UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Audio-Visual Speaker Verification

Diploma Thesis

Foteini Zakzagki

Supervisor: Gerasimos Potamianos

June 2025

UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Audio-Visual Speaker Verification

Diploma Thesis

Foteini Zakzagki

Supervisor: Gerasimos Potamianos

June 2025

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Οπτικο-Ακουστική Ταυτοποίηση Ομιλητή

Διπλωματική Εργασία

Φωτεινή Ζακζάγκη

Επιβλέπων: Γεράσιμος Ποταμιάνος

Ιούνιος 2025

Approved by the Examination Committee:

Supervisor **Gerasimos Potamianos**
Associate Professor, Department of Electrical and Computer Engineering, University of Thessaly

Member **Antonios Argyriou**
Professor, Department of Electrical and Computer Engineering, University of Thessaly

Member **Hariklia Tsalapata**
Laboratory Teaching Staff, Department of Electrical and Computer Engineering, University of Thessaly

Acknowledgements

I would like to thank Associate Professor Gerasimos Potamianos for his continued support and patience during the whole process of completing this thesis, as well as my friends and family, for helping me through the mentally hardest parts of it.

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

«Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I also declare that the results of the work have not been used to obtain another degree. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism».

The declarant

Foteini Zakzagki

Diploma Thesis

Audio-Visual Speaker Verification

Foteini Zakzagki

Abstract

In this thesis we present a multimodal approach to an audio-visual speaker verification system. Specifically, we combine three modalities, namely audio, face, and lip motion, in an attempt to surpass the results of uni-modal approaches and more importantly bi-modal architectures based on audio and face only. In order to achieve this, we employ neural network models, specific to each modality, to extract meaningful features and measure the performance of each individual stream, as well as their fusion, which is performed at the score level. The neural networks used for the implementation of the system were based on the TDNN, ResNet, MobileFaceNet, and MCNN models, making it possible for our multimodal system to outperform the baseline bi-modal approach.

Keywords:

Speaker Verification, Deep Learning, Multimodal System, Computer Vision, Audio Processing, Audio-Visual Fusion

Διπλωματική Εργασία

Οπτικο-Ακουστική Ταυτοποίηση Ομιλητή

Φωτεινή Ζακζάγκη

Περίληψη

Σε αυτή την διπλωματική εργασία παρουσιάζουμε μια πολυροϊκή προσέγγιση για ένα οπτικο-ακουστικό σύστημα επαλήθευσης της ταυτότητας του ομιλητή. Συγκεκριμένα, συνδυάζουμε τρεις ροές, ήχο, πρόσωπο, και κινήσεις χειλιών, σε μια προσπάθεια να ξεπεράσουμε τα αποτελέσματα των μονοροϊκών προσεγγίσεων και, πιο σημαντικά, των διροϊκών αρχιτεκτονικών, που βασίζονται αποκλειστικά στον ήχο και στο πρόσωπο. Για την επίτευξη αυτού του στόχου, χρησιμοποιούμε μοντέλα νευρωνικών δικτύων, εξειδικευμένα για κάθε ροή, ώστε να εξάγουμε ουσιαστικά χαρακτηριστικά και να αξιολογήσουμε την απόδοση κάθε μεμονωμένης ροής, καθώς και του συνδυασμού τους, ο οποίος πραγματοποιείται στο επίπεδο του σκορ. Τα νευρωνικά δίκτυα που χρησιμοποιήθηκαν για την υλοποίηση του συστήματος βασίστηκαν στα μοντέλα TDNN, ResNet, MobileFaceNet και MCNN, καθιστώντας δυνατό το να ξεπεράσει το πολυροϊκό σύστημά μας την βασική διροϊκή προσέγγιση.

Λέξεις-κλειδιά:

Ταυτοποίηση Ομιλητή, Βαθιά Μάθηση, Πολυροϊκό Σύστημα, Όραση Υπολογιστών, Επεξεργασία Ήχου, Οπτικο-Ακουστική Συγχώνευση

