Details from the five worst settings of the Verbmobil II corpus according to the DER

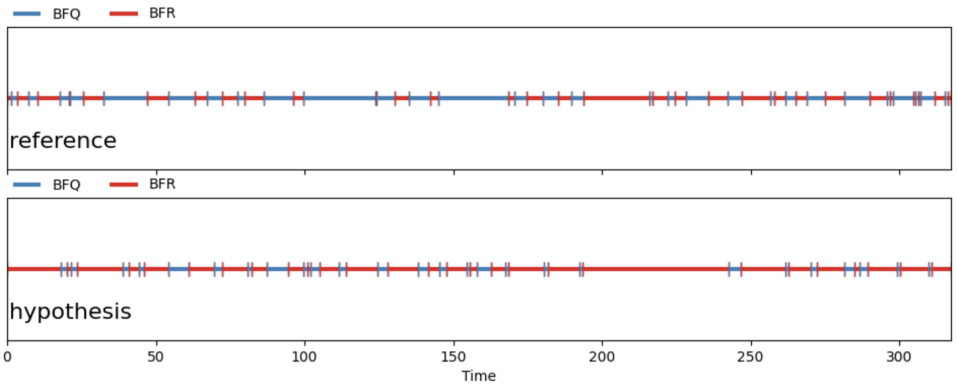

Figure 4.7: Comparison between the calculated and reference speaker changes of an audio recording. The figure shows homogeneous speaker segments based on the worst performing conversation of the used corpus.

Figure 4.7 visualizes the worst conversation (ID 5) evaluated in table 4.2 by comparing the hypothesis calculated by the system to the reference speaker changes. If the speaker changes are compared, it is noticeable that only a few segments match. This is an indication that the system is not performing well for the setting.

A manual qualitative analysis has shown that it can be difficult even for a human to distinguish these speakers.

4.4 Evaluation on a Different Corpus

To be able to compare to other state-of-the-art systems, conversations generated from the LibriSpeech corpus [33] were used. To represent a real-world scenario, different conversations were assessed and the diarization was executed with the parameters discussed in section 4.1.5. The conversations were generated by randomly concatenating audio recordings from different speakers. The speakers were chosen sequentially, which means that the speakers were the same for all conversations and just one new person was added to increase the total number per experiment.

Table 4.3 shows the results with various numbers of speakers as well as different conversation durations.

By evaluating the DER compared to the number of speakers, it can be seen that raising the number of conversation participants also slightly raises the DER. Surprisingly, the DER does not augment dramatically as the number of speakers rises, although increasing the number of speakers makes a SD setting a lot more complex.

In most cases incrementing the conversation duration also leads to a higher DER. But again, no significant increase is apparent.

Based on the findings just explained it can be assumed that a further proliferation of speakers and conversation duration will not unexpectedly increase the DER by a multiple.

ID	No. of Speakers	Conversation Length	DER	Accuracy
1	2 (1*x* f, 1*x* m)	8:36 min	1.7%	95.7%
2	2 (1*x* f, 1*x* m)	17:12 min	3.1%	95.5%
3	3 (1*x* f, 2*x* m)	08:54 min	4.8%	92.6%
4	3 (1*x* f, 2*x* m)	17:42 min	2.9%	96.9%
5	4 (2*x* f, 2*x* m)	09:00 min	2.9%	96.9%
6	4 (2*x* f, 2*x* m)	17:54 min	3.8%	94.2%
7	5 (3*x* f, 2*x* m)	09:06 min	4.8%	91.8%
8	5 (3*x* f, 2*x* m)	18:00 min	4.9%	90.8%

Table 4.3: Evaluated results on conversations generated from LibriSpeech corpus

Chapter 5

Conclusion

This thesis presented a system to perform SD with a short amount of voice recording for each speaker as training data. The conversations are cut into small chunks of 0.25 - two seconds. The latter is fed into VGGish, a VGG based model from Keras, trained solely on audio files, to perform feature extraction. These features are then predicted by a variety of classifiers.

