CZECH TECHNICAL UNIVERSITY
IN PRAGUE

Faculty of Electrical Engineering

BACHELOR'S THESIS

Mohammed Ali

Methods for speaker identification from an acoustic signal.

Program: Electrical Engineering and Computer Science
Specialization: Computer Science

Department of Cybernetics

Thesis supervisors:

doc. Ing. Jan Macek, Ph.D, doc. Ing. Daniel Novak, Ph.D.

BACHELOR'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Ali Mohammed** Personal ID number: **491113**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Electrical Engineering and Computer Science**

II. Bachelor's thesis details

Bachelor's thesis title in English:

Speaker Identification from Acoustic Signal

Bachelor's thesis title in Czech:

Identifikace řečníka z akustického signálu

Guidelines:

The topic of this bachelor thesis is the task of speaker identification from speech audio signal. This task is a subset of the speaker recognition.
1) Perform survey of the techniques and state of the for speaker identification, particularly with focus on data preprocessing, feature extraction, predictive model architecture, training and evaluation.
2) Provide an overview of available reference datasets and evaluate their benefits and limitations with regards to the practical applicability of the speaker identification models.
3) Implement your own speaker identification model based on the state-of-the-art deep neural network architecture for this task. Use reference datasets identified during the survey to train and evaluate the model(s).

Bibliography / sources:

[1] Matějka et al., "Analysis of DNN approaches to speaker identification," 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016, pp. 5100-5104, doi: 10.1109/ICASSP.2016.7472649.
[2] L. Schmidt, M. Sharifi and I. L. Moreno, "Large-scale speaker identification," 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014, pp. 1650-1654, doi: 10.1109/ICASSP.2014.6853878.
[3] Li, R., Jiang, J.-Y., Wu, X., Hsieh, C.-C., Stolcke, A. "Speaker Identification for Household Scenarios with Self-Attention and Adversarial Training," 2020, Proc. Interspeech 2020, 2272-2276, doi: 10.21437/Interspeech.2020-3025
[4] C. Kumar, F. ur Rehman, S. Kumar, A. Mehmood and G. Shabir, "Analysis of MFCC and BFCC in a speaker identification system," 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), 2018, pp. 1-5, doi: 10.1109/ICOMET.2018.8346330.
[5] R. Jahangir et al., "Text-Independent Speaker Identification Through Feature Fusion and Deep Neural Network," 2020, in IEEE Access, vol. 8, pp. 32187-32202, 2020,

Name and workplace of bachelor's thesis supervisor:

doc. Ing. Daniel Novák, Ph.D. Analysis and Interpretation of Biomedical Data FEE

Name and workplace of second bachelor's thesis supervisor or consultant:

Date of bachelor's thesis assignment: **09.02.2022** Deadline for bachelor thesis submission: **15.08.2022**

Assignment valid until: **30.09.2023**

doc. Ing. Daniel Novák, Ph.D. Supervisor's signature	prof. Ing. Tomáš Svoboda, Ph.D. Head of department's signature	prof. Mgr. Petr Páta, Ph.D. Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

. _____
Date of assignment receipt

_____
Student's signature

Declaration of Independent Work

I hereby declare that this bachelor's thesis is the product of my own independent work and that I have clearly stated all information sources used in the thesis according to Methodological Instruction No. 1/2009 – "On maintaining ethical principles when working on a university final project".

In Prague on .....

Mohammed Ali

Acknowledgement

I would like to acknowledge and give my warmest thanks to doc. Ing. Jan Macek, Ph.D, who assisted me in writing this thesis. Would also like to extend my acknowledgment to doc. Ing. Daniel Novak, Ph.D, who gave me advice regarding the overall structure of the thesis. I would also like to acknowledge MAMA AI for their support and guidance for this thesis.

I would also like to give special thanks to my family for constantly extending their support throughout my entire time at the university.

Abstract

The work done in this thesis primarily deals with the description of a Speaker Identification system, a type of Speaker Recognition system. Theoretically, It mainly focuses on the different configurations, acoustic analysis, and methods, where the main attention is given to GMM-UBM with i-vector, and current baseline deep learning methods, namely, X-vector and ECAPA-TDNN.
The practical part deals with implementing a Speaker identification pipeline based on the specific task assigned by MAMA AI. It consists of implementing a pipeline using an open-source speech toolkit, SpeechBrain. Pre-trained models were tested for various test cases the best-performing model was selected. The model has then experimented under varying scenarios namely, performance against varying lengths of audio samples, performance against noisy data (non-intelligible and intelligible), performance against various languages, and performance against artificially generated audio samples. The selected model, ECAPA-TDNN, performed excellently for all of these scenarios, with the lowest IR (%) being no less than 70% (apart from the final experimentation, where IR values were lower, but is favorable based on the experimentation circumstances) and was concluded to be used in the final Speaker Identification pipeline.

List of Abbreviations

1. MFCC - Mel-Frequency Cepstral Coefficients
2. LPC - Linear Prediction Coefficients
3. PLP - Perceptual Linear Prediction
4. DFT - Discrete Fast Fourier Transform
5. DCT - Discrete Cosine transform
6. DNN - Deep Neural Network
7. GMM - Gaussian Mixture Model
8. UBM - Universal Background Model
9. VQ - Vector Quantization
10. DTW - Dynamic Time Warping
11. JFA - Joint Factor Analysis
12. LDA - Linear Discriminant Analysis
13. CDF - Cosine Distance Formulation
14. ECAPA-TDNN - Emphasized Channel Attention, Propagation and Aggregation in a Time Delay Neural Network
15. VAD - Voice Activity Detection
16. SE-Res2Block - 1-Dimensional Squeeze-Excitation Res2Blocks
17. MS-SNSD - Microsoft Scalable Noisy Speech Dataset
18. EER - Equal Error Rate
19. FAR - False Acceptance Rate
20. FRR - False Rejection Rate
21. VOiCES - The Voices Obscured in Complex Environmental Settings
22. SIVA - Speaker Identification and Verification Archives
23. NIST SRE - National Institute of Standards and Technology Speaker Recognition Evaluation
24. ELSDSR - English Language Speech Database for Speaker Recognition
25. TED-LIUM - Technology, Entertainment, Design - Laboratory of Informatics of Le Mans University
26. LDC - Linguistic Data Consortium
27. ELRA - European Language Resources Association

List of Figures

1) Comparison between closed and open set configuration……11
2) Structure of the human throat, where various parts are labeled[4]……12
3) An example for a speech signal, uttering the words “will we ever forget”[6].....13
4) Mel-filterbanks [35]…..……..……15
5) Pipeline for a GMM-UBM system……..19
6) DNN model for d-vector extraction[25]…….……20
7) DNN model for the embedding extraction method [24].....…….22
8) The topology of the ECAPA-TDNN architecture [26].....23
9) The topolyg of the SE-Res2Block [31].....24
10) Proposed pipeline for the speaker identification system…….26
11) Example of a FAR vs. FRR graph [33].....29
12) Threshold estimation results for the selected model……...36
13) IR (%) values for 3 seconds (left) and 10 seconds (right) length utterances……..39
14) Top Confidence (%) values for 3 seconds (left) and 10 seconds (right) length utterances……40
15) IR (%) values for 3 seconds (left) and 10 seconds (right) length utterances……..40
16) Top Confidence (%) values for 3 seconds (left) and 10 seconds (right) length utterances……40
17) IR (%) values for 3 seconds (left) and 10 seconds (right) length utterances……42
18) Top Confidence (%) values for 3 seconds (left) and 10 seconds (right) length utterances……42
19) IR (%) values for 3 seconds (left) and 10 seconds (right) length utterances……..43
20) Top Confidence (%) values for 3 seconds (left) and 10 seconds (right) length utterances…….43
21) IR (%) values for 3 seconds (left) and 10 seconds (right) length utterances……...44
22) CR (%) values for 3 seconds (left) and 10 seconds (right) length utterances…….45
23) IR (%) values for 3 seconds (left) and 10 seconds (right) length utterances……..46
24) CR (%) values for 3 seconds (left) and 10 seconds (right) length utterances……46

List of Tables

1) X-vector architecture [31]……23
2) Summaries for the surveyed datasets……...31
3) Distribution of speakers for the test datasets……...35
4) Matric results tested on mixed data……35
5) File distribution for experiment 1……...37
6) Results of the experiment 1……38
7) File distribution for the experiment 2……39
8) File distribution for the experiment 3……..42
9) File distribution for the experiment 4……..44
10) File distribution for the experiment 5……..45

Table of Content

1. Introduction....10
2. Speaker Recognition....10
   - 2.1. Speaker Identification....10
     - 2.1.1. Applications of a speaker identification system....10
     - 2.1.2. Open-set versus closed-set....11
     - 2.1.3. Text-dependant vs Text Independent....11
   - 2.2. Acoustic Analysis....12
     - 2.2.1. Biology of human voice production....12
     - 2.2.2. Representation of audio signals....12
     - 2.2.3. Feature Extraction....14
       - 2.2.3.1. Mel-Frequency Cepstral Coefficients....14
       - 2.2.3.2. Filterbanks....15
3. Survey of methods....15
   - 3.1. History of methods....16
     - 3.1.1. Gaussian Mixture Models....16
     - 3.1.2. Universal Background Model (UBM) and GMM-UBM....18
     - 3.1.3. Supervectors and identity vector....19
     - 3.1.4. Deep-vector....20
   - 3.2. Current Baseline Methods....21
     - 3.2.1. X-vector....22
     - 3.2.2. TDNN-ECAPA....23
4. Implementation....25
   - 4.1. Aim....25
   - 4.2. Technical information....25
   - 4.3. SpeechBrain Toolkit....25
   - 4.4. The architecture of the speaker identification system....26
     - 4.4.1. Implementation structure....27
5. Experimentation....28
   - 5.1. Experimentation Setup....28
     - 5.1.1. Decision Making....28
     - 5.1.2. Evaluation of performance....28
     - 5.1.3. Threshold Estimation....29
   - 5.2. Survey of publicly available datasets....29
   - 5.3. Selection of the best-performing model....34
     - 5.3.1. Setup....34
     - 5.3.2. Results....34
     - 5.3.3. Assessment....35
     - 5.3.4. Threshold Estimation....35
   - 5.4. Optimization Task....36
     - 5.4.1. Datasets utilized....36
     - 5.4.2. Performance against variation in the length of the audio samples....37
       - 5.4.2.1. Setup....37

5.4.2.2. Results....37
5.4.2.3. Assessment....38
5.4.3. Performance against noisy data (non-intelligible)....38
5.4.3.1. Setup....38
5.4.3.2. Results....39
5.4.3.3. Assessment....41
5.4.4. Performance against noisy data (intelligible)....41
5.4.4.1. Setup....41
5.4.4.2. Results....42
5.4.4.3. Assessment....43
5.4.5. Performance against multiple languages....44
5.4.5.1. Setup....44
5.4.5.2. Results....44
5.4.5.3. Assessment....45
5.4.6. Performance against artificially generated voice samples....45
5.4.6.1. Setup....45
5.4.6.2. Results....46
5.4.6.3. Assessment....46
6. Conclusion....47
7. References....48

1. Introduction

Biometric recognition refers to identifying a person based on the unique traits present in all people. However, amongst the popular recognition methods such as fingerprint scanners and face recognition, the most robust and easily acquired form is speech recognition. We refer to such methods/systems as Speaker Recognition [1]. Not only is speaking one of the fundamental parts of human interaction, but a person's voice can express a person's individual traits due to varying auditory factors such as the vocal tract resonance, individual pitch, and other unique "quirks" people may have such as various manners of speaking and regionally dependant accents. Speaker Recognition also only requires no more than a simple microphone, adding more to its simplicity. Such technology is useful in cases such as bank transactions, and other auditory banking tasks, security and authorization purposes, and even for surveillance [1], [5].

2. Speaker Recognition

Speaker recognition generally refers to two separate tasks: Speaker verification and Speaker identification. Speaker Verification involves identifying a speaker's voice with their claimed identity. In contrast to this, Speaker Identification involves using a speaker's voice sample and comparing with all stored samples in the voice database, which results in higher time complexity. As such, Speaker Verification systems are utilized strictly for situations where single user verification is needed such as bank authorization given their name. This thesis is focused on Speaker Identification, and it aims to explore the process behind it, including the audio processing and the state-of-the-art pattern recognition and machine learning techniques used for classification [3],[5] and an implementation of such a system based on the survey.

2.1. Speaker Identification

Speaker identification is a process that can be defined as a system that utilizes the voice characteristics of a speaker that best matches a person from a given pool of known speakers. The complexity of such a process depends on the number of the already stored known speakers in the system. This system can be derived into two parts: open-set and closed-set identification. It can then be further derived into text-dependent and text-independent [1],[2].

There are various steps involved in the process of Speaker Identification. Firstly, it contains two major parts, training, and testing. In the training phase, all the voice samples from known users in the pool are used to train specific "voice models" for each speaker. When the system is given a voice sample of an unknown person, it calculates the degree of similarity with each model present in the database and identifies the person based on the highest degree selected [2],[5].

2.1.1. Applications of a speaker identification system

There are several use cases for a speaker identification system, but mainly they are utilized in the area of authentication, surveillance, and forensics [2]. Users are able to authenticate themselves using speaker identification systems, depreciating previous methods such as PINs or passwords and providing a more secure and robust method using the unique attributes in one's voice [2].

While collecting data using auditory samples, Speaker Identification technologies may be useful for filtering out specific people of interest, thereby making their discovery significantly more efficient [3].
Speaker Identification Systems may also be used for simple forensics in criminal cases, making identifying criminals easier and assisting in evidence production at the court [3].

2.1.2. Open and closed set settings

Under a closed-set setting, the system always results by matching the unknown voice sample with at least one model present in the database; it is expected that under this type of setting, speakers from outside the domain of the speaker database are not expected to appear. Therefore, such a setting is useful for a small group of people. In broader areas with more people, open-set settings are preferred [2].

Under an open-set setting, after calculating the degree of similarity of the unknown voice sample, the degree is then compared with a threshold value and only permitted if it is below it [2]. Refer to Figures 1 and 2 for visualization.

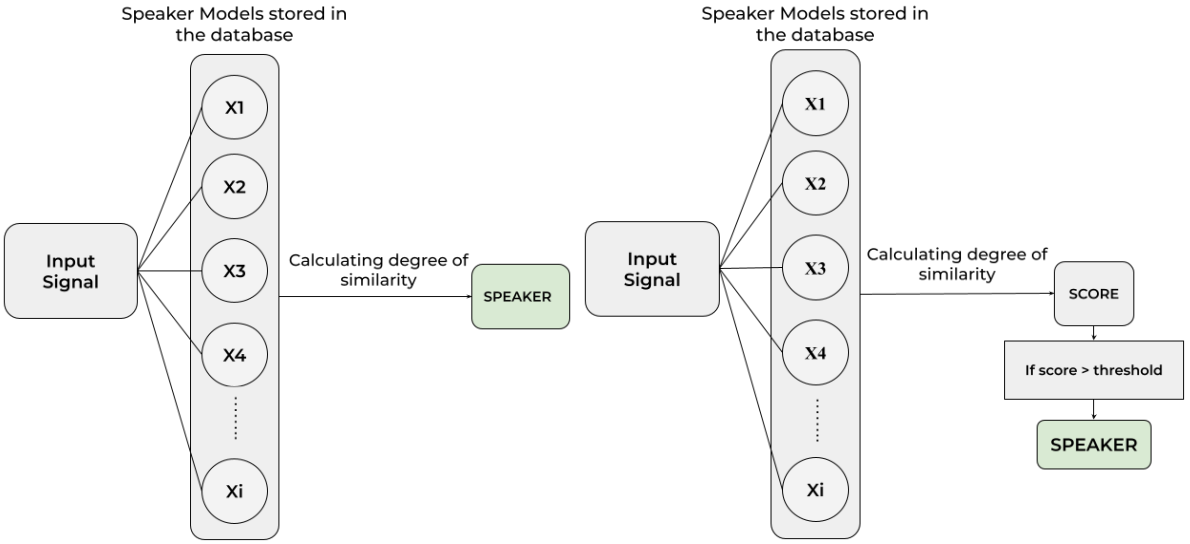

Figure 1: Closed-set [right] and Open-set [left]

2.1.3. Text-dependent and text-independent

In the case of text-dependent Speaker Identification, an utterance is particularly requested when training by the speaker, and during testing, the utterance must be repeated for identification by the unknown speaker. This is generally a more reliable but more complex method compared to the text-independent method since it requires the need for a speech recognition system as well [2],[1].