Table of Contents

List of Figures

List of Tables

Abbreviations

AAM	Active Appearance Model
ACM	Active Contour Model
ASM	Active Shape Model
ASR	Automatic Speech Recognition
AUC	Area Under the Curve
BN	Batch Normalization
CNN	Convolutional Neural Network
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
EER	Equal Error Rate
FAR	False Acceptance Rate
FC	Fully-Connected
FFT	Fast Fourier Transform
FRR	False Rejection Rate
GMM	Gaussian Mixture Model
GRU	Gated Recurrent Unit
HiLDA	Hierarchical Linear Discriminant Analysis
HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradients
LBP	Local Binary Pattern
LDA	Linear Discriminant Analysis
LPCC	Linear Predictive Coding Coefficient
LSTM	Long Short-Term Memory
MAP	Maximum a-Posteriori
MCNN	Multi-scale Convolutional Neural Network

MFA	Multi-layer Feature Aggregation
MFCC	Mel-Frequency Cepstral Coefficient
MTCNN	Multi-Task Cascaded Convolutional Neural Network
minDCF	minimum Detection Cost Function
NIST	National Institute of Standards and Technology
PCA	Principal Component Analysis
PLDA	Probabilistic Linear Discriminant Analysis
PLPCCs	Perceptual Linear Predictive Cepstral Coefficients
PReLU	Parametric Rectified Linear Unit
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
ROI	Region of Interest
SE	Squeeze-Excitation
SGD	Stochastic Gradient Descent
SRE21	Speaker Recognition Evaluation 2021
SVM	Support Vector Machine
TCN	Temporal Convolutional Network
TDNN	Time Delay Neural Network
UBM	Universal Background Model

Chapter 1

Introduction

Speaker recognition is a field that has attracted significant interest in the literature for several decades now. It is comprised of two main tasks: speaker identification, where the identity of a speaker from a set of known speakers is determined, and speaker verification, where a claimed speaker is verified to indeed be the person they claim to be. Traditionally, speaker recognition has depended mainly on audio processing due to the rich information contained in the human voice and its discriminative qualities. Despite its advancements however, speaker recognition faces inherent challenges. One of the most significant is the variability and potential degradation of audio quality, which can render sound an unreliable modality in certain scenarios due to recording conditions and transmission channels. To address these limitations, researchers have increasingly turned to multimodal approaches, integrating visual data alongside audio to enhance system robustness and reliability. These efforts are critical, as the applications of speaker recognition span a wide range of domains, including security and authentication [1], [2], personalization in consumer electronics like smart assistants [3], and law enforcement and forensics [4]. Consequently, there is a boom of interest in the development of reliable speaker recognition systems that utilize more than one modality, particularly given the wide range of applications for this technology. Scientists had been mainly approaching the subject with the use of handcrafted features and probabilistic models, until machine learning algorithms and neural networks were introduced to the field. Their ability to model speaker characteristics from raw data, integrate multiple modalities, and generalize to complex, real-world scenarios has rendered deep-learning, and particularly end-to-end models, the leading technology used in the area of speaker recognition.

1.1 Thesis Objective

In this thesis we will investigate the impact of incorporating the lip modality (i.e., visual speech) alongside a traditional neural network-based audio-visual speaker verification system. Existing audio-visual systems primarily rely on full-face visual data and an audio track. The former can present certain limitations in some challenging environments, such as low resolution, non-frontal views, lighting conditions, or occlusions, while the latter can prove unreliable in scenarios where the data is noisy due to recording conditions and transmission channels as previously mentioned. Leveraging lip movements, which contain speaker-related discriminative information, could potentially enhance robustness across different viewing angles. Effectively performing the fusion of the three modalities is crucial, which is why there is a focus in this work on modality alignment, score normalization, and weight distribution. These techniques are used in order to balance the contribution of each modality, ensuring optimal performance. The system design is inspired by the 2021 NIST Audio-Visual Speaker Recognition Challenge (SRE21) [5], which introduced a baseline system for audio-visual speaker verification using an extended Time Delay Neural Network (TDNN) [6] for audio and a ResNet101 [7] model for face recognition. However, due to system limitations, this study uses slightly different architectures trained and evaluated on the Lombard Grid database [8] rather than the dataset provided in SRE21. Each modality is processed independently using state-of-the-art models, namely an ECAPA-TDNN [9] for the audio, a MobileFaceNet [10] with ArcFace loss [11] for the face, and a Multi-scale Convolutional Neural Network (MCNN) [12] for the lip modality. The outputs of the three modalities are combined through score fusion with normalized weighting, after calculating the scores across modalities.