Evaluating the best performing parameters, cross-validated SVC with linear kernel paired with a training data duration of 25 seconds and a chunk size of 0.75 seconds has proven the most successful. When merging the chunks together, a skip factor (section 4.4) of 1.2 have shown best results.

The system proposed has an average accuracy of 89% and a DER of 11.3% over all results evaluated on the Verbmobil II corpus (section 3.4.2, appendix 7.3).

5.1 Comparison with Initial Objective

A system for SD based on SI with a-priori speaker samples was built in this thesis according to the main objective, described in section 1.2. This system can be hosted on any infrastructure, which was, according to section 1.3.1, a sub-objective of this thesis.

By evaluating more than 58'000 parameter combinations (all combinations can be found in appendix 7.2), the most suitable parameters for reducing the DER (section 1.3.2) have been identified.

Due to the lack of time it was not possible to implement the voice database, described in section 1.4.

5.2 Significance of this Thesis

The results shown in this thesis have proven that SD through identification is a promising concept.

However there are some limitations which should be kept in mind:

- The parameters were evaluated on a limited corpus.
- A conversation in the corpus is on average three minutes long. Having a training data duration of 25 seconds restricts the expressiveness of the DER, because the training data can take up to one-seventh of the original file. This does not reflect a real interview or conversation.
- Each conversation of this dataset consists of only two persons.
- There is no overlapping speech.

Part of these limitations have been justified by evaluating the system on a second corpus described in section 4.4.

5.3 Questions Arising

During the implementation phase of the system, a few questions were raised. To create a production-ready system, this questions should be analyzed in-depth.

5.3.1 Voice Activity Detection

The VAD, which is responsible to cut out non-speech segment, could be too aggressive. This may lead to the deletion of sections which incorporate speech. However, the calculation of DER in this thesis does not take this error into account.

To reduce the risk of having this issue, the detected non-speech segments should be re-applied to the final result.

5.4 Comparison to Other Systems

In section 4.4, a brief, qualitative evaluation of the system was made on the LibriSpeech Corpus, which resulted in an average accuracy of 94.3% on a system of two to five speakers. Compared to Vélez, Rascon and Fuentes-Pineda [14], who also used VGG for SI and LibriSpeech Corpus for the evaluation, the system of this thesis performed 3% worse. It can be assumed that with the implementation of the suggestions for improvement, which are listed in section 5.5, an increase of accuracy becomes noticeable. Furthermore, only a small amount of comparative data was taken from the LibriSpeech corpus during the evaluation. For an accurate comparison, a quantitative evaluation should be carried out.

The average DER on the Verbmobil II corpus is 11.3% and 3.6% on the

LibriSpeech corpus. Zhang et al. reached in the paper published in 2019 [15] a DER of 7.6%. Even though the difference between the DER achieved with the Verbmobil II corpus and the DER of Zhang et al. is not striking, it is important to note that Zhang et al. [15] has made its evaluation on the NIST SRE 2000 CALLHOME corpus [34], which includes telephone conversations with overlapping speech. The comparison should therefore be treated with caution. Instead, the system from this thesis should be evaluated on the same corpus in order to be able to make a valid comparison.

5.5 Directions for Future Research

The scope of this thesis was to build a SD system with a low DER. The following suggestions explain how the performance of the proposed system can be improved.

- **Train own model** A custom trained model may lead to better performance, since it is trained exactly for the use case. This includes the adjustment of the training data to fit exactly the parameters defined in section 3.4.1.
- **Different training and test data lengths** In the evaluation of the parameters the chunk size of the training data was always chosen to be exactly the same length as the testing data. However, it cannot be assumed that this leads to better results. The effect of varying chunk size and how to optimize should therefore be investigated.
- **Fully connected VGGish** Instead of using a classifier for transfer learning, additional layers can be added to the pre-trained VGGish model to perform the direct method mentioned in sec-

tion 1.1. This could have an impact on the suggested parameters.