Text-independent Speaker Identification primarily relies solely on the unique voice features extracted from a voice sample given by the unknown speaker and thus does not require the unknown speaker to adhere to anything specific. However, during training, more rigorous and higher-quality voice samples may be required to ensure the correct classification during testing. The voice sample provided during testing by the unknown speaker generally also is required to be of higher quality and more detailed than for the case with text-dependent Speaker Identification [2],[1].

2.2. Acoustic Analysis.

2.2.1 Biology of human voice production.

Sound is produced through vibrations, which causes the air molecules to oscillate which causes changes in the air pressure and produces a wave. Speech produced by us humans is due to the vibrations in our vocal cords. The contraction and relaxation of the muscles in the vocal cords and the wideness of the slit present in the middle are the factors that allow us to produce intricate sounds by simply moving those muscles as needed [4].

There are three parts that are involved in sound production in humans, namely, the lungs, the larynx (voice box), and the articulators. Vocal cords are a reference to the folds present in the voice box. The lungs, acting as “pumps”, push out air which is then moderated by the muscles present in the larynx. This results in sound with various frequencies, pitches, and resonances. Due to the involvement of three various parts, each and every slight variation, be it the frequency or the length of the vocal cords present, produces different sounds which is the factor utilized for speaker identification [4].

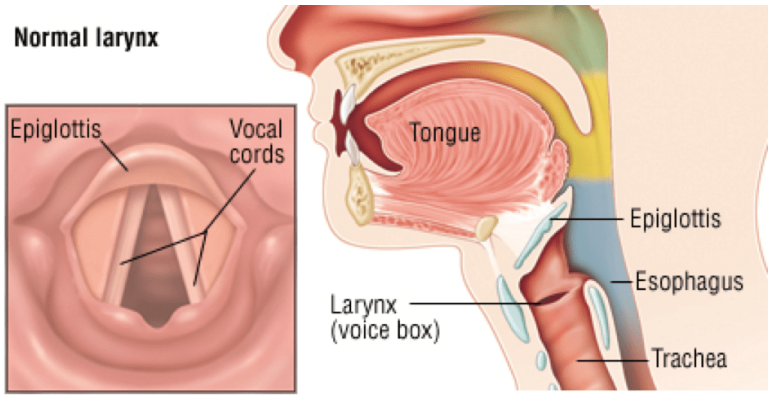

Figure 2: The structure of the human throat. [4]

The varying shapes of the vocal cords are the main factor for the uniqueness present for everyone, and this feature, namely the vocal cords resonance (also called formants [5]) and “Thus, the vocal tract shape can be estimated from the spectral shape (e.g., formant location and spectral tilt) of the voice signal.” [5].

2.2.2. Representation of audio signals

Sounds that are produced and recorded are analog signals. They can be classified into many various types, such as periodic and aperiodic, simple and complex, continuous and discrete, and so on. For the digital analysis of audio, the conversion of the received signals into the digital form is a necessary step for Speaker Identification. This is generally accomplished by an Analog-Digital-Converter (ADC). ADCs are systems that take in analog signals, such as audio signals from a microphone, and convert them into digital signals to process on a computer [6].

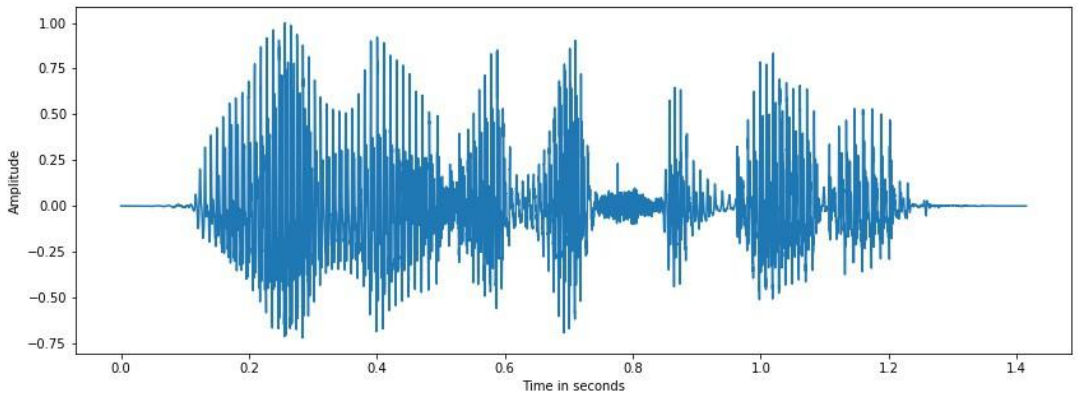

Figure 3: The speech signal of the utterance “will we ever forget” [6].

The most effective way to represent a given audio signal would be as feature vectors. The goal of the front-end part of the Speaker Identification system is to isolate effective information that represents the human voice and represent the signal at a low dimensional level [8].
Audio signals are generally presented as spectrograms before going through the backend. To obtain these spectrograms, a few pre-processing methods are commonly applied to the audio signal [1]. The most common procedure is as follows:

1. **Digitization:** Audio signals are present as continuous in nature, therefore we need to convert them to discrete for further analysis. Quantization is a commonly used technique where the signal is divided into finite intervals with the goal of mapping continuous infinite values to a set of discrete values [8].

2. **Pre-emphasis:** Simply put, this part of the process emphasizes the higher frequencies in a given waveform by amplifying them. This is done by using a simple high-pass filter,
$$y(t) = x(t) - 0.97x(t - 1) \quad (1)$$
Where x(t) is the input signal and y(t) is the resulting signal and the value of 0.97 is commonly used as a coefficient. [1]

3. **Framing:** The division of the waveform into fixed segments is known as framing. The typical duration is 25 ms, each one generated every 10 ms [1]. Each frame is also multiplied with a windowing function to ensure a smooth and artifact-free spectrum [1]. The most commonly used one is Hamming window:
$$\omega(n) = 0.54 - 0.46cos(2\pi n \div N - 1) \quad (2)$$
Where N is the number of data in each frame and n = 0,1,2… N-1 [8]
4. **Voice Activity Detection:** To emphasize on speech data and also improve CPU processing efficiency, silent parts from the audio waveform are removed. Experiments [8] suggest that zero-crossing rate and short-time energy are the two useful ways to distinguish silence in a given audio waveform.

2.2.3. Feature Extraction

The aim of this part of the system is to convert the provided speech signal into a parametric form for further processing and analysis. We need to extract speaker-specific features from each frame [1]. The most common feature is the Mel-Frequency Cepstral Coefficients (MFCCs), but others such as Linear Prediction Coefficients (LPCs), which are directly derived from the speaker model, Perceptual Linear Prediction (PLP) coefficients, are utilized [1][5][6][8].

According to research in [7], the main basis for the extraction of features was the variations in pitch, and in addition to that, other parameters such as individual word durations, the spectral gradient, and the voice source, which represent the glottal-based vocal effects, the voice onset time, which is the length of time between stop release and voice onset. Models based on the cepstrum, such as the linear prediction cepstral coefficients (LPCCs) and the MFCCs performed better than other coefficients [7].

2.2.3.1 Mel-Frequency Cepstral Coefficients

MFCCs are commonly utilized as they represent the auditory perception of the human ear really well [8]. The resultant feature also contains a large number of coefficients which further improves the performance of a speaker recognition system. The use of the Mel-scale highly assists it to be one of the most popular feature extraction methods used today.

The procedure of obtaining MFCCs overlaps with the transformations mentioned in the previous section. The algorithm generally involves framing, windowing, Fast Fourier Transform, Extracting Mel filters, Taking logs of energies from N filters, and finally, a Discrete Cosine Transform at the very end [8].

1. **DFT spectrum:** This step allows the conversion from the time domain to the frequency domain. For each frame, Discrete Fourier Transform (DFT) is applied to obtain the magnitude spectrum.
   The algorithm is given by:
   $$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{\frac{-j2\pi nk}{N}};\ 0 \le k \le N - 1 \quad (3)$$
   Where N is the number of points used for the computation of DFT [9].
2. **Mel-scale:** The Mel-scale is used to perceive sound the same way humans do. As the human hearing does not perceive pitch linearly, this scale was introduced to compensate for that and allow for a better representation. It can be approximated by:
   $$f_{Mel} = 2595\, log_{10}\left(1 + \frac{f}{700}\right) \quad (4)$$
   Where f is the frequency in Hz and the result is the Mel frequency.
   While filter banks can be obtained from both the time and frequency domains, for the calculation of MFCCs, they are commonly obtained from the frequency domain [9].
   The mel-filters are obtained by multiplying the magnitude spectrum from equation 3, X(k) by each of the triangular Mel weighing filters:
   $$s(m) = \sum_{k=0}^{N-1}\left[|X(k)|^2 H_m(k)\right];\ 0 \le m \le M - 1 \quad (5)$$
   "where M is the total number of triangular Mel weighting filters [5, 6]. Hm(k) is the weight given to the kth energy spectrum bin contributing to the mth output band and is expressed as" [9, page 2]:

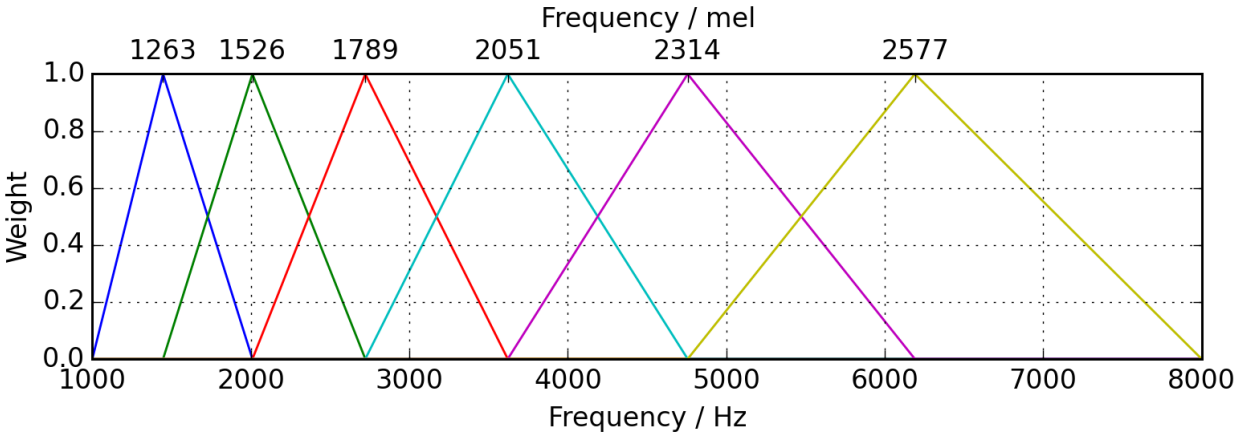

Figure 4: Mel filterbanks [35].

3. **Discrete Cosine transform (DCT):** The final step is to perform a discrete cosine transform to the obtained Mel frequency coefficients. According to [9] most of the signal is represented by a small number of MFCC coefficients, which can be extracted by "ignoring the higher-order DCT components"[9, page 2]. This can be calculated by:

$$c(n) = \sum_{m=0}^{M-1} log_{10}(s(m))cos\left(\frac{\pi n(m-0.5)}{M}\right); \; n = 0, 1, 2, \ldots\ldots C - 1 \qquad (6)$$

Where C represents the number of MFCCs and c(n) represents the resultant cepstral coefficients [9]. Typically, only 12 MFCCs are extracted [1] however, values such as 20 or higher aren't very uncommon to observe in many publications regarding speaker recognition systems.

2.2.3.2 Filterbanks

Although MFCCs are a widely popular choice, using simply the Filterbank coefficients in the Step 2 in the previous sections is becoming a popular choices of recent.A publication [31] discusses that the final DCT (Step 3 in the previous section) might not be a necessary step when dealing with Deep Neural Networks. This is due to the fact that the filter banks computations are the values that represent the human perception of signals, while the final DCF step is mainly done to decorrelate the filter bank coefficients, a step needed for a method such as the popular Gaussian Mixture Model, but not with Deep Neural Network as they are less susceptible to highly correlated input. It should also be noted that it is a linear transformation, thereby discarding non-linear information from the signal which may help deliver better performance.

3. Survey of methods

This section aims to deliver a survey on the methods used for speaker identification, which will be further utilized in the implementation part of this thesis.
While there have been many different techniques used with various machine learning algorithms, the most successful usage is:

1) Training a background speaker-independent model, trained with a fairly large and diverse corpus to serve as an "embedding extractor".
2) Extracting speaker embeddings from said "embedding extractor" for a particular audio sample of a unique speaker for either storage in the enrollment phase or testing in the test phase.

3.1. History of methods

While this thesis will primarily concern itself with the more recent deep learning modeling methods in the implementation section, I believe it is imperative to describe how the idea of a speaker embedding extraction technique came to be used with the baseline machine learning techniques used today.
Over time, many modeling techniques have been utilized to produce the background model. The goal is to obtain a speaker-independent model, which emphasizes the features of the human voice.
They can be largely split into two parts, generative and discriminative.
A popular, depreciated technique used was Vector Quantization (VQ), which famously utilizes the K-means algorithm [7]. This type of model follows an encoder-decoder type structure, where the encoder encodes a given feature vector, which is usually based on cepstrum models defined in section 3.4, and quantizes them into a smaller subset, called a codebook [10]. Each element at index value i is called a "codeword"[10] which represents the centroids for the particular codebook. The decoder also contains the same codebook, and given a certain index, outputs the vector in a lookup-table type fashion. For an SR task, a codebook is generated for each speaker during the training session, and in the testing phase, each input is vector quantized using the trained codebooks, and the distance between the resultant vectors and the codewords from known speakers is calculated. The codebook which has the least distance is chosen and the speaker corresponding to that is outputted as a result. This can be applied in the case of speaker verification, where the vector is compared against a particular codebook, or in the case of speaker identification where the vector is compared against every codebook in the database. VQ is an example of a template model, another example being dynamic time warping (DTW), which was popular prior to the emergence of stochastic models, in particular, the Gaussian Mixture Model (GMM).

3.1.1 Gaussian Mixture Models (GMM)

Originally proposed by Reynold [11], GMM quickly became one of the most popular and long-running modeling methods for the Speaker Recognition task. The feature vectors extracted are expected to follow a Gaussian distribution, and each distribution, also known as a mixture model, is unique to each speaker trained in the training phase.

Other clustering techniques such as K-means follow a "hard clustering" format, where it always points a data point towards one cluster or another, hence no probabilistic estimations. GMM solves the problems where clusters overlap each other, see Fig 3, by using a probabilistic estimate for some unknown data point. Each cluster is referred to as a "Mixture model". [12] For the task of speaker identification, each model represents the vocal characteristics of a particular speaker,

Each Mixture model is represented by three distinct parameters [11]:
μ - the mean, defining the center.
$\sum$ - the covariance matrix, which defines the "shape" of the cluster
w_k - Some mixture weight

Where it is true for all mixture weights belonging to N mixture models:

$$\sum_{k=1}^{M} w_k = 1 \tag{7}$$

Therefore, the Gaussian density function can be formulated as

$$p(x \mid \lambda) = \sum_{k=1}^{M} w_k N(x \mid \mu_k, \Sigma_k) \tag{8}$$

Where M is the total number of component densities, for some d-dimensional vector x , w_k refers to the mixture weight, and the multivariate gaussian function is given by:

$$N(x \mid \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{d/2}\left|\Sigma_k\right|^{1/2}} exp\left\{-\frac{1}{2}(x - \mu_k)'\Sigma_k^{-1}(x - \mu_k)\right\} \tag{9}$$

Where μ_k is the mean vector and Σ_k is the covariance matrix for mixture model k, and λ represents the combination of all mixture models and their respective components.