1.2 Contribution

This thesis aims to investigate the integration of the lip modality in an audio-visual speaker verification system to improve performance and robustness under challenging conditions. In order to achieve this:

- Different neural networks were implemented to process each modality independently. Inspired by the baseline systems and top performing models [13] in the NIST SRE21 challenge, an ECAPA-TDNN architecture was used for the audio modality. The ResNet models, though mainly used in the challenge and despite their excellent performance, were not selected for this thesis due to system limitations, and a modified MobileNet architecture was utilized instead for the face modality along with an MTCNN-based face detector from [14]. Lastly, a third lip modality was implemented alongside the traditional two (audio and face), using an MCNN model based on [15] and [16].
- Score fusion was used to first combine the outputs of the audio and face modality, and the same was done for the combination of all three modalities. The fusion was performed using normalized weighting and score alignment, to ensure the balanced contribution of each modality.
- Each individual model and the fused systems were trained and evaluated on the Lombard Grid database. System performance was measured using Equal Error Rate (EER) and the minimum Detection Cost Function (minDCF), and performance comparisons among all systems were made.

1.3 Thesis Structure

The rest of the thesis is structured as follows:

- Chapter 2: Related Work – Contains a historical overview of the methods used for speaker verification for each individual modality, as well as the different fusion techniques used to combine them.
- Chapter 3: Data and Preprocessing – Describes the Lombard Grid database and the way it is split for training and testing, as well as the preprocessing steps required for each modality before the main pipeline.
- Chapter 4: Methodology – Presents the model architecture and training protocol of each modality stream, along with the evaluation protocol and the fusion techniques used.
- Chapter 5: Experimental Results – Analyzes the single-stream performance results and compares them to the performance of the fused architectures.
- Chapter 6: Conclusion – Provides a summary of the thesis and a discussion of future research approaches.

Chapter 2

Related Work

2.1 Audio-Based Approaches

Speaker recognition systems rely on audio-based approaches to extract, model, and classify speaker-specific characteristics for verification and identification tasks. The evolution of these methods has progressed from early handcrafted features and statistical models to modern deep-learning techniques that automatically learn high-level representations.

Audio features are specific elements of a speech signal that capture the individuality of a speaker's voice. In the early stages of speaker recognition, researchers primarily relied on direct waveform characteristics like amplitude, frequency, and duration of the speech signal [17], [18], [19], [20], [21]. These features could easily be extracted from the raw audio waveform without complex processing or analysis, which wasn't easily feasible at the time, due to hardware limitations.

These early systems, however, were not able to handle the variability in speech signals caused by differences in speaking styles, environmental conditions, and recording devices. To address these problems, researchers developed feature extraction techniques based on spectral and cepstral analysis, which were able to capture far more complex characteristics of a speaker's voice by analyzing the frequency components in the speech signal [22], [23]. Among these, Mel-Frequency Cepstral Coefficients (MFCCs), introduced by Bridle and Brown in 1974 [24], became a cornerstone of feature extraction during the 1980s. MFCCs use the Mel scale to mimic the non-linear frequency sensitivity of the human auditory system, capturing spectral energy distributions over time. Their ability to encode speaker-specific information while reducing noise sensitivity constituted a breakthrough in audio-based speaker

recognition systems.

Driven by the effectiveness of MFCCs in capturing these complex speech characteristics, new kinds of coefficients, such as Linear Predictive Coding Coefficients (LPCCs) and Perceptual Linear Predictive Cepstral Coefficients (PLPCCs), were developed by researchers. While both LPCCs and PLPCCs are based on linear predictive modeling of the signal, LPCCs modeled the vocal tract's filter characteristics and were particularly effective in text-dependent applications [25], whereas PLPCCs introduced a perceptual weighting function to the power spectrum, emphasizing perceptually relevant components [26], making them known for their robustness, particularly in noisy environments.

The integration of these features with statistical modeling techniques led to significant progress in the field. Hidden Markov Models (HMMs) [27], which were already being used in automatic speech recognition (ASR) since the 1970s, were adapted for speaker recognition during the 1990s [28] and since then have been a staple in the field. These statistical models excelled at capturing the temporal dynamics of speech, allowing them to account for the variations in a person's speech across different utterances and conditions. HMMs could adapt to the specific speaking style and characteristics of each speaker, making them more accurate in distinguishing different speakers, as well as handling background noise. Their ability to model continuous speech and scale to accommodate numerous speakers is what established HMMs as a benchmark in the field of speaker recognition.