- **Fine tuning of the applied heuristics** Since only a few heuristics are implemented (section 3.3.1) and evaluated, finding more appropriate heuristics may have a huge impact on the DER and the overall performance. This is considered as a key element to the system.
- **Replace VAD** To solve the risk discussed in section 5.3.1 the VAD could be fully removed. As an alternative, a silent speaker which is trained on a variety of background noises could be introduced.
- **Adding principal component analysis** As a postprocessing step of the feature extraction, a principal component analysis (PCA) could be applied. This would lead to an increasing interpretability of the data while minimizing the information loss.

Chapter 6

Registers

6.1 Bibliography

[1] S. E. Tranter and D. A. Reynolds, “An overview of automatic speaker diarization systems”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, pp. 1557–1565, 2006.

[2] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-End Factor Analysis for Speaker Verification”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[3] G. Sell and D. Garcia-Romero, “Speaker diarization with plda i-vector scoring and unsupervised calibration”, *2014 IEEE Spoken Language Technology Workshop (SLT)*, pp. 413–417, 2014.

[4] F. Richardson, D. Reynolds, and N. Dehak, “Deep Neural Network Approaches to Speaker and Language Recognition”, *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1671–1675, 2015.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”, *NIPS*, pp. 1106–1114, 2012.

[6] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision", *arXiv preprint arXiv:1512.03385*, 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Rethinking the inception architecture for computer vision", *arXiv preprint arXiv:1512.03385*, 2015.

[8] K. Simonyan and A. Zissermann, "Very deep convolutional networks for large-scale image recognition", *arXiv:1409.1556*, pp. 131–135, 2014.

[9] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database", *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[10] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, "CNN architectures for large-scale audio classification", *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[11] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books", *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210, 2015.

[12] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: a large-scale speaker identification dataset", 2017. arXiv: 1706.08612.

[13] H. Eghbal-Zadeh, B. Lehner, M. Dorfer, and G. Widmer, "A hybrid approach with multi-channel i-vectors and convolutional neural networks for acoustic scene classification", *2017 25th Eu-*

ropean Signal Processing Conference (EUSIPCO), pp. 2749–2753, 2017.

[14] I. Velez, C. Rascon, and G. Fuentes-Pineda, "One-Shot Speaker Identification for a Service Robot using a CNN-based Generic Verifier", 2018. arXiv: 1809.04115.

[15] A. Zhang, Q. Wang, Z. Zhu, J. Paisley, and C. Wang, "Fully Supervised Speaker Diarization", p. 4, 2018. arXiv: 1810.04719.

[16] L. Roberts. Understanding the Mel Spectrogram, [Online]. Available: https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53. (accessed: 26.05.2020).

[17] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors", *Speech communication*, vol. 52, no. 1, pp. 12–40, 2010.

[18] S. Saha. A Comprehensive Guide to Convolutional Neural Networks, [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. (accessed: 26.05.2020).

[19] J. Wei. VGG Neural Networks: The Next Step After AlexNet, [Online]. Available: https://towardsdatascience.com/vgg-neural-networks-the-next-step-after-alexnet-3f91fa9ffe2c. (accessed: 26.05.2020).

[20] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio Set: An ontology and human-labeled dataset for audio events", *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 776–780, 2017.

[21] The TensorFlow Authors. VGGish, [Online]. Available: https://github.com/tensorflow/models/tree/master/research/audioset/vggish. (accessed: 26.05.2020).

[22] S. Asiri. Machine Learning Classifiers, [Online]. Available: https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623. (accessed: 27.05.2020).

[23] Scikit-learn developers. Support Vector Machines, [Online]. Available: https://scikit-learn.org/stable/modules/svm.html. (accessed: 27.05.2020).

[24] Scikit-learn developers. Cross-validation: evaluating estimator performance, [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html. (accessed: 27.05.2020).

[25] A. Mishra. Classification Accuracy, [Online]. Available: https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234. (accessed: 27.05.2020).

[26] CNRS. Reference, evaluation metrics, [Online]. Available: https://pyannote.github.io/pyannote-metrics/reference.html. (accessed: 27.05.2020).