The training phase of a GMM-based architecture for some unknown speaker $\boldsymbol{\theta}$ is a parameter estimation task with the goal of estimating parameters for model $\boldsymbol{\theta}$. There are numerous ways to accomplish this task, but according to [11], the most common and well-established method is maximum likelihood estimation.

Given some training vector X, the estimation formula is as follows:

$$p(X \mid \boldsymbol{\theta}) = \prod_{i=1}^{N} p(x_i \mid \boldsymbol{\theta}) \tag{10}$$

Where x_i belongs to a given set of training vectors X = $\{x_1, x_2 \ldots\ldots x_N\}$.
An iterative algorithm, namely, the expectation-maximization (EM) algorithm is used, with some initial model set as $\boldsymbol{\theta}_t$ which is utilized to generate a new model $\boldsymbol{\theta}_{t+1}$ until convergence, under the restriction [11]:

$$p(X \mid \boldsymbol{\theta}_{t+1}) \geq p(X \mid \boldsymbol{\theta}_t) \tag{11}$$

Where t represents the iteration, and therefore, can also be terminated by some convergence threshold *t'*.
Each iteration, the mean, the variances, and the mixture weight for the model are updated using re-estimation formulas [11].
Means:

$$\overline{\mu}_k = \frac{\sum_{i=1}^{N} p(k \mid x_i, \boldsymbol{\theta}) x_i}{\sum_{i=1}^{N} p(k \mid x_i, \boldsymbol{\theta})} \tag{12}$$

Variances:

$$\overline{\sigma}_k^2 = \frac{\sum_{i=1}^{N} p(k \mid x_i, \boldsymbol{\theta}) x_i^2}{\sum_{i=1}^{N} p(k \mid x_i, \boldsymbol{\theta})} - \overline{\mu}_i^2 \tag{13}$$

And the mixture weight is calculated by:

$$\overline{w}_k = \frac{1}{N}\sum_{i=1}^{N} p(k \mid x_i, \boldsymbol{\theta}) \tag{14}$$

The a posteriori probability for some speaker class k is given by:

$$p(k \mid x_i, \boldsymbol{\theta}) = \frac{w_k N(x_i \mid \mu_k, \Sigma_k)}{\Sigma_{j=1}^{K} w_j N(x_i \mid \mu_j, \Sigma_j)} \quad (15)$$

Where K is a pre-selected order of the mixture.
Given a set of M speakers in some database where M = $\{\lambda_1, \lambda_2, \ldots. \lambda_M\}$ and given a sequence of test vectors X = $\{x_1, x_2, \ldots.. x_N\}$, for the task of identification in the testing phase, the classification can be performed by:

$$y = Argmax \sum_{i=1}^{N} log\, p(x_i \mid \lambda_k) \quad (16)$$

3.1.2 Universal Background Model (UBM) and GMM-UBM.

While the GMM alone is a powerful statistical technique, training a specific GMM for each speaker in a database can be resource hungry and requires a lot of speaker-specific data [13]. To combat this issue, a general speaker-independent model named the Universal Background Model (UBM) was formally introduced in the publication [15].

UBM is a model created using a large quantity of data with the goal of representing the distribution of speech characteristics. It can be, and is generally, gender-specific and is trained using the EM algorithm. [14] Similar to GMM, speaker-specific models are extracted but rather than initiating an iterative algorithm from scratch, the model parameters are estimated using Bayes adaptation techniques using the UBM model. This has proven to perform better than the usual GMM approach and also allows for faster backend processing [15].

The pipeline for a GMM-UBM follows the three-phase structure described in earlier sections:

1) Training: A large UBM is trained for usually around 512 to 2048 mixtures, using the EM algorithm described in section 4.1.1, and trained parameters are outputted.
2) Enrollment: For each speaker to be enrolled, speaker-dependent model parameters are estimated using Bayes adaptation methods.
3) Testing: Scoring is done for the Speaker Identification system.

The process of generating the UBM model is similar to the speaker-dependent model training described in section 4.1.1. The adaptation in the enrollment phase is done using the Maximum a Posteriori Adaptation (MAP) method [15]. The trained parameters obtained from the UBM are used as initial values and using MAP, the mean, variances, and mixture weight for specific speakers using the speech vectors provided for the enrollment are calculated. [15] After the speaker-dependent models are generated, the identification is done as described in equation (16).

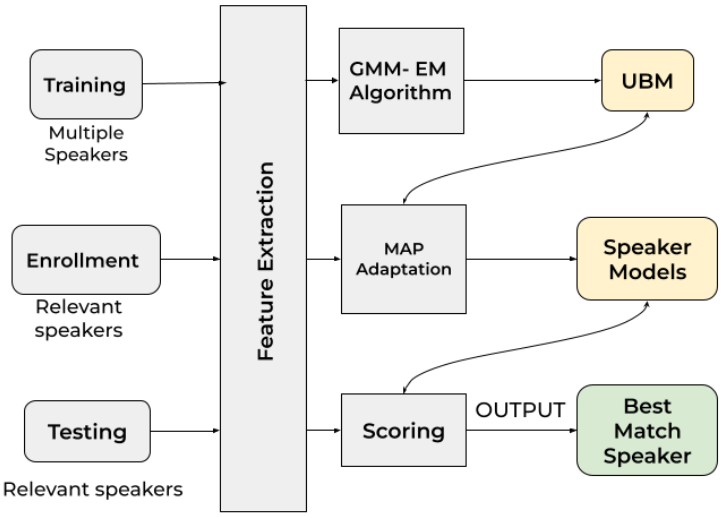

Figure 5: Pipeline for GMM-UBM system

3.1.3 Supervectors and identity vector

GMMs provide a frame-level representation of an audio sample, which tends to disregard the channel/session distortions present in various speech signals. The channel variability can be defined as the differences in the utterances belonging to the same speaker but in different recording sessions. Research [18] suggests the use of latent factor analysis compensates for the variabilities. One such approach is to stack the means of the mixture components [18, 19].

The model based on Joint Factor Analysis (JFA) accomplishes this by separating the speaker and channel variabilities into two subspaces [17]. The main way of obtaining this model is by mapping each GMM component in a model adapted from the UBM, using a kernel mapping function which then results in fixed-length vectors. These vectors are then combined together to obtain a high-dimensional supervector [19]. It is defined as [17]:

$$s = m + Vy + Ux + Dx \tag{17}$$

Where m is a speaker- and channel-independent supervector generated from the UBM, V, U, and D are eigen speaker, eigenchannel and residual matrices respectively, and x,y,z are speaker, channel and residual factors, respectively. The process of utilizing this model was to first estimate the subspace matrices (V, U, and D matrices) using sufficient labeled data, and then approximate the speaker and channel factors (x,y, and z) forming a speaker-dependent vector for a particular speaker utterance. The unique feature of the JFA model is that the speaker subspace is isolated from the channel subspace. However, in an experiment described in [17], the channel factors were found to contain important speaker-related information as well, which may dwarf the performance of the SR system. As a result, a new low-dimensional model based on "total variability space", which combined both the speaker and channel subspace, was introduced, named the “i-vector” [17]. Due to its low dimensionality and post-extraction channel compensation (as total variability

space is susceptible to channel distortions) [17], this model remained the best performing model for the task of speaker recognition for a long time. The model is defined as:

$$s = m + Tw \tag{18}$$

Where m is a speaker- and channel-independent supervector generated from the UBM, T is the total variability matrix obtained using a similar process as the eigenspeaker matrix described in the JFA model, with the difference being the assumption that every utterance from a single speaker belongs to different speakers and w is the total variability factor [17].

As described in [17], The total factor w is obtained using Baum-Welch statistics, which are extracted using the UBM. Given a speech utterance u and an UBM λ with C components defined in D dimension:

$$w = (I + T^t\Sigma^{-1}N(u)T)^{-1}T^t\Sigma^{-1}F(u) \tag{19}$$

Where, where Σ of dimension $CD \times CD$ represents the covariance matrix modeling the residual variability, $N(u)$ and $F(u)$ being the Baum-Welch statistics of a given utterance u, in particular, $N(u)$ is a $CD \times CD$ diagonal matrix having the diagonal blocks as N_cI where c = {1, 2, . . . , C} matrix of size D and $F(u)$ is of dimension $CD \times 1$ and is obtained by stacking up all first-order Baum-Welch statistics $F'c$ and T is the total variability matrix.

To handle the channel distortions, two post-processing techniques were tested in [17], namely, the within-class covariance normalization (WCCN) and linear discriminant analysis (LDA), with the advantage of LDA being the minimization of the removal of relevant speaker information and maximization of the inter-class variation, while disregarding the directions in space, compared to WCCN, which compensates for inter-class variation while preserving the directions. The best results were found when LDA and WCCN were combined [17]. For classification, two methods, cosine distance formulation (CDF) and Probabilistic Linear Discriminant Analysis (PLDA) were used, with CDF producing the best results [17].

3.1.4. Deep-vector

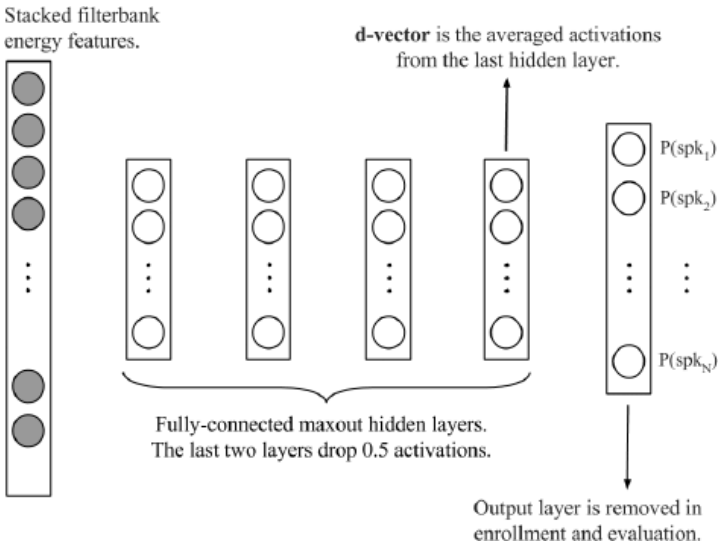

Figure 6: DNN model for d-vector extraction [25]

Deep vector or more commonly referred to as D-vector is one of the very first DNN-based embeddings to be introduced [21]. While inspired by the i-vector, they are fundamentally different. As described in the previous sections, i-vectors are obtained from an unsupervised, generative model, while d-vectors are obtained from a supervised, discriminative model, specifically a neural network. These vectors are better performing than the former [25] and have various advantages, namely,

- it represents speaker-relevant information by reducing speaker-irrelevant variance,
- Can be produced given very short utterances.

The architecture trained in order to extract the d-vectors is given in Figure, where the inputs are frame-level 40-dimensional log filterbank energies stacked, and the resultant vector is N-dimensional, where N corresponds to the number of speakers in the training data, and the only non-zero component in the vector refers to the identity of the speaker [25]. Then, every frame from a given utterance belonging to a speaker from outside the training dataset, is propagated through the trained neural network in a feedforward fashion, and the activations in the final activation layer are extracted as the new representation of the speaker, that is, the d-vector. The extracted vector is then stored in some database, serving as a speaker-dependent model. The evaluation is then performed in a similar fashion as the i-vector, using some similarity matching technique present in the backend part of the speaker recognition system.

In the experiment performed in [25], The background DNN model is trained using the dropout strategy, which prevents overfitting by deploying regularization techniques. Only the last two layers are configured to drop 0.5 activations. Rectified linear units were used as activation functions, and 0.001 as the value for the learning rate with exponential decay set to 0.1 for every 5M steps. The final model consisted of 600,000 parameters which are comparable to “the smallest baseline i-vector system”[25]. The results mentioned that the d-vector system outperformed the i-vector in both noisy and clean environments, and was also quite robust to additive noise.

3.2. Current Baseline Methods

Over the years, due to advancements made in the field of machine learning and the increasing performance of modern computer hardware [7], Deep Neural Networks (DNNs) have become quite popular, especially in the field of image and speech recognition, and is an active area of research in the field of speaker recognition. In recent years, many speaker recognition competitions such as the NIST Speaker Recognition Evaluation (NIST SRE), VoxCeleb Speaker Recognition Challenge, etc have reported their best performing models to be based on DNN architectures.
The main point of inception for the utilization of DNNs for the task of speaker recognition originally came from their feature extraction capabilities [21], which were applied alongside the i-vector method, yielding impressive results which outperformed the traditional GMM-UBM model [7, 21]. Motivated by this success, numerous speaker recognition methods based on deep neural networks were introduced, delivering state-of-the-art performance even under challenging conditions [22, 23]. Inspired by the i-vector method, deep speaker embeddings-based models such as the d-vector and x-vector were introduced [21, 24, 25], which further gave inspiration to models as such the “Emphasized Channel Attention, Propagation and Aggregation in a Time Delay Neural Network” (ECAPA-TDNN) [23, 26]. However, most of these implementations were built on top of baseline architectures such as the time delay neural network, residual networks, and convolutional neural networks.

3.2.2. X-vector

A new model was introduced in [24]. Compared to the frame-level representation for the d-vector, This model aims to represent the entire utterance [24]. The extraction of the embedding for an enrolled speaker is similar to the way described for the d-vector, with the differences lying in the architecture of the model and the specific layer that is extracted. This architecture builds upon the baseline time-delay neural network. Figure 7 describes the architecture, where the DNN is split into groups of layers, the first five layers operate on a frame-level using time-delay architecture, where layers of size 512 to 1536 are outputted. Then, a statistics pooling layer aggregates the output of the frame-level operational layers, then passes through two hidden layers of dimensions 512 and 300 respectively, which can be used as "speaker embeddings". Finally, the final layer is a softmax classification layer. The final model consists of 4.4 million parameters. [24]

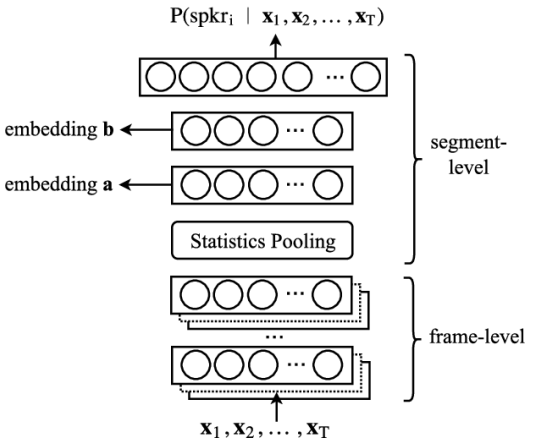

Figure 7: The architecture of the DNN for embedding extraction. The layer extracted comes after the statistics pooling, namely, the embedding b [24].

The input consists of 20-dimensional MFCCs with a frame length of 25ms, which are normalized to up to 3 seconds using a sliding window. A VAD is then used to isolate speech frames from the non-speech ones. Compared to the d-vector input, the frames are not stacked but rather handled by layers inspired by the time-delay neural network (the first five layers). The same PLDA backend used for the i-vectors was utilized here.

The results produced were quite competitive with the i-vector results, with the i-vectors performing better for longer utterances however, the embeddings generated using this architecture produced better results for shorter utterances. [24] It was also noted in the conclusion that the PLDA backend may not be the best method for matching the embeddings generated by a Deep Neural Network.

This method was then later improved upon even further in [31], where the embeddings extracted were officially coined as "x-vectors". One of the biggest differences in the publication [31] was the idea of neural networks being able to handle larger amounts of data more effectively than the earlier baselines. Therefore, an expensive way of introducing noise in the training data was used, by simply augmenting the training data with noise audio files [31]. The architecture used in [31] is detailed below:

Layer	Layer context	Total context	Input x output
frame1	$[t-2, t+2]$	5	120x512
frame2	$\{t-2, t, t+2\}$	9	1536x512
frame3	$\{t-3, t, t+3\}$	15	1536x512
frame4	$\{t\}$	15	512x512
frame5	$\{t\}$	15	512x1500
stats pooling	$[0, T)$	T	1500Tx3000
segment6	$\{0\}$	T	3000x512
segment7	$\{0\}$	T	512x512
softmax	$\{0\}$	T	512xN

Table 1: X-vector architecture, where the embeddings are extracted via *segment 6*, and *N* in the softmax layer represents the number of speakers [31].