During the same period, Reynolds and Rose introduced in [29] the concept of using Gaussian Mixture Models (GMMs) to model speaker-specific characteristics in speech and showed that GMM-based systems could be effective in text-independent speaker identification [30], [31]. The GMMs were used to model the distribution of acoustic features, like MFCCs, for individual speakers. They represented the speaker's voice characteristics as a mixture of multiple Gaussian distributions, each modeling a different feature of the speaker's voice. Since then, GMMs have been a fundamental component of many speaker recognition systems. A major advancement in this area was the introduction of the GMM-Universal Background Model (GMM-UBM), which provided a speaker-independent baseline for adapting individual speaker models via Maximum a-Posteriori (MAP) adaptation. This adaptation process enhanced robustness against limited training data and variability, which was crucial for reliable speaker recognition [32].

In 2010, Dehak et al. [33] introduced i-vectors, a revolutionary technique for speaker

recognition. I-vectors are low-dimensional representations that efficiently capture speaker-specific characteristics by applying factor analysis to GMM-UBM supervectors. Probabilistic Linear Discriminant Analysis (PLDA) was often paired with i-vectors to model intra- and inter-speaker variability, further improving classification robustness [34]. This approach excelled in distinguishing between speakers, even when limited training data was available, while the i-vectors' ability to handle session and channel variability made them a preferred choice for real-world applications. Their discriminative power and adaptability made it possible to integrate them with modern machine learning methods, contributing to their widespread use in speaker recognition, surpassing traditional features like MFCCs in terms of precision and efficiency. While newer methods have emerged, i-vectors remain a crucial component of the speaker recognition toolkit and have become a benchmark for future works.

The introduction of deep-learning marked a new era in speaker recognition. Instead of relying on handcrafted features, researchers began using deep neural networks to extract speaker embeddings directly from the raw audio input. More specifically, Snyder et al. [35] introduced a new concept of deep neural network-based representations of speaker identities, which utilized TDNNs to capture both short-term and long-term dependencies in speech [36]. Unlike traditional handcrafted features such as MFCCs and i-vectors, these embeddings, named x-vectors, are learned directly from the raw audio signal, making them capable of capturing complex and abstract speaker characteristics. Classification of x-vectors was performed using either cosine similarity or PLDA, with substantial improvements reported across various datasets and conditions [34], [35]. This approach allows for the automatic extraction of high-level speaker characteristics, which are more robust and informative than traditional acoustic features, making it particularly effective in both text-dependent and text-independent speaker recognition tasks, even in challenging acoustic conditions and noisy environments. X-vectors have gained widespread popularity and have become a new benchmark in the field, enabling state-of-the-art performance in various speaker recognition applications. Their ability to capture complex speaker characteristics and adapt to different data conditions has made them a fundamental component in the development of modern speaker recognition systems, advancing the accuracy and robustness of these systems in real-world scenarios.

Other deep-learning models further expanded the field. Convolutional Neural Networks (CNNs) extracted hierarchical features from spectrograms, enabling systems to capture both

local and global patterns in speech signals [37]. Long Short-Term Memory (LSTM) networks, known for modeling temporal dependencies, refined these features by capturing the sequential nature of speech data [38]. Hybrid architectures, such as the one implemented in [39], where a combination of a CNN, an LSTM, and a TDNN was used to extract speaker representations, have given models the ability to focus on the most relevant portions of the audio signal producing more discriminative speaker embeddings.

The use of hybrid models that combine deep-learning with traditional techniques such as GMMs and HMMs [40], [41] has also been a prevalent approach in this line of research. These hybrid systems take advantage of the deep-learning models' ability to generate high-discriminative speaker embeddings, while using the traditional models to reliably capture the temporal dynamics and enhance noise robustness.

End-to-end training frameworks have also become increasingly popular in recent years, streamlining feature extraction, modeling, and classification into a unified pipeline. These systems directly yield speaker embeddings from raw audio and optimize speaker verification loss functions, simplifying the design process and improving overall performance [6], [42].

The progression from handcrafted features like MFCCs and LPCCs to learned embeddings like x-vectors has significantly improved the accuracy and robustness of speaker recognition systems, while the transition from statistical and probabilistic modeling techniques like HMMs and GMMs to end-to-end deep-learning systems has enabled reliable performance in challenging environments with background noise, reverberation, and channel mismatch. These advancements have made speaker recognition systems more reliable, rendering them more effective in real-world applications.