[27] A. Kumar and A. Kumar. Unsupervised Speaker Diarization, [Online]. Available: https://pdfs.semanticscholar.org/2923/d954545ab410a0a7569248a471ca109a2002.pdf. (accessed: 28.05.2020).

[28] T. Drugman, Y. Stylianou, Y. Kida, and M. Akamine, "Voice Activity Detection: Merging Source and Filter-based Information", *IEEE Signal Processing Letters*, vol. 23, no. 2, pp. 252–256, 2016.

[29] The WebRTC project authors. VAD, [Online]. Available: https://webrtc.googlesource.com/src/+/refs/heads/master/modules/audio_processing/vad/. (accessed: 25.05.2020).

[30] Scikit-learn developers. SVC, [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html. (accessed: 07.06.2020).

[31] University of Munich, Institut of Phonetics. VERBMOBIL II Dialog Database (BAS Edition), [Online]. Available: https://www.phonetik.uni-muenchen.de/Bas/BasVM2eng.html. (accessed: 28.05.2020).

[32] Scikit-learn developers. Classifier API Reference, [Online]. Available: https://scikit-learn.org/stable/modules/classes.html. (accessed: 10.06.2020).

[33] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books", *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210, 2015.

[34] A. Canavan, D. Graff, and G. Zipperlen. CALLHOME American English Speech, [Online]. Available: https://catalog.ldc.upenn.edu/LDC97S42. (accessed: 31.05.2020).

6.2 Glossary

accuracy Metric for evaluating classification models. iii, iv, 1, 5, 8, 11, 22–26, 35, 37

AudioSet A large YouTube human labeled dataset for audio events. 1, 8

diarization error rate De-facto standard metric for measuring the quality of a speaker diarization system. iii, 1, 5

i-vector Vector of fixed size which is used to represent speech utterances in a compact way. 1–3

LibriSpeech Large-scale corpus of read English speech which can be used free of charge. iii, iv, 1, 3, 23, 33, 37, 38

overfitting Modeling error which occurs if a function is fitted too close to a limited set of data points. 1, 11

principal component analysis A technique to reduce dimensions while keeping the variation present in the dataset. 1, 39

rectified linear units A non-linear activation function which is popular in deep learning. 1, 10

speaker change detection Aims to find the boundries between speaker turns. 1, 2

speaker clustering The task of differentiating speakers in an audio file. 1, 2

speaker diarization The task of detecting speaker changes and identifying when the same speaker speaks again. iii, iv, 1

speaker identification The task of finding a given speaker among a list of predefined speakers. iii, iv, 1

speaker recognition Umbrella term for speaker verification, speaker identification and speaker diarization. iii, 1

speaker verification The task of verifying whether a given utterance fits a claimed identity or not. 1

spectrogram A visual representation of an audio signal. 1, 3, 7, 16

TensorFlow A end-to-end open source platform for machine learning. 1, 8, 9, 16

transfer learning A strategy which uses stored knowledge gained while solving one problem and uses it for a related problem. 1, 2

Verbmobil II Corpus which contains dialogue recordings developed by the Bavarian Archive for Speech Signals at the University of Munich. iii, iv, 1, 20, 21, 29, 35, 37, 38

VGGish A solely on audio files trained VGG model. 1, 8, 9, 16, 35, 38

Visual Geometry Group A CNN developed by the Visual Geometry Group from the University of Oxford. iii, iv, 1, 3

voice activity detection A technique used to distinguish speech and non-speech parts. 1, 14

VoxCeleb Speech and video corpus which contains over one million utterances extracted from YouTube videos. 1, 3

6.3 List of Figures

3.1 System architecture and data flow overview 14

3.2 Visualization of the sliding window algorithm used to create audio chunks 16

3.3 Four 750 ms long audio chunks visualized as Mel spectrogram and Log-Mel spectrogram 17

4.1 Performance of various classifiers on a range of different training data durations measured by accuracy and DER . 24

4.2 Performance of linearSVC with and without cross-validation on a range of different training data lengths measured by accuracy and DER 25