The results reported that the x-vectors performed better than the then baseline i-vector, tested on the same data, producing a new industry standard. It should also be noted that the i-vector utilized for comparison in [31] was the best performing version at the time which even utilized transcribed speech for improved performance, while on the other the x-vector did not, and still managed to outperform the baseline method.

3.2.3. Emphasized Channel Attention, Propagation and Aggregation in TDNN

While the x-vector implementation showed promising results [21, 24], several improvements were made to the original implementation. For the VOiCES 2019 Challenge, three “extended” X-vector systems were introduced which outperform the original implementation [28]. Surveying every improved model would be a difficult task due to the time constraints, but one interesting improved implementation will be discussed in this section. [26] introduces a novel approach where it highlights the shortcomings of the x-vector and manages to outperform the state-of-the-art structure.

Figure 8 provides the topology of the newly proposed model:

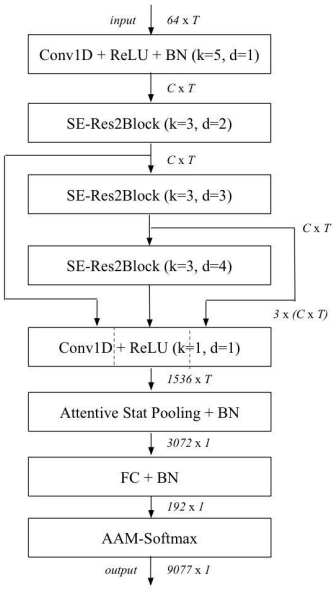

Figure 8: The topology of the proposed ECAPA-TDNN architecture. k and d stand for kernel size and dilation respectively, and C and T correspond to the channel and temporal dimension respectively [26].

The intuition behind this architecture is the improvement on the statistic pooling layer and the introduction of 1-Dimensional Squeeze-Excitation Res2Blocks. [26] Soft self-attention is utilized in recent x-vector architectures for the calculations for the weight statistics in the temporal pooling layer [26]. Self-attention mechanism allows the network to focus on relevant frames by giving them higher weights, which can be interpreted as a Voice Activation Detection (VAD) method discussed in the 3rd chapter of this thesis, Acoustic Analysis. A multi-headed attention technique, which can be described as running several attention mechanisms in parallel allowed for the extraction of a wider variety of speaker characteristics as mentioned in [26]. Due to this, [26] proposed an extension for the temporal attention mechanism to better cater for speaker characteristics that do not activate at the same time such as e.g. ``speaker-specific properties of vowels versus speaker-specific properties of consonants.” [31].

Furthermore, the introduction of 1-Dimensional Squeeze-Excitation Res2Blocks (SE-Res2Block) expands the temporal context, as it has shown to benefit the performance. This method was borrowed from recent advancements made in computer vision, namely, the “Squeeze-and-Excitation” Networks. These new blocks essentially are able to better map the global channel interdependencies. The figure below this specific block:

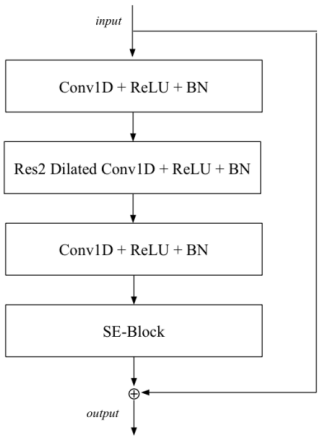

Figure 9: The SE-Res2Block of the ECAPA-TDNN architecture. The standard Conv1D layers have a kernel size of 1. The central Res2Net [16] Conv1D with scale dimension s = 8 expands the temporal context through kernel size k and dilation spacing d [31].

Another concept introduced was the multiplayer feature aggregation. In the x-vector architecture, only complex features strongly correlated with the speaker are used from the feature map in the last frame layer, [26] less complex features can also contribute towards producing robust embeddings. This is done by aggregating all output feature maps from every SE-Res2Block, which are then processed by a dense layer, producing features for the attentive statistical pooling layer. Additionally, residual connections between each of the SE-Res2Blocks is made for each of the previous blocks, where the sum of the output is served as the input as another way to exploit multi-layer information, which provided promising results in other publications [31]. Cosine similarity was used to match the produced speaker embeddings. This architecture was evaluated against other baselines such as the x-vector and r-vector and showed significant improvements.

4. Implementation

This section lists the details of the implementation part of the thesis. The aim for the implementation is to survey available datasets, utilize the survey done for the acoustic analysis, modeling methods, dataset survey in the earlier sections to produce an industry standard speaker identification pipeline, which meets the required task of this thesis.

4.1. Aim

Task for this thesis is to implement a speaker identification system that is catered toward identifying customers on call. Upon receiving a call from an unknown customer, a bot should be able to utilize this pipeline to recognize the unknown customer, given their consent, while prioritizing ease of use, and delivering a quick service for the customer. If the customer's voiceprint is not stored in the database, the bot may prompt the customer to record their voice print, for a better telephony experience in the future, and therefore, the pipeline should be able to produce a voiceprint of the customer and subsequently store it in the database in real-time.

The proposed system is produced using Python Programming Language, and using an open-source PyTorch-powered speech toolkit named, SpeechBrain. In the experimentation part of this thesis, a signal-to-noise ratio concatenation module named, audiolib, found in the Microsoft Scalable Noisy Speech Dataset (MS-SNSD).

4.2. Technical information

All the scripts in the experimentation section are run on the same computer. The configuration are as follows:

Operating System - Ubuntu 22.04 LTS
CPU - AMD Ryzen 5 5800H (8 physical cores, 16 threads)
Graphic Card - Nvidia RTX 3070 mobile
RAM - 16 Gb

Since pre-trained models are utilized, they have been trained on some other system.

4.3. SpeechBrain Toolkit

SpeechBrain is an open-source all-in-one speech toolkit[33] based on PyTorch, mainly focusing on deep learning technology. This toolkit is aimed towards research and development of speech technologies, such as speech recognition, speaker recognition, etc. The main advantage of this toolkit is the ease of use, flexibility and the availability of not only pre-trained models but also recipes to train the models using state-of-the-art architectures. The biggest advantage of this toolkit is the non-restrictive format for utilization. The training scripts are easily automated, and hyperparameters are set using .yaml files. The "Brain" class allows for easy creation of model architectures and data loaders.

For the scope of this thesis, the training recipes and pre-trained models for the X-vector and ECAPA-TDNN will be utilized.

4.4. The Architecture of the speaker identification system.

Judging by the nature of the task, a text-independent speaker identification system is proposed under an open-set configuration.

The system is split into three phases:

1. Training Phase
2. Enrollment Phase
3. Testing Phase

All the three phases go through audio augmentation and feature extraction. While the audio augmentation may differ based on which phase the system is currently at, the feature extraction remains the same for all the three phases. The feature extraction is handled by SpeechBrain classes while augmentations are manually implemented depending on the task at hand.

The training phase produces a speaker-independent model, which is saved in the 'models' directory or if using pre-trained models, is downloaded automatically from HuggingFace database online. For the enrollment phase, the raw audio signal is processed as needed and is sent through the model which then produces an embedding for that speaker, which is stored in the 'database' directory. Finally, for the testing phase, the raw audio signal is processed as needed and passed through the model which then produces an embedding, which is then compared against all the models stored in the 'database', using a similarity technique. The highest similarity score is saved and compared against the pre-set threshold.

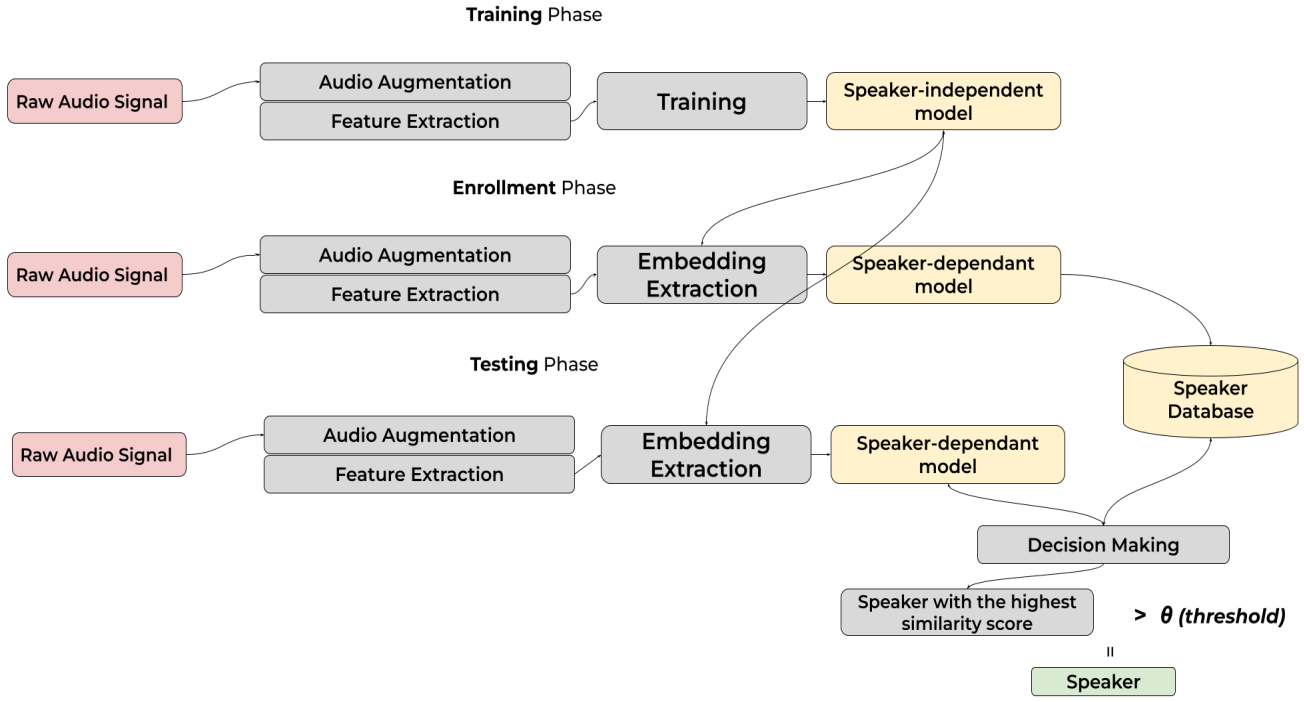

Figure 10: Proposed pipeline for the Speaker Identification system.

We have an open-set configuration which expects to encounter speakers during testing which may not belong in the database. In general, given a set of known speakers $S = \{S_1, S_2, \ldots. S_N\}$ where N refers to the total

number of speakers enrolled in the database, and given some test utterance *u* from an unknown speaker, The output for the open-set configuration using some decision-making technique *f* would be

$$s' = max\, f(u) \quad (20)$$

$$s = s', \quad if\ and\ only\ if\ s' > \theta \quad (21)$$

Where *s'* is the speaker with the highest similarity score, θ is a pre-set threshold described earlier and s is the predicted speaker.

It should be noted that speaker verification is performed in a similar way, the difference being N = 1. A lot of fundamental techniques are shared between the two systems, therefore it can be hypothesized that techniques that perform well for speaker verification, perform similarly well for the task of speaker identification [21].

4.4.1. Implementation Structure

This section details the implementation structure of the proposed speaker identification system. The tree-structure of the project directory is as follows:

- *root*
    - *Database*
    - *Models*
    - *Train*
    - *Data*
        - *Enroll*
        - *Identify*
    - *identification.py*
    - *main.py*

Where 'Database' stores the enrolled embeddings, 'Models' stored the trained or pre-trained models, 'Train' contains the recipes for training specific models, 'Data' consists of two subdirectories, where 'Enroll' is the directory where the audio files for the speakers to be enrolled are stored, and 'Identify' is where the audio file from the unknown speaker will be stored. In addition to the directories above, there are other directories for experimentations where the results and scripts for the experimentations are stored.

In the identification.py module, the class SpeakerIdentificationSystem is present, which takes in five arguments,

- db - which refers to the path where the generated embeddings will be stored [Default is 'Database' in root directory; if not found, will create automatically]
- threshold - for setting a custom threshold value for the identification task.
- enroll_prompt - [Default = True], whether to ask the user for enrollment is not found in the database.
- logs - [Default = True] toggling logs for the system.

In the main.py module, a class instance of the system defined above is initialized and runs on an endless loop with 5 second intervals. This script scans the 'Enroll' directory every interval and enrolls the speakers into the database, (note: the title of the audio files will be stored as their identification) and scans the 'Identify' directory to identify the unknown speaker or prompt to enroll if not in the system.

5. Experimentation

The goal of this section is to produce a model by determining the best performing model from the selected baseline models based on the survey done in the earlier sections, perform a survey of publicly available datasets, select and calculate other factors of the speaker identification system based on the previous surveys, and finally experiment under various conditions and optimize the model if needed.

5.1. Experimentation Setup

Two architectures were selected based on the survey of methods, namely,

1) DNN-based based on X-vector
2) DNN-based based on TDNN-ECAPA

The reason for selecting these specific architectures are as follows:

- They are recent methods that have been experimented with and shown impressive results when compared against other state-of-the-art architectures.
- They had freely available pre-trained models, readily available for usage.
- They were implemented using SpeechBrain that allows for easy experimentation with custom data.

Before the experimentations, it is imperative to declare few elements of the speaker identification system, namely,

1) The technique for decision making.
2) The method for threshold estimation.
3) The method for the evaluation of the models.

5.1.1 Decision Making

Based on the survey from various publications [1, 17, 21, 23, 24, 25, 27, 28], Cosine Distance formulation (CDF) is the two most prominent techniques, especially where speaker embeddings such as the i-vector, and deep speaker embeddings such as x-vector are utilized.

Given two speaker-representation vectors x_{target} and x_{test}, the measure of cosine similarity between them is calculated as

$$score(x_{test}, x_{target}) = \frac{x_{test} \cdot x_{target}}{\|x_{test}\| \times \|x_{target}\|} \quad (22)$$

5.1.2 Evaluation of performance

The basis of the evaluation of a speaker identification system is usually an identification measure, which signifies the system's ability to discriminate between different speakers.

Given a set of known speakers S ={S_1, S_2, ….. S_N} where N refers to the total number of speakers enrolled in the database, and S_C is the number of correctly predicted speakers, the Identification rate *I* can be calculated as:

$$I\ (\%)\ = \frac{S_C}{N} \tag{23}$$

Due to the open-set encountering not enrolled speakers, there also is a need to set a threshold. The metrics used for speaker verification purposes, namely the EER allow the ability to set the ideal threshold.

5.1.3 Threshold Estimation

The threshold estimation is an important factor to consider under the open-set configuration. The optimal threshold value highly depends on the chosen model and the data the model is being tested on. In this thesis, for the selected model, a threshold estimation test will be conducted where the optimal value will be estimated. This will be done by selecting a number of speakers, and enrolling only a part of them into the system. The remaining speakers will be unauthorized speakers. Then, the selected model will be tested using the decision making technique described in the earlier section, where the threshold value will fluctuate between the range of 0 and 100, with 1 increment. To estimate the optimal threshold, two values need to be recorded from the test, namely, False Acceptance Rate (FAR) and False Rejection Rate (FRR).

a. FAR is defined as the number of unauthorized speakers accepted into the system.
b. FRR is defined as the number of authorized speakers rejected by the system.

At a low threshold value, FAR performs better and as the threshold increases, the FRR performs better. The point of their intersection is estimated to be the optimal threshold value for the selected model, also known as the Equal Error Rate (EER).

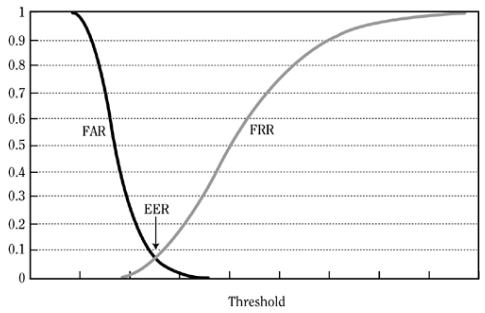

Figure 11: Example of a FAR vs. FRR graph [33].

5.2. Survey of publicly available datasets

In this section, a brief survey of publicly-available datasets is presented in order to make an educated selection for the implementation part of this thesis. It is to be noted that this section does not list every dataset publicly available, but rather the ones that were found either through other publications on Speaker or Speech recognition or by researching on the internet.

Table below summarized the datasets, and also lists information regarding those datasets, namely:

- The name of the dataset
- Language
- Sample rate
- The number of speakers available (if provided)
- The availability status, and whether the dataset is freely available or not
- Details on the dataset

Name	**Language**	**Sample Rate**	**Number of speakers**	**Access**	**Details**
VoxCeleb 1 & 2	Multilingual	16 Khz	7000+	Free	A mix of clean and noisy data; primarily collected from multi-media sources
LibreSpeech	English	16 Khz	2484	Free	Very clean data; Audio Book Recordings
TIMIT*	English	16 Khz	630	Licensed [LDC]	Broadband recordings of various American dialects. The speech was recorded at TI, transcribed at MIT and verified and prepared for CD-ROM production by the National Institute of Standards and Technology (NIST).
NTIMIT	English	16 Khz	630	Licensed [LDC]	Broadband recordings of various American dialects. (TIMIT corpus re-recorded after being transmitted through various channels)
CABank English CallHome Corpus	English	8 Khz	240	Free	Mix of noisy and clean data; Telephone conversation recordings. Each recording contains two speakers having a conversation.
The Voices Obscured in Complex Environmental Settings (VOiCES)	English	16 Khz	300	Free	Noisy data; Recordings in acoustically challenging environments, with a lot of reverberations to simulate real-life scenarios.
VoxForge	Multilingual	8 Khz - 44.1 Khz	2000+	Free	Random online audio contributions, a mix of clean and dirty data, however, the repository really messy (need a lot of arranging and cleaning)
Mozilla Common Voice	Multilingual	8 Khz - 44.1 Khz	-	Free	Random online audio contributions, a mix of clean and dirty data.
Speaker Identification and Verification Archives (SIVA)	Italian	8 Khz	400+	Licensed [ELRA]	Telephony speech dataset based on Italian telephone conversations.
Switchboard-1	English	8 Khz	543	Licensed [LDC]	American two-sided Telephone speech corpus collected by Texas Instruments.

National Institute of Standards and Technology Speaker Recognition Evaluation (NIST SRE) (2000 - 2022)	Multilingual	8 Khz - 16 Khz	-	Licensed [LDC]	Training, Evaluation and Test datasets specially curated by the National Institute of Standards and Technology (NIST) and LDC for the NIST Speaker Recognition Challenges.
RedDots	Multilingual	8 Khz - 44.1 Khz	572	Free	Random speech recordings collected via mobile devices from online submissions, with emphasis on diversity.
English Language Speech Database for Speaker Recognition (ELSDSR)	English	16 Khz	22	Free	Corpus designed by Technical University of Denmark (DTU). Speech recorded under controlled environment to deliver rich audio from a limited number of speakers.
Technology, Entertainment, Design - Laboratory of Informatics of Le Mans University (TED-LIUM)	English	16 Khz	-	Free	Corpus created by collecting public-speaking presentations posted online by TED Talks.

* Title derived due to corpus design being a joint effort among the Massachusetts Institute of Technology (MIT), SRI International (SRI) and Texas Instruments, Inc. (TI)

Table 2: Summaries for the surveyed datasets.

5.2.1. VoxCeleb

A large-scale dataset primarily designed for the task of speaker recognition under noisy and unconstrained conditions. The samples are multilingual and collected from YouTube. In total, there are over a million utterances, a total duration of 2000+ hours, with a split of 39% female to 61% male, consisting of more than 7000 speakers, where each segment is at least 3 seconds long. The dataset is freely available for research purposes, upon sending a form to receive credentials for access [23, 36, 37].

5.2.2. LibreSpeech

A corpus consisting of read English speech, primarily designed training and evaluation for speech recognition. However, unlike most datasets for speech recognition, LibreSpeech contains a high number of speakers (2848), which makes it suitable for speaker recognition as well. The data was derived from the LibriVox project and contains 1000 hours of speech, all sampled at 16 Khz, and noisy segments from the speech data were filtered out [38]. The dataset is freely available for anyone at OpenSLR [39], using the identifier SLR12.

5.2.3. TIMIT

An American read speech corpus consisting of broadband recordings of 630 speakers of eight major dialects of American English, each reading ten phonetically rich sentences. The corpus was mainly designed for the task of automatic speech recognition. The sound samples are provided at 16 Khz, and are single channel. The corpus was designed by the Massachusetts Institute of Technology (MIT), SRI International (SRI) and Texas Instruments, Inc. (TI). Access to the dataset is not freely available and requires a license through the Linguistic Data Consortium (LDC) [40]

5.2.4. NTIMIT

This corpus was created by transmitting the TIMIT dataset through a telephone over varying channels as a telephone bandwidth supplement to the TIMIT dataset. It was developed by NYNEX Science and Technology Speech Communication Group, and contains similar characteristics to the TIMIT dataset, including the access to the dataset. [41]

5.2.5. CABank English CallHome Corpus

This is a telephone speech, collected and transcribed by the LDC, primarily for the use in Large Vocabulary Conversational Speech Recognition (LVCSR), sponsored by the U.S. Department of Defense. This corpus contains 120 unscripted two-way conversations between native American speakers, with 5 to 10 minutes long segments. A total of 240 speakers are present, where each segment consists of 2 speakers. The licensing to the dataset is similar to the previous one.

5.2.6. VOiCES

This creative commons corpus consists of speech segments in acoustically challenging environments with truth data for the purpose of transcription, denoising and speaker identification. Clean dataset from LibreSpeech was augmented with acoustic noise samples, primarily focusing on various reverabial environments. A total of 300 speakers are provided, sampled at 16 Khz. Dataset is freely available for usage via the Registry of Open Data on AWS [42, 43].

5.2.7. VoxForge

This is a database containing a collection of transcribed speech, submitted through online submissions, primarily used for speech recognition tasks. A large variety of speech samples are available for various languages, dialects, sample rate, and containing a mix of both clean and noise speech. All data is very well labeled. The dataset is available publicly for free [44].

5.2.8. Mozilla Common Voice

This corpus consists of submission by online contributors from around the world. There are specific datasets available for various languages, and prior to the release of a dataset, it gets validated. Therefore, it is a fairly reliable source to get quality audio files. There are a lot of 15,234 validated

hours and about 96 languages. All audio samples are provided at 44 Khz sample rate. There is no specific task for which the dataset was created, but rather as an open source source for quality audio samples. There is no distinction between the noisy and clean data. The dataset is available publicly for free [45].

5.2.9. SIVA

A corpus consisting of telephony data in Italian language, primarily created for the task of speaker verification. There are more than 2000 calls included in the dataset. This dataset contains a total of 637 speakers, with a split between normal speakers and imposters. All audio samples are sampled at 8 Khz. This dataset is distributed by the European Language Resources Association (ELRA). [46, 47]

5.2.10. Switchboard-1

A telephony speech corpus consisting of approximately 260 hours of American English speech, collected by Texas Instruments in 1990, designed for speech recognition and speaker identification. There about 543 speakers present in the dataset, and the audio samples are sampled at 8 Khz. This dataset is distributed by the LDC [48].

5.2.11. NIST SRE Datasets

The National Institute of Standards and Technology Speaker Recognition (NIST SRE) is a series of speaker recognition evaluations held by the NIST since 1996. The languages, quality, number of speakers and other technical details for this dataset provided by them varies vastly depending on the plan for the competition of that respective year. This dataset is distributed by the LDC [49].

5.2.12. RedDots

This is an english corpus containing short duration audio samples of variable phonetic content. The recordings are made by having speakers read a large text on their mobile phones, and is expected to contain rich inter-speaker and intra-speaker variabilities. Currently, only 62 speakers are available. As the dataset is primarily aimed towards Speaker Verification, the dataset also contains varying splits of target speakers and imposters. It should be noted that the dataset is still under production and hopes to expand into a large dataset. The dataset is freely available for download for research purposes [50].

5.2.13. ELSDSR

ELSDSR is an English language speech database primarily developed for the task of automatic speaker recognition. It consists of read speech collected by a small group of researchers and students at the Technical University of Denmark (DTU). There are a total of 22 speakers. The aim for this dataset is to produce rich audio samples under controlled conditions by a small group of non-native english speakers. The audio samples are sampled at 16 Khz. The dataset is freely available for download for research purposes [51].

5.2.15. TED-LIUM

This corpus was developed by LIUM for Automatic Speech Recognition (ASR), based on the TED Talks. It is composed of approximately 452 hours of speech, alongside their respective transcripts [52].

5.3. Selection of the background model

5.3.1. Setup

As mentioned in 5.1, two architectures were selected, namely:

- X-vector (please reference section 3.2.2. for description)
- ECAPA-TDNN (please reference section 3.2.3. for description)

As the objective is to produce the best performing model possible, the use of a large augmented dataset is imperative. As training such a model can be extremely resource heavy, this thesis will be utilizing the pre-trained models available via the SpeechBrain toolkit. Both the models were trained as mentioned in the original publication [26] with the only difference being the features extracted are Filterbanks rather than MFCCs.

VoxCeleb2-test dataset was selected for this test.

Dataset	Type	Number of speakers	Total number of files for enrollment	Total number of files for identification
VoxCeleb2-test	Mixed	100	100	1000

Table 3: Distribution of speakers for the test datasets.

5.3.2. Results

It should be noted that both the models were trained on the same corpus, namely, VoxCeleb1+2. Furthermore, the ECAPA-TDNN model was built upon the shortcomings of the X-vector architecture, therefore, it is hypothesized to perform better. However, accuracy is not the metric that will be considered for this test. To make sure the customer experience is as smooth as possible, the technical performance of the model is also important to consider. Therefore, while the ECAPA-TDNN has produced better results in [26] against the X-vector, factors such as identification accuracy, time taken for enrollment and identification and the space occupied by the embeddings produced were not reported. Therefore, the additional metrics described below were also tested:

1) Top-1 percentage (Top-1): The percentage of correctly predicted speakers against total number of speakers. This value was measured for each sample from each speaker, averaged over the number of samples taken, and then averaged again over the total number of speakers present.

2) Top-5 percentage (Top-5): The percentage of currently predicted speakers present in the top five predictions made by the model against the total number of speakers. This value was measured for each

sample from each speaker, averaged over the number of samples taken, and then averaged again over the total number of speakers present.

3) Time taken for enrollment (TTE): The average amount of time it takes the model to enroll a speaker. The value was calculated for each speaker and then averaged over N number of speakers.

4) Time taken for identification (TTI): The average amount of time it takes the model to identify an unknown speaker. The value for identification per test sample was averaged over the total number of samples.

5) Size per embedding (SDS): The size of each embedding produced by each model.

Model	Top-1 (%)	Top-5 (%)	TTE (ms)	TTI (ms)	SDS (kb)
X-vector	63.4	83.2	4.72	41.353	2.795
ECAPA-TDNN	96.9	98.7	14.46	50.389	1.515

Table 4: Metric results tested on mixed data.

5.3.3. Assessment

From the results generated by the test, the biggest differences can be noticed in the Top-1 (%) and TTE values. While it takes the ECAPA-TDNN system approximately 3 times longer to enroll a speaker, the ECAPA-TDNN performs 33% better in terms of accuracy. In addition to that, ECAPA-TDNN performs about 15% times better for the Top-5 (%) and also takes up almost twice as less space per embedding. Another important measure is the time taken for identification. As we can observe, ECAPA-TDNN only takes about 1.2 times more time to identify, therefore, ECAPA-TDNN is selected to be the better model.

This decision was made due to the fact that in the case of enrollment, the process will occur only once per customer, and will take place once they give their consent to have their voice print be stored in the database. Furthermore, the process takes place after the customer has entered the call and in the background, thereby not hindering their experience.

5.3.4. Threshold estimation

As mentioned in section 5.1.3, threshold estimation test was run for the ECAPA-TDNN model.
The same dataset used in 5.3.1 was used for this test as well, with the difference being only half the number of speakers were enrolled.

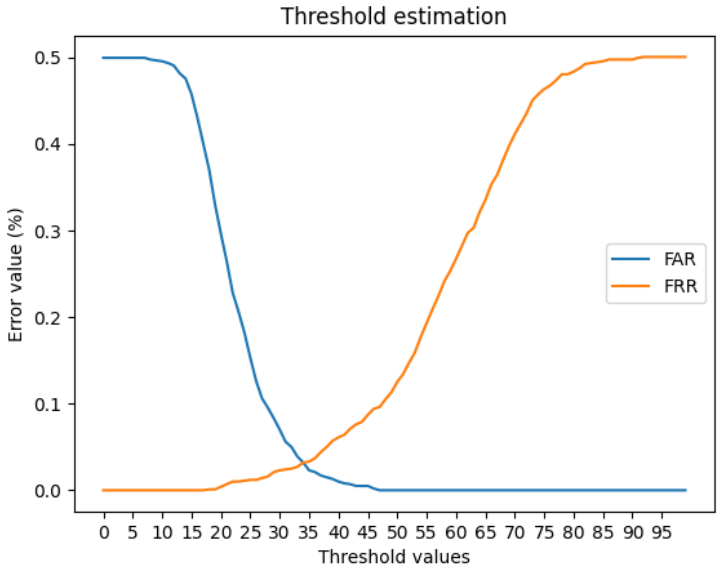

Figure 12: Threshold estimation results.

Based on the results observed, the threshold estimate is 0.34.

5.4. Optimization Task

The selected model from the previous section was further experimented on in this section. The aim of this section is to identify various factors affecting the identification task during a call received, evaluate the model and optimize and produce the best performing model possible within the scope of the author of this thesis. Knowing the nature of the task, there are certain factors that affect the overall quality of the identification system. The factors experimented with are:

- Performance against variation in the length of the test utterances.
- Performance against noisy data (non-intelligible).
- Performance against noisy data (intelligible).
- Performance against multiple languages.
- Performance against artificially generated voice samples.

It should be noted that all the experiments are conducted under the “closed set” configuration. While the goal of this thesis is to produce an Speaker Identification system under the open set configuration, the addition of a threshold for these experimentation scenarios impedes the inference to be made from their results. Additionally, the estimated threshold in the earlier section, while fairly robust, can be subjected to weak performance and possible false rejections due to various real-life factors such as large amounts of noise present in a signal.

5.4.1. Datasets utilized

LibreSpeech dataset is utilized for clean speech utterances. Specifically, hundred speakers were taken from the train-clean dataset [38], where each speaker contains at least fifteen minutes worth of speech.

Noise samples are gathered from Freesound [53] and Demand [54] for noise concatenation experiments. All audio samples are present in .wav format, sampled at 16 Khz and are single channel.
For the multilingual experiment, datasets were gathered from Mozilla Common Voice Dataset [45]. All audio samples are present in .mp3 format, sampled at 16 Khz (downsampled from the original 32 Khz) and are single channel. Finally, for the mimicry experiment, audio samples were provided from MAMA AI.

5.4.2. Performance against variation in the length of the utterances.

5.4.2.1. Setup

The objective of this experiment is to determine the model's ability to identify speakers given varying lengths of utterances as test data. In a real-life scenario, it is expected from the system to be able to identify the unknown speaker ideally within the first few seconds. For this experiment, clean speech audio files from LibreSpeech were taken, and merged together. The merged file was then systematically split into varying lengths ranging from 3 to 10 seconds, and at least 10 samples were generated for each length.