4.3 Evaluation of the selected chunk sizes based on their achieved scores 27

4.4 Evaluation of the skip factors based on the DER 27

4.5 Evaluation of the different conversation settings based on the DER 29

4.6 Comparison between the calculated and reference speaker changes of an audio recording. The figure shows homogeneous speaker segments based on the best performing conversation of the used corpus. 31

4.7 Comparison between the calculated and reference speaker changes of an audio recording. The figure shows homogeneous speaker segments based on the worst performing conversation of the used corpus. 32

6.4 List of Tables

2.1 VGGish architecture implemented by TensorFlow 10

3.1 An example of the division of an initial sentence into chunks 15

3.2 Parameters applied to the system during the experimental phase 19

3.3 Key metrics of the used Verbmobil II speech corpus 21

4.1 Details from the five best settings of the Verbmobil II corpus according to the DER 30

4.2 Details from the five worst settings of the Verbmobil II corpus according to the DER 32

4.3 Evaluated results on conversations generated from LibriSpeech corpus 34

6.5 Abbreviations

CNN Convolutional Neuronal Network. iii, iv, 1, 3, 8

CV Cross-validation. 1, 11, 20, 25, 26

DCASE2016 Challenge on Detection and Classification of Acoustic Scenes and Event 2016. 1, 3

DER Diarization Error Rate. iii, iv, 1, 5, 6, 12, 19, 20, 22–26, 28–31, 33, 35–39

DNN Deep Neuronal Network. 1–3

kNN k-Nearest Neighbor. 1, 11, 24

NN Neuronal Network. 1, 2

PCA Principal Component Analysis. 1, 39

RBF Radial Basis Function. 1, 19, 20, 24

ReLU Rectified Linear Units. 1, 10

SD Speaker Diarization. iii, 1, 2, 12, 19, 33, 35, 36, 38

SI Speaker Identification. iii, 1, 3, 4, 7, 19, 36, 37

SR Speaker Recognition. iii, 1, 3, 7, 16, 17

SV Speaker Verification. 1

SVC Support Vector Classification. 1, 11, 24–26, 28, 35

SVM Support Vector Machine. 1, 10, 11

VAD Voice Activity Detection. 1, 14, 15, 18, 37, 39

VGG Visual Geometry Group. iii, iv, 1, 3, 8, 16, 35

WebRTC Web Real-Time Communication. 1, 14

Chapter 7

Appendix

7.1 Technical Instructions

Requirements

- Python 3.6 or newer
- Pip
- Virtualenv

Setup environment

The setup instructions as well as the code can also be found in the following Github repository: Link.

- Create a virtualenv: `virtualenv env`
- Activate environment: `source env/bin/activate`
- Install packages `pip install -r requirements.txt`
- Additionally execute `pip install git+https://github.com/beasteers/pumpp@tf_keras`

Data Preparation

Caution: Check that the environment variables in the `.env` file are set correctly.

Verbmobil II Corpus

- The corpus must be present in the following folder structure: `corpora/corpora_german/gxxa`
- Execute `data_preparation/data_preparation_german.py`

This should generate a `current/` folder inside `corpora/`.

LibriSpeech Corpus

- Adjust parameters in `data_preparation/librispeech_preparation.py`.
- Execute `data_preparation/data_preparation_german.py`.

Run Locally

To let the system run locally, you need to set some environment variables. Put a `.env` file in the root folder of the project with the following content:

```
corpora_path=corpora/
corpora_german_path=corpora/corpora_german/
model_persistence_path=models/
results_path=results/
plots_path=plots/
```

Adjust the variables as needed. Also, check the variables set in `params_config.py`

Execute `main.py`.

Containerization

Docker

Everything Docker related is done inside the docker directory.

1. Get the `audioset_weights.h5` file from here: Download link
2. Make `docker\build.sh` executable: `chmod +x docker\build.sh`
3. To build the image, run `./build.sh` inside the docker directory. This creates a tag called `speaker-diarization-with-few-training-samples:latest`.