Task	Number of speakers	Number of files
Enrollment	100	100
Identification	100	8000

Table 5: File distribution for experiment.

5.4.2.2. Results

Both the identification rate (IR) and top three confidence values are generated for each utterance length. For each utterance length, the IR was averaged over ten samples per speaker, and then averaged by the hundred speakers. Additionally, to hypothesize the model's ability to distinguish between each speaker model, the top three confidence values that are produced for every test sample are also recorded. Similar to the IR value, these values are averaged over the number of samples per speaker, and then over the total number of speakers for each utterance length.

Length of utterance (s)	IR (%)	Top three confidence values (%)		
3	99.3	0.676	0.352	0.311
4	99.3	0.714	0.364	0.320
5	99.4	0.738	0.372	0.329
6	99.4	0.756	0.375	0.331
7	99.5	0.766	0.377	0.344
8	99.8	0.781	0.384	0.339

9	99.8	0.791	0.386	0.341
10	99.8	0.799	0.390	0.344

Table 6:Results of the experiment.

5.4.2.3. Assessment

The results produced look very promising. Performance was expected to drop given a short utterance length, however, the identification rate is on par with the longer ones. The most noticeable drop can be observed for the best matched confidence value for the 3 and 10 second utterances, 0.676 compared to 0.799.

5.4.3. Performance against noisy data (non-intelligible)

5.4.3.1. Setup

The objective of this experiment is to determine the model's ability to handle noisy data, specifically, non-intelligible noise where human speech is not present. Noise in a signal is one of the most popular problems in speech analysis, and considering the task of the proposed Speaker Identification System is concerned with customer service, noisy signals are to be expected. While it is difficult to measure the level of background noise present in a signal, however, a signal-to-noise ratio (SNR) is commonly used as an estimate [55]. It is defined as ratio between the intensity of the signal and the intensity of the noise present in a signal, expressed using decibels (dB):

$$SNR_{dB} = 10log(S_E / N_E) \text{ [55]} \quad (24)$$

Where S_E and N_E are the energies of the signal and noise signal, respectively, which can be calculated by

$$E = sum(s[n]^2) \text{ [55]} \quad (25)$$

If, for instance, a clean speech signal at 80 dB and some background noise signal at 50 dB are concatenated together, the SNR value of the resultant signal would be 30 dB, or in other words, the clean speech is 30 dB "louder" than the noise in the signal.

To simulate the noisy signals, this experiment takes the clean speech used in the previous experiment, and concatenates noise to each sample at varying SNR ratios. This is done using the audiolib module provided by the Microsoft Scalable Noisy Speech Dataset (MS-SNSD). A range from -5 dB till 20 dB was considered. Furthermore, four non-intelligible noise profiles were selected, namely, sound from a vacuum cleaner (cleaner), air conditioner (aircon), various noises in a park (park) and noises from passing traffic (traffic). The noise samples were collected from Freesound and Demand [53],[54] (only files with Creative Commons license were selected).

While it is imperative to expect noise from test samples, it is also important to consider the case where the audio samples used for enrollment may also contain noise. Therefore, two cases were tested,

1) Clean speech for enrollment, Noisy Speech for testing.
2) Noisy speech for enrollment, Noisy Speech for testing.

1) Clean speech for enrollment, Noisy speech for testing.

Similar to the previous experiment, 100 speakers were selected, where a clean, 10 second-long sample was obtained for each speaker for enrollment, and 10 samples each from 3 second and 10 second utterances for each speaker were utilized. Each test sample was also then concatenated with the four noise profiles, at different SNR values ranging from -5 till 20 dB.

2) Noisy speech for enrollment, Noisy speech for testing.

Similar to the previous experiment, 100 speakers were selected, where a clean, 10 second-long sample was obtained for each speaker for enrollment, and was concatenated with a randomly chosen noise profile and SNR value. For testing, the same distribution as the previous case was used.

Task	Number of speakers	Number of samples per speaker	Number of lengths for utterances	Number of noise profiles	Number of SNR values	Total number of files
Enrollment	100	1	1 (10 seconds)	None, 1 (noisy; randomly selected)	None, 1 (noisy; randomly selected)	100
Identification	100	20	2 (3 and 10 seconds)	4	6	48000

Table 7: File distribution for the experiment.

5.4.3.2. Results

Similar to the previous experiment, both IR and confidence values were generated, the difference being only the top one confidence values are averaged over all the samples, due to the large number of samples present.

1) Clean speech for enrollment, Noisy speech for testing.

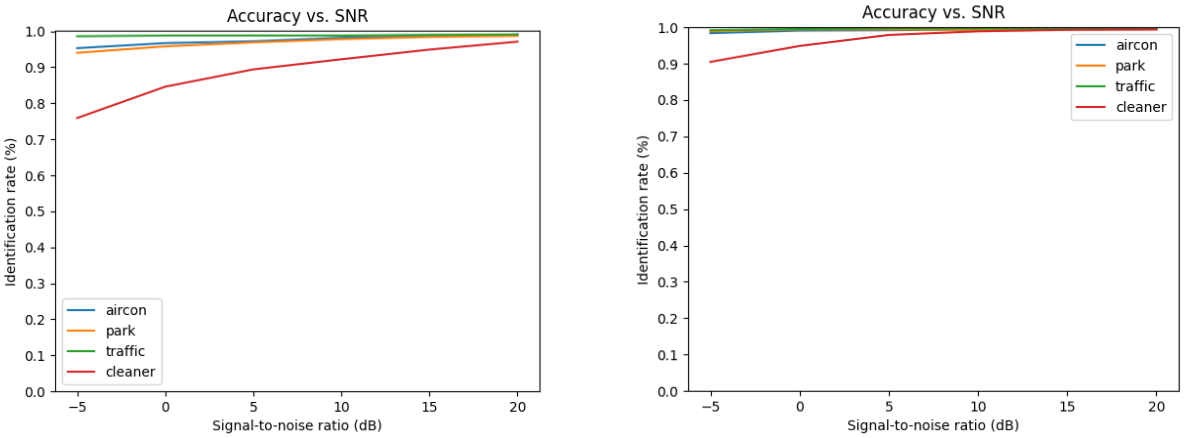

Figure 13: IR (%) values for 3 seconds (left) and 10 seconds (right) length utterances. (clean speech enrolled, noisy speech tested)

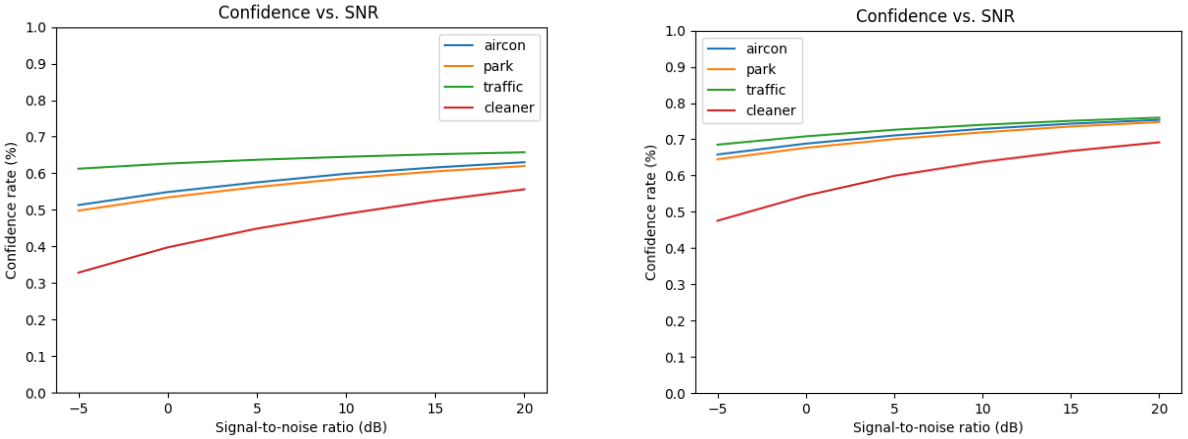

Figure 14: Top Confidence (%) values for 3 seconds (left) and 10 seconds (right) length utterances. (clean speech enrolled, noisy speech tested)

2) Noisy speech for enrollment, Noisy speech for testing.

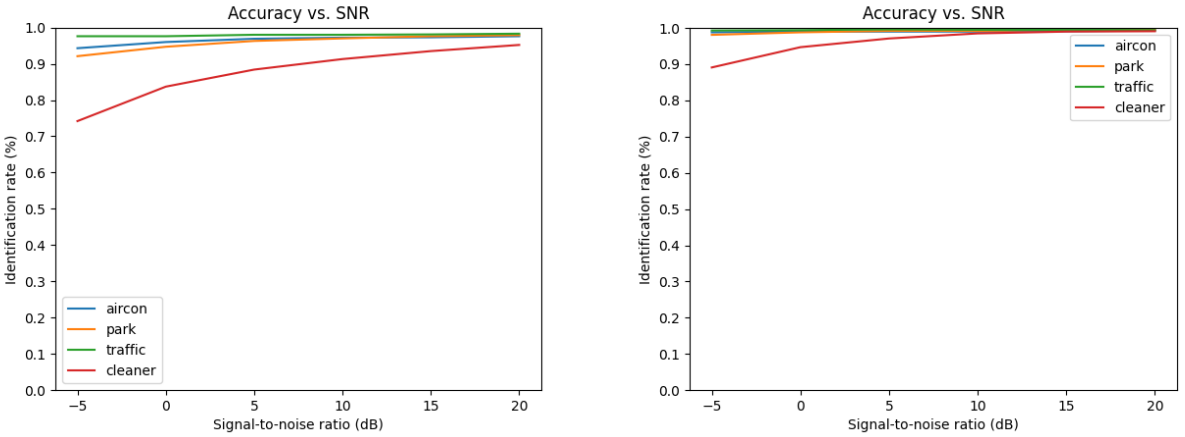

Figure 15: IR (%) values for 3 seconds (left) and 10 seconds (right) length utterances. (noisy speech enrolled, noisy speech tested)

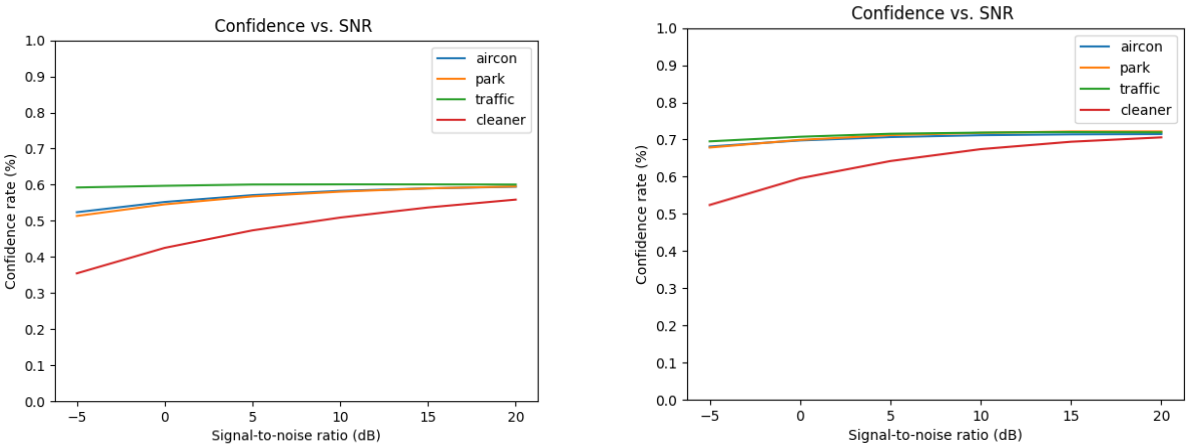

Figure 16: Top Confidence (%) values for 3 seconds (left) and 10 seconds (right) length utterances. (noisy speech enrolled, noisy speech tested)

5.4.3.3. Assessment

For the case where clean speech was enrolled, results are very similar for both 3 and 10 second utterances. The lowest IR was given by the vacuum cleaner (cleaner) noise profile. This may be due to the fact that unlike the other noise profiles, the noise in the sample was constantly present throughout. The performance of the model was good, with the lowest IR value being 75.9% for 3 second utterance for the cleaner noise profile, while the lowest confidence value being 32.8% for the same. As this value is lower than the estimated threshold in the previous section, the threshold will be lowered to 32%.
For the case where noisy speech was enrolled, results are slightly lower than the clean speech enrollment, however, the confidence values are higher than the previous case. This may be due to the fact that some of the random noise profiles that were concatenated to enrollment samples were able to match the noise profiles in the test samples at similar SNR levels. Additionally, while the confidence values may have risen for certain samples, the overall accuracy has dropped by a small percentage. Nevertheless, the performance of the model still remains good, with the lowest IR value being 74.2% for again, the 3 second utterance for the cleaner noise profile.

5.4.4. Performance against noisy data (intelligible)

5.4.4.1 Setup

The objective of this experiment is to determine the model's ability to handle noisy data, specifically, intelligible noise where human speech is present.
To simulate the noisy signals, this experiment concatenates noise unto clean audio as done in the previous experiment, however, four intelligible noise profiles were selected, namely, background noise from a cafe (cafe), background noise from a nearby crowd (crowd), noise from neighbors (neighbors) and airport announcements at an airport (airport). The noise samples were collected from Freesound and Demand [53],[54] (only files with Creative Commons license were selected). Additionally, the same two cases for speech enrollment were also tested.

1) Clean speech for enrollment, Noisy speech for testing.

Similar to the previous experiment, 100 speakers were selected, where a clean, 10 second-long sample was obtained for each speaker for enrollment, and 10 samples each from 3 second and 10 second utterances for each speaker were utilized. Each test sample was also then concatenated with the four noise profiles, at different SNR values ranging from -5 till 20 dB.

2) Noisy speech for enrollment, Noisy speech for testing.

Similar to the previous experiment, 100 speakers were selected, where a clean, 10 second-long sample was obtained for each speaker for enrollment, and was concatenated with a randomly chosen noise profile and SNR value. For testing, the same distribution as the previous case was used.

Task	Number of speakers	Number of samples per speaker	Number of lengths for utterances	Number of noise profiles	Number of SNR values	Total number of files
Enrollment	100	1	1 (10 seconds)	None, 1 (noisy; randomly selected)	None, 1 (noisy; randomly selected)	100
Identification	100	20	2 (3 and 10 seconds)	4	6	48000

Table 8: File distribution for the experiment 3

5.4.4.2. Results

Similar to the previous experiment, both IR and top confidence values were generated.

1) Clean speech for enrollment, Noisy speech for testing.