If you want to build, tag and push the image to docker hub, append the repository name: `./build.sh username/the-repository`

Singularity

To run the container on the ZHAW GPU cluster (or any other cluster with Singularity and Slurm):

1. Transfer the `/corpora` folder to the cluster
2. Pull the image from docker hub and convert it into a Singularity image:
   `singularity pull docker://repo/image-name:latest`
3. Create a slurm batch job to execute the system in the background:

```
sbatch
--ntasks=1
--cpus-per-task=5
--mem=32G
--gres=gpu:1
--output=main.log --error=main.err
```

```
singularity exec --cleanenv --nv
speaker-diarization-with-few-
  training-samples-latest.simg
env
corpora_path=corpora/
corpora_german_path=
  corpora/corpora_german/
model_persistence_path=models/
results_path=results/
plots_path=plots/
python /usr/main.py
```

This command executes the `main.py` on 5 cpus, 32GB memory and 1 GPU. Alter the environment variables according to your needs.

7.2 Result Structure

After executing the system described above, the following folders and files are generated:

models/

This folder contains all persisted models. Each model is named as follows:

`dataset_char_classifier_trainDuration_chunkSize.clf`

plots/

This folder contains all DER plots. Each plot is named as follows:

`dataset_char_classifier_trainDuration_chunkSize_factor.png`

results/

This folder contains a single result.csv file. Each run (parameters, model name, plot name, accuracy and DER) is listed in this file.

diarization_object/
This folder acts as a "temp" folder and is persists all files which are used by the system.

diarization.log
This file contains all log entries higher or equal to the `INFO` level. `DEBUG` levels are only logged to the console.

7.3 Additional Content

The following content will be handed over with this thesis. A zip file named "BA20_ciel_02.zip" with the given structure will be uploaded:

- **Thesis.pdf**: This thesis as a pdf document.
- **directory_explanation.pdf**: This pdf explains the directory structure uploaded.
- **code/**: Contains the source code and documentation developed for this thesis.
- **results/**:
  - **Verbmobil_II**: Contains the result.csv file with all the parameter constellations used with the Verbmobil II corpus.
  - **LibriSpeech**: Contains the result.csv file with the tested conversations based on the LibriSpeech corpus.
- **plots/**
  - **Verbmobil_II**: Contains DER plots for all parameter constellations mentioned in the result.csv file.

- **models/**
  - **Verbmobil_II**: Contains the trained models for all parameter constellation mentioned in the result.csv file.
- **thesis_code/**: Contains the Latex code of this thesis.

7.4 Initial Problem Description

This is the initial problem description provided by Prof. Dr. Mark Cieliebak:

We are working on a solution for automatic transcription of interviews and meetings, i.e. generating text transcripts from audio recordings. One crucial subtask is to identify who is currently talking.

If the participating persons would be known, and if we would have audio samples of each participant, we could train a Speaker Identification (SI) system to solve this task. However, in many settings there are no audio samples available beforehand. In these cases, Speaker Diarization (SD) systems try to detect when speakers change within the audio file, and when the same (anonymous) speaker talks again. As you might imagine, the quality of Diarization is much worse than a well-trained speaker identification system.

In order to circumvent this gap, we want to apply a four-step approach:First, we run SD (and automatic speech recognition) on the audio file and present the results to the user. Then we ask him to identify for each speaker some few utterances (say 1-2 minutes of audio per speaker). Using this data, we want to train an SI system on the fly, and finally apply this system to the entire audio file hopefully getting much better speaker assignments.

Project Goals: The goal of this project is to implement a system to train a Speaker Identification system on the fly, based on few utterances of each speaker in the audio file. This includes the following steps:

* Identify potential technical solutions for SI on few training samples (and few speakers, usually 2-8)

* Implement one such solution

* Apply the solution to test data (e.g. RT corpus) to assess its feasibility and properties (error rate, training duration, required amount of training data etc.)

7.5 Project Management

Darker fields: Planed timeline
Lighter fields: Actual timeline