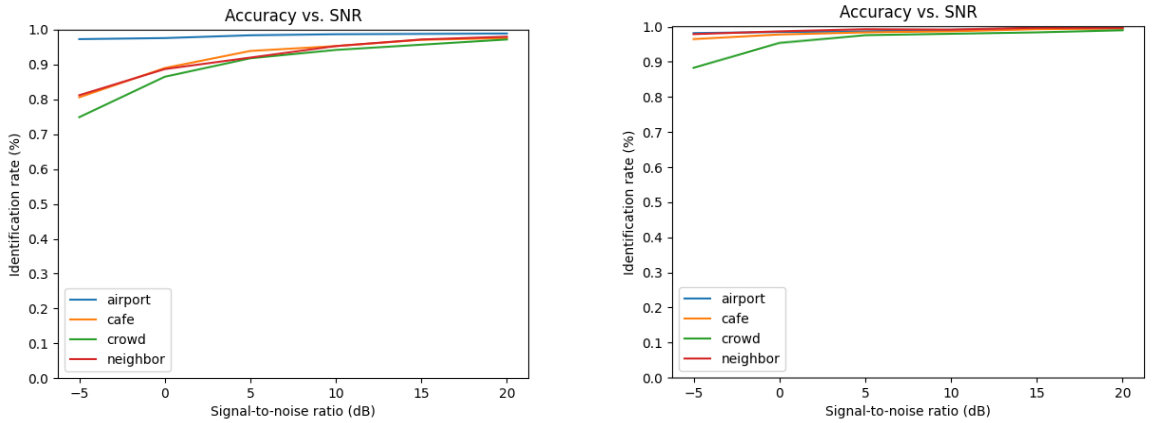

Figure 17: IR (%) values for 3 seconds (left) and 10 seconds (right) length utterances. (clean speech enrolled, noisy speech tested)

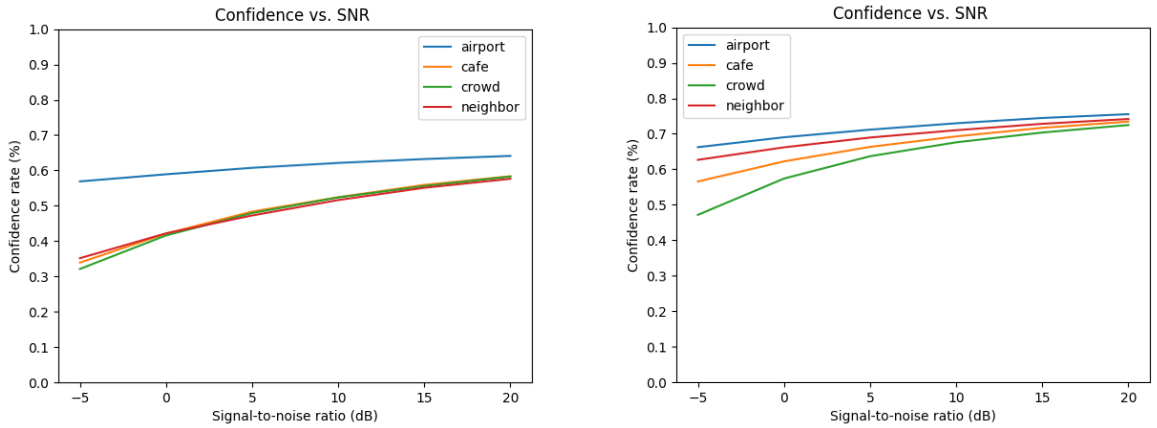

Figure 18: Top Confidence (%) values for 3 seconds (left) and 10 seconds (right) length utterances. (clean speech enrolled, noisy speech tested)

2) Noisy speech for enrollment, Noisy speech for testing.

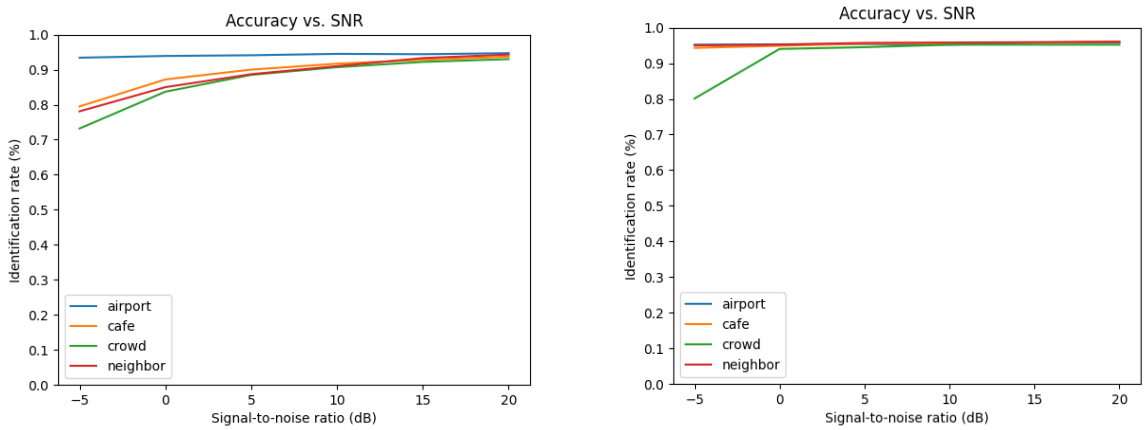

Figure 19: IR (%) values for 3 seconds (left) and 10 seconds (right) length utterances. (noisy speech enrolled, noisy speech tested)

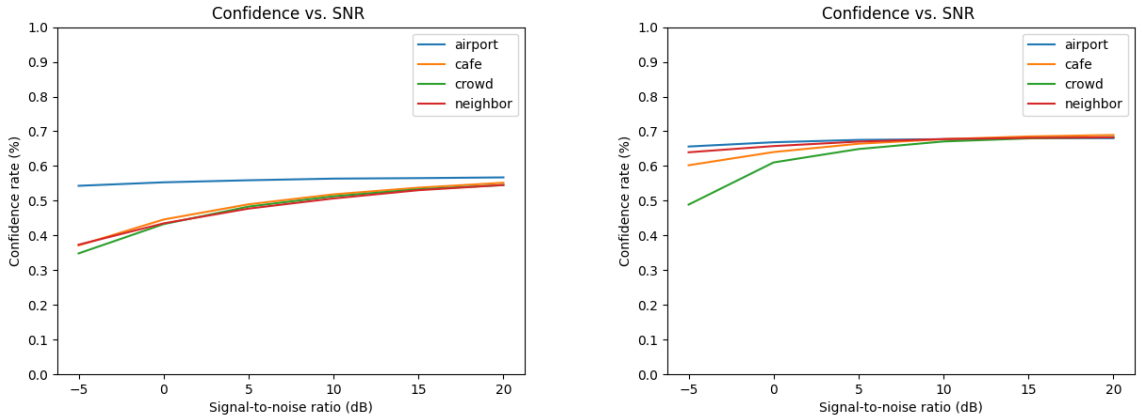

Figure 20: Top Confidence (%) values for 3 seconds (left) and 10 seconds (right) length utterances. (noisy speech enrolled, noisy speech tested)

5.4.4.3. Assessment

For the case where clean speech was enrolled, results are very similar for both 3 and 10 second utterances. The lowest IR was given by the crowd noises (crowd) noise profile.This was expected, considering crowd noises or babble is the most challenging type of background noise in speech technology, due to the fact it overlaps the clean signal. Regardless, the performance of the model was still good, with the lowest IR value being 74.9% for 3 second utterance for the crowd noise profile, while the lowest confidence value being 32.1% for the same.

For the case where noisy speech was enrolled, results are slightly lower than the clean speech enrollment, however, the confidence values are higher than the previous case. This may be due to the fact that some of the random noise profiles that were concatenated to enrollment samples were able to match the noise profiles in the test samples at similar SNR levels. Additionally, while the confidence values may have risen for certain samples, the overall accuracy has dropped by a small percentage. Nevertheless, the performance of the model still remains good, with the lowest IR value being 73.2% for again, the 3 second utterance for the crowd noise profile.

5.4.5. Performance against multiple languages.

5.4.5.1. Setup

In this experiment, the multilingual capabilities of the model were assessed. Given that the training data used for the model is a multilingual dataset (VoxCeleb), the model is expected to perform fairly well. The data for this test was gathered from Mozilla Common Voice Dataset [45]. Four varying languages were selected, namely, Czech, Hindi, Japanese and Chinese.
Since a large corpus with a high number of speakers from the same dataset can be difficult to source, only 10 speakers per language were selected. For every language dataset from Common Voice, metadata is provided, which details information for every audio sample, including speaker_id, transcription, age, gender etc.
The test.tsv file selected for every language, and 10 speakers and about 200 seconds worth of their speech was extracted. Since the audio samples are provided at a sample rate of 44 Khz, all audio samples were downsampled to 16 Khz.

Task	Number of speakers	Number of samples per speaker	Number of lengths for utterances	Number of languages	Total number of files
Enrollment	10	1	1 (10 seconds)	4	40
Identification	10	20	2 (3 and 10 seconds)	4	800

Table 9: File distribution for the experiment 4

5.4.5.2. Results

Similar to the previous experiments, IR (%) and the top CR (%) was generated.

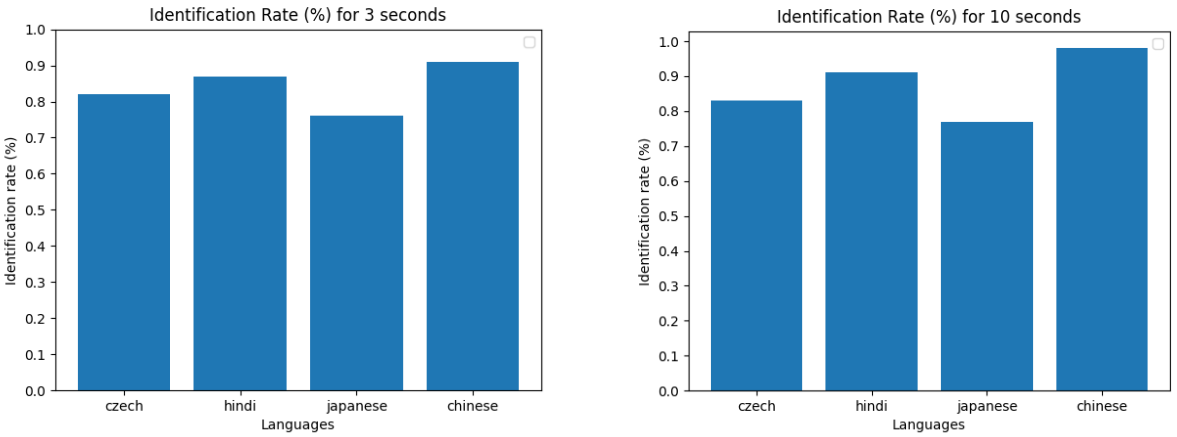

Figure 21: IR (%) values for 3 seconds (left) and 10 seconds (right) length utterances.

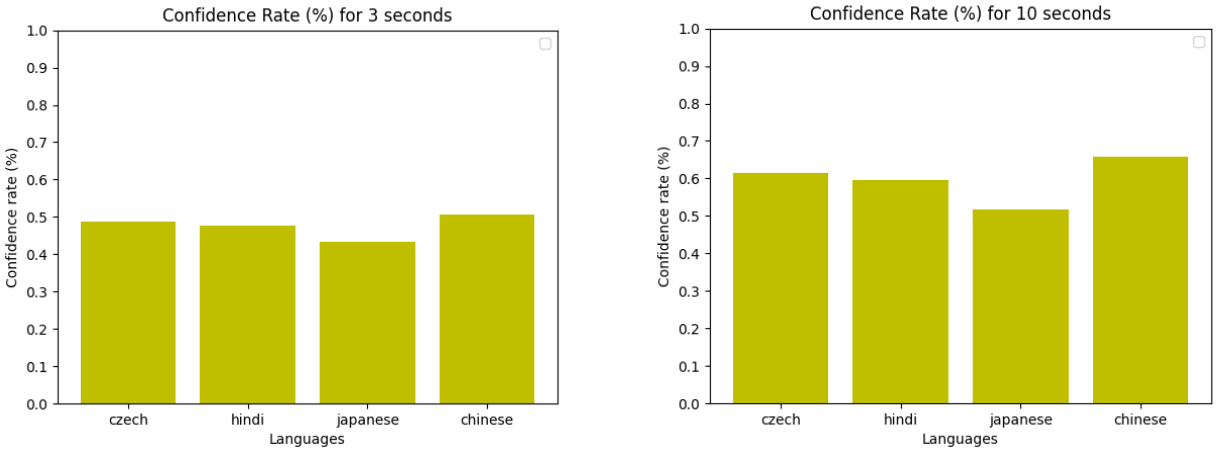

Figure 22: CR (%) values for 3 seconds (left) and 10 seconds (right) length utterances.

5.4.5.3. Assessment

The results seem fairly promising. The highest IRs was Chinese for both the 3 and 10 second utterances, while Japanese was the lowest. The lowest IRs was Japanese for 3 second utterances at 75.9%, while highest being Chinese for the 10 second utterances at 98%. Furthermore, None of the CRs dropped below the threshold 0.32.

5.4.5. Performance against artificially generated voice samples.

5.4.5.1. Setup

In this experiment, the model's capability to distinguish between voice samples from a real speaker and an artificial generated one. Due to deep neural voice technology being out-of-scope for this thesis, data for this experiment was obtained from MAMA AI. A combination of FastSpeech Text-to-Speech model and vocoder based on Univnet Generative adversarial neural network, was utilized to generate samples based on both professional and non-professional speakers. However, due to time constraints and the fact that constructing a speech synthesizer for a speaker is resource heavy, audio samples from 5 models were generated.

In total, 4 speakers are present, Jana, Tomas, Karolina and Lenka. Similar to the previous experiments, 10 seconds long audio samples from the real speaker were enrolled, and 10 samples for 3 second and 10 second long samples from each of the speakers were generated by the deep neural voice model.

Task	Number of speakers	Number of samples per speaker	Number of lengths for utterances	Total number of files
Enrollment	4	1	1 (10 seconds)	100
Identification	4	20	2 (3 and 10 seconds)	80

Table 10: File distribution for the experiment 5

5.4.5.2. Results

The audio samples provided for enrollment have a sample rate of 48 Khz, while the samples for test files (artificially generated) have a sample rate of 16 Khz. Therefore, three distinct tests were performed, where the samples for enrollment were enrolled at 48 Khz, 22 kHz and 16 kHz.

Similar to the previous experiments, both IR and CR values are calculated.

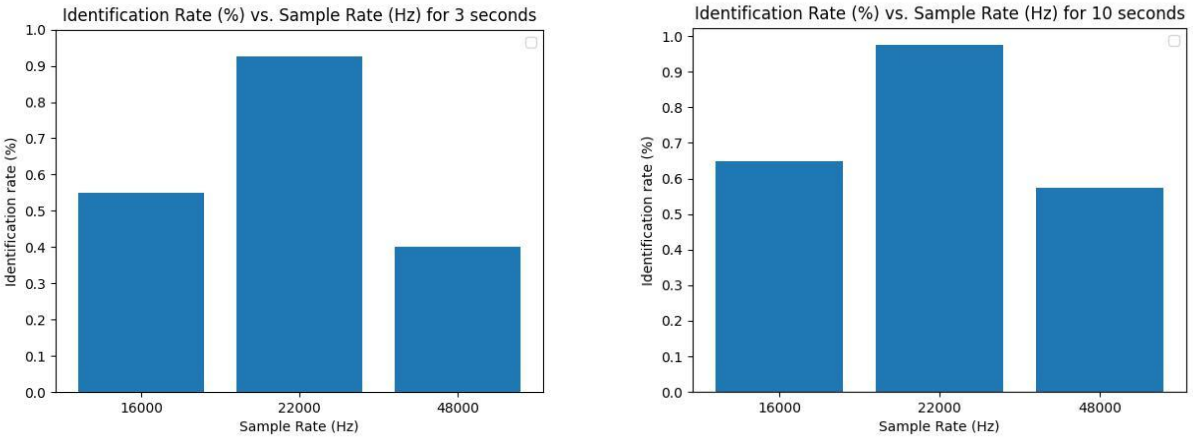

Figure 23: IR (%) values for 3 seconds (left) and 10 seconds (right) length utterances.

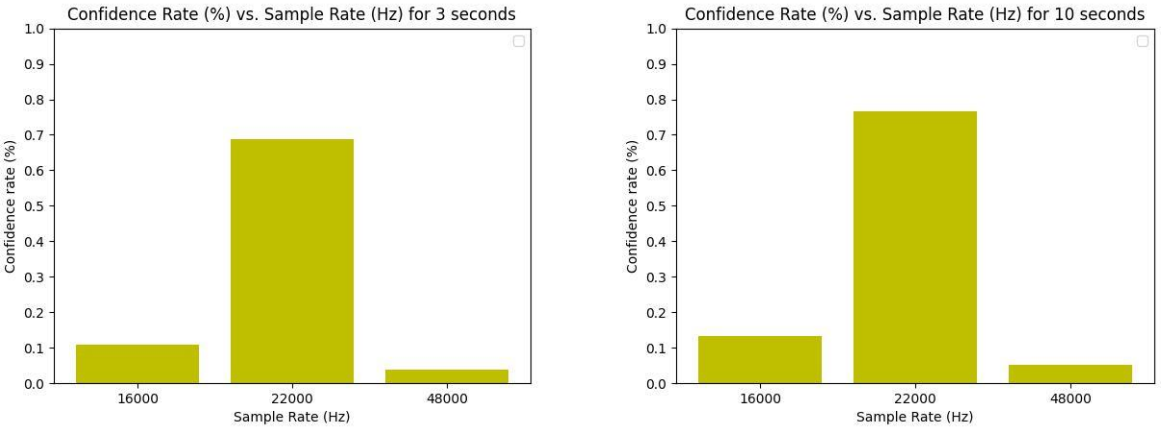

Figure 24: CR (%) values for 3 seconds (left) and 10 seconds (right) length utterances.

5.4.5.3. Assessment

Due to the setup of this experiment, lower accuracy values are preferred (artificially generated utterances are not able to fool the system). As it can be observed, the best performance is observed when the enrollment samples are present in their original sample, and when they are downsampled to 16 Khz. While the accuracy values may seem high, one should take a note of the confidence values. The highest confidence value calculated for 16 and 48 Khz, is 13.2 % (or 0.13) which is much lower than our previously set threshold for the system, 0.13. However, the main oddity is when the samples are downsampled to 22 Khz. The highest IR rate is calculated for the 10 second utterance at 97.5 %, with the highest confidence rate being 76.6 % (or 0.76). This will completely bypass the threshold set previously, and be able to fool the speaker identification system. This may be due to the fact that for the production of the speaker model, the training files have a

sample rate of 22 Khz. Nevertheless, this experiment was catered more towards future research since the topic is out-of-scope of the thesis.

6. Conclusion

The work done in this thesis deals with methods for speaker identification from an acoustic signal. The theoretical part highlights the extensive research to educate the author of the process of speaker identification, where emphasis was given towards acoustic analysis, which consisted of the extraction process of the popular feature that represents human speech, MFCCs and also details on why Filterbanks might be a better choice for especially neural networks. Another highlight is dedicated towards the architecture of the past and current baseline methods, namely, GMM-UBM with i-vector and X-vector, ECAPA-TDNN respectively.

The practical part consisted of the implementation of a speaker identification pipeline based on a particular task described by MAMA AI, and a survey on publicly available datasets. The speaker identification pipeline was implemented using an open-source speech toolkit called SpeechBrain, and other auditory python libraries. Two pre-trained models were selected which were based on the current baseline architecture detailed in the theoretical part, and were tested against each other using various metrics. Both the pre-trained models were trained using the same features, training dataset, augmentations and training parameters. The ECAPA-TDNN model performed approximately 33% better in terms of Top-1 identification rate, 15% better in terms of Top-5 identification rate, taking up almost twice as less space to store the embedding than the competitor, with the only drawbacks being the time taken to enroll a speaker was 3 times slower and time taken to identify being 1.2 times slower than the competition, X-vector. However, the process of enrollment is of less importance than the other metrics, and as time taken for identification is not much different, it was selected as the better performing model. The optimal threshold was then estimated by performing a test where 50% of the speakers were imposters. The intersection value for the False Rejection and False Acceptance rate was found to be 0.34, which was set as the estimated “optimal threshold”.

The selected model was then experimented on under various scenarios, namely, performance against varying of utterances, noise concatenation (intelligible and unintelligible), various languages. The model portrayed excellent performance against all the experiments, with the lowest IR (%) being 73.2, and lowest confidence value being 32.1%. A final experimentation was performed which was concerned with the ability of the model against artificially generated voice samples using Text-to-speech synthesizers. This experiment was out of scope for this thesis and is meant as a reference for future research. This was the only experiment where the model performed poorly, as the artificially generated voice samples were able to fool the system for one of the three cases tested.

In conclusion, the model portrayed satisfactory results given all the challenges, and was selected to be part of the final speaker identification pipeline produced in the implementation part of this thesis. The estimated value of the threshold was lowered from 0.34 to 0.32 to accommodate noisy signals with crowds.

References:

[1] R. Togneri and D. Pullella, "An Overview of Speaker Identification: Accuracy and Robustness Issues," in *IEEE Circuits and Systems Magazine*, vol. 11, no. 2, pp. 23-61, Second Quarter 2011, doi: 10.1109/MCAS.2011.941079.

[2] Kekre, Hemant & Kulkarni, Vaishali. (2013). Closed set and open set Speaker Identification using amplitude distribution of different Transforms. 2013 International Conference on Advances in Technology and Engineering, ICATE 2013. 1-8. 10.1109/ICATE.2013.6524764.

[3] Nilu Singh, R.A. Khan, Raj Shree, (2012). Applications of Speaker Recognition, Procedia Engineering, Volume 38, 2012, Pages 3122-3126, ISSN 1877-7058, https://doi.org/10.1016/j.proeng.2012.06.363.

[4] kavya gupta 0098, (2021), How is Sound Produced by Humans?, available at: https://www.geeksforgeeks.org/how-is-sound-produced-by-humans/, (Accessed : 10th, December, 2021)

[5] J. P. Campbell, "Speaker recognition: a tutorial," in Proceedings of the IEEE, vol. 85, no. 9, pp. 1437-1462, Sept. 1997, doi: 10.1109/5.628714.

[6] K V Vijay Girish, (2019) Beginner's guide to Speech Analysis, available at: https://towardsdatascience.com/beginners-guide-to-speech-analysis-4690ca7a7c05 (Accessed: 15th December, 2021)

[7] P. K. P.S., G. Thimmaraja Yadava and H. S. Jayanna, "Text Independent Speaker Identification: A Review," 2017 2nd International Conference On Emerging Computation and Information Technologies (ICECIT), 2017, pp. 1-6, doi: 10.1109/ICECIT.2017.8453360.

[8] Z. Wanli and L. Guoxin, "The research of feature extraction based on MFCC for speaker recognition," *Proceedings of 2013 3rd International Conference on Computer Science and Network Technology*, 2013, pp. 1074-1077, doi: 10.1109/ICCSNT.2013.6967289.

[9] Rao, K.S. and Vuppala, A.K. (2014). Speech Processing in Mobile Environments. SpringerBriefs in Electrical and Computer Engineering. Cham: Springer International Publishing. doi:10.1007/978-3-319-03116-3.

[10] Sonkamble, B. and Doye, D. (2012). Speech Recognition Using Vector Quantization through Modified K-meansLBG Algorithm. [online]. Available at: https://core.ac.uk/download/pdf/234644507.pdf [Accessed 15 Aug. 2022].

[11] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," in IEEE Transactions on Speech and Audio Processing, vol. 3, no. 1, pp. 72-83, Jan. 1995, doi: 10.1109/89.365379.

[12] Carrasco, O.C. (2020). Gaussian Mixture Models Explained. [online] Medium. Available at: https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95.

[13] T. R. J. Kumari and H. S. Jayanna, "Comparison of LPCC and MFCC features and GMM and GMM-UBM modeling for limited data speaker verification," 2014 IEEE International Conference on Computational Intelligence and Computing Research, 2014, pp. 1-6, doi: 10.1109/ICCIC.2014.7238329.

[14] F. R. Chowdhury, S. -A. Selouani and D. O'Shaughnessy, "Distributed automatic text-independent speaker identification using GMM-UBM speaker models," 2009 Canadian Conference on Electrical and Computer Engineering, 2009, pp. 372-375, doi: 10.1109/CCECE.2009.5090157.

[15] Reynolds, D.A., Quatieri, T.F. and Dunn, R.B. (2000). Speaker Verification Using Adapted Gaussian Mixture Models. Digital Signal Processing, 10(1-3), pp.19–41. doi:10.1006/dspr.1999.0361.

[16] V. X. Nguyen, V. P. H. Nguyen and T. V. Pham, "Robust speaker identification based on hybrid model of VQ and GMM-UBM," *2015 International Conference on Advanced Technologies for Communications (ATC)*, 2015, pp. 490-495, doi: 10.1109/ATC.2015.7388377.

[17] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel and P. Ouellet, "Front-End Factor Analysis for Speaker Verification," in IEEE Transactions on Audio, Speech, and Language Processing, vol. 19, no. 4, pp. 788-798, May 2011, doi: 10.1109/TASL.2010.2064307.

[18] W. M. Campbell, D. E. Sturim, D. A. Reynolds and A. Solomonoff, "SVM Based Speaker Verification using a GMM Supervector Kernel and NAP Variability Compensation," 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, 2006, pp. I-I, doi: 10.1109/ICASSP.2006.1659966.

[19] S. Sarkar and K. S. Rao, "Speaker verification in noisy environment using GMM supervectors," 2013 National Conference on Communications (NCC), 2013, pp. 1-5, doi: 10.1109/NCC.2013.6487995.

[20] Lei, H. (n.d.). Joint Factor Analysis (JFA) and i-vector Tutorial. [online] Available at: https://www1.icsi.berkeley.edu/Speech/presentations/AFRL_ICSI_visit2_JFA_tutorial_icsitalk.pdf [Accessed 15 Aug. 2022].

[21] Bai, Z. and Zhang, X.-L. (2021). Speaker recognition based on deep learning: An overview. Neural Networks, 140, pp.65–99. doi:10.1016/j.neunet.2021.03.004.

[22] M. McLaren, L. Ferrer, D. Castan, A. Lawson, The speakers in the wild (SITW) speaker recognition database., in: Interspeech, 2016, pp. 818– 822.

[23] A. Nagrani, J. S. Chung, A. Zisserman, VoxCeleb: A large-scale speaker identification dataset, Proc. Interspeech 2017 (2017) 2616–2620.

[24] Snyder, D., Garcia-Romero, D., Povey, D., Khudanpur, S. (2017) Deep Neural Network Embeddings for Text-Independent Speaker Verification. Proc. Interspeech 2017, 999-1003, doi: 10.21437/Interspeech.2017-620

[25] E. Variani, X. Lei, E. McDermott, I. L. Moreno and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014, pp. 4052-4056, doi: 10.1109/ICASSP.2014.6854363.

[26] ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification, https://arxiv.org/abs/2005.07143

[27] P. Matějka et al., "Analysis of DNN approaches to speaker identification," 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016, pp. 5100-5104, doi: 10.1109/ICASSP.2016.7472649.

[28] Snyder, David & Villalba, Jesús & Chen, Nanxin & Povey, Daniel & Sell, Gregory & Dehak, Najim & Khudanpur, Sanjeev. (2019). The JHU Speaker Recognition System for the VOiCES 2019 Challenge. 2468-2472. 10.21437/Interspeech.2019-2979.

[29] Borgström, Bengt J.. “Discriminative Training of PLDA for Speaker Verification with X-vectors.” (2020).

[30] Doddington, G.R., Przybocki, M.A., Martin, A.F. and Reynolds, D.A. (2000). The NIST speaker recognition evaluation – Overview, methodology, systems, results, perspective. Speech Communication, 31(2-3), pp.225–254. doi:10.1016/s0167-6393(99)00080-1.

[31] Haytham Fayek (2016). Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What’s In-Between. [online] Haytham Fayek. Available at: https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html.

[32] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, “X-vectors: Robust DNN embeddings for speaker recognition,” in Proc. ICASSP, 2018, pp. 5329–5333.

[33] flylib.com. (n.d.). EER | Biometrics for Network Security (Prentice Hall Series in Computer Networking and Distributed). [online] Available at: https://flylib.com/books/en/4.400.1.71/1/ [Accessed 15 Aug. 2022].

[34] speechbrain.github.io. (n.d.). About SpeechBrain. [online] Available at: https://speechbrain.github.io/about.html [Accessed 13 Aug. 2022].

[35] Mel-filter banks. (n.d.). Available at: http://siggigue.github.io/pyfilterbank/melbank.html.

[36] Nagrani, A., Chung, J. S., Xie, W., & Zisserman, A. (2020). Voxceleb: Large-scale speaker verification in the wild. Computer Speech & Language, 60, 101027. https://doi.org/https://doi.org/10.1016/j.csl.2019.101027

[37] Chung, J. S., Nagrani, A., & Zisserman, A. (2018, September). VoxCeleb2: Deep Speaker Recognition. Interspeech 2018. https://doi.org/10.21437/interspeech.2018-1929

[38] 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), 2015, pp. 5206-5210, doi: 10.1109/ICASSP.2015.7178964.

[39] www.openslr.org. (n.d.). openslr.org. [online] Available at: https://www.openslr.org/ [Accessed 15 Aug. 2022].

[40] catalog.ldc.upenn.edu. (n.d.). TIMIT Acoustic-Phonetic Continuous Speech Corpus - Linguistic Data Consortium. [online] Available at: https://catalog.ldc.upenn.edu/LDC93S1.

[41] catalog.ldc.upenn.edu. (n.d.). NTIMIT - Linguistic Data Consortium. [online] Available at: https://catalog.ldc.upenn.edu/LDC93S2 [Accessed 15 Aug. 2022].

[42] registry.opendata.aws. (n.d.). Voices Obscured in Complex Environmental Settings (VOiCES) - Registry of Open Data on AWS. [online] Available at: https://registry.opendata.aws/lab41-sri-voices/ [Accessed 15 Aug. 2022].

[43] Richey, C., Barrios, M. A., Armstrong, Z., Bartels, C., Franco, H., Graciarena, M., Lawson, A., Nandwana, M. K., Stauffer, A., van Hout, J., Gamble, P., Hetherly, J., Stephenson, C., & Ni, K. (2018). Voices Obscured in Complex Environmental Settings (VOICES) corpus.

[44] www.voxforge.org. (n.d.). Free Speech... Recognition (Linux, Windows and Mac) - voxforge.org. [online] Available at: http://www.voxforge.org/home [Accessed 15 Aug. 2022].

[45] commonvoice.mozilla.org. (n.d.). Common Voice by Mozilla. [online] Available at: https://commonvoice.mozilla.org/en.

[46] M. Falcone and A. Gallo, "The "SIVA" speech database for speaker verification: description and evaluation," Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP '96, 1996, pp. 1902-1905 vol.3, doi: 10.1109/ICSLP.1996.608005.

[47] catalog.elra.info. (n.d.). The 'SIVA' Speech Database for Speaker Verification and Identification – ELRA Catalogue. [online] Available at: http://catalog.elra.info/en-us/repository/browse/ELRA-S0028/ [Accessed 15 Aug. 2022].

[48] Upenn.edu. (2015). Switchboard-1 Release 2 - Linguistic Data Consortium. [online] Available at: https://catalog.ldc.upenn.edu/LDC97S62.

[49] NIST 2021 Speaker Recognition Evaluation (SRE21). (2021). NIST. [online] Available at: https://www.nist.gov/itl/iad/mig/nist-2021-speaker-recognition-evaluation-sre21 [Accessed 15 Aug. 2022].

[50] sites.google.com. (n.d.). RedDots Project. [online] Available at: https://sites.google.com/site/thereddotsproject/home [Accessed 15 Aug. 2022].

[51] www2.imm.dtu.dk. (n.d.). ELSDSR - env. [online] Available at: http://www2.imm.dtu.dk/~lfen/elsdsr/index.php?page=env [Accessed 15 Aug. 2022].

[52] Hernandez, F. ois, Nguyen, V., Ghannay, S., Tomashenko, N., & Estève, Y. (2018). TED-LIUM 3: Twice as Much Data and Corpus Repartition for Experiments on Speaker Adaptation. In Speech and Computer (pp. 198–208). Springer International Publishing. https://doi.org/10.1007/978-3-319-99579-3_21

[53] Freesound (2012). Freesound. [online] Freesound.org. Available at: https://freesound.org/.

[54] Thiemann, J., Ito, N. and Vincent, E. (2013). DEMAND: a collection of multi-channel recordings of acoustic noise in diverse environments. [online] Zenodo. Available at: https://zenodo.org/record/1227121#.XRKKxYhKiUk [Accessed 15 Aug. 2022].

[55] www1.icsi.berkeley.edu. (n.d.). *ICSI Speech FAQ - 4.1 How is the SNR of a speech example defined?* [online] Available at: https://www1.icsi.berkeley.edu/Speech/faq/speechSNR.html#:~:text=SNR%20(Signal%2Dto%2Dnoise [Accessed 15 Aug. 2022].