2.2 Visual-Based Approaches

Visual-based speaker recognition methods focus on extracting, modeling, and classifying speaker-specific visual features such as facial structure, lip movements, and expressions. Similarly to the audio modality, these methods have evolved from early handcrafted techniques to deep-learning-based approaches that can automatically extract and model discriminative features from facial and lip movement data.

2.2.1 Face-Based Methods

Speaker recognition based on visual data has evolved significantly over the years, beginning with early approaches that focused on pixel-based and appearance-based methods. Pixel-based methods, such as optical flow and pixel intensity analysis, were among the first techniques used to analyze facial features. Algorithms like Lucas-Kanade [43] and Horn-Schunck [44] flow played a crucial role in tracking facial movements by analyzing pixel displacements across consecutive video frames. These methods extracted motion-based features, which were particularly effective in capturing lip movements and facial expressions. These features were then modeled using HMMs [45], which captured temporal dependencies in facial motion. Classification was performed by computing the likelihood of an observed sequence belonging to a trained speaker model, enabling systems to recognize patterns in dynamic facial movements.

Appearance-based methods utilized statistical methods such as Histogram of Oriented Gradients (HOG), Local Binary Pattern (LBP), and Gabor filters to capture texture and shape-based features from the face. HOG was particularly effective in edge detection, which is crucial for identifying facial contours [46]. LBP encoded local texture by analyzing patterns of pixel intensity differences, making it robust to illumination changes [47]. Gabor filters captured spatial frequency and orientation features, enabling multi-resolution and multi-scale analysis of facial characteristics [48].

Much like in audio-based recognition, the extracted facial features were modeled using HMMs [49], GMMs [50], or Support Vector Machines (SVMs) [51] for classification, capturing speaker-specific characteristics and providing clear separation of speaker identities in high-dimensional feature spaces. SVMs are a type of supervised machine learning algorithm that works by finding the optimal hyperplane that best separates different speaker classes in a high-dimensional feature space, based on their speech and/or facial characteristics.

Combined with dimensionality reduction techniques such as Principal Component Analysis (PCA), which identified key features with the highest variance, and Linear Discriminant Analysis (LDA), which maximized inter-class separability, these models could ensure computational efficiency while effectively managing and classifying high-dimensional data [52]. An approach called Eigenfaces [53] introduced the idea of using PCA for face recognition, which was previously widely used in pattern recognition and signal processing, revolutionizing the field. While Eigenfaces are no longer state-of-the-art, they played a crucial role in

transitioning face recognition from template-matching techniques to data-driven statistical approaches.

In the past decade, deep-learning has revolutionized face-based speaker recognition. CNNs have proven highly effective in automatically learning hierarchical features from facial images. Popular architectures such as Residual Networks (ResNets) [54], [55], VGG-Face [56], Inception [57], FaceNet [14], and Siamese Networks [58] have enhanced face recognition accuracy by learning robust facial embeddings. The ResNets' skip connections allowed deeper networks to avoid vanishing gradient issues, while VGG-Face provided a fine-tuned architecture for facial recognition tasks. Inception networks introduced multi-scale convolutional filters to capture features at various resolutions, FaceNet utilized a triplet loss to generate highly discriminative facial embeddings, and Siamese networks employed pairwise comparisons to learn similarity metrics for speaker verification tasks. These learned embeddings were then classified using various Softmax layers, where probabilistic outputs determined the speaker's identity. With the rise of deep-learning, the boundary between feature extraction and speaker modeling has become increasingly blurred, leading to end-to-end systems that optimize recognition performance.

The transition from handcrafted features to deep-learning-based embeddings marked a significant leap in facial recognition, enabling systems to achieve higher accuracy and robustness across diverse conditions. Deep-learning methods, combined with traditional statistical models, when necessary, have established a strong foundation for state-of-the-art visual-based speaker recognition systems, which have become a vital tool for applications in security, surveillance, and user authentication.

2.2.2 Lip-Based Methods

Lip-related features were originally developed for speech recognition rather than speaker recognition. Early methods focused on analyzing lip movements to recognize phonemes and words, improving ASR in noisy environments. Techniques such as Optical Flow, Active Shape Models (ASMs) [59], Active Contour Models (ACMs) [60], and Active Appearance Models (AAMs) [61], [62] were extensively used for modeling and analyzing the appearance and shape variations of the face, with a focus on eyes and lips. Optical flow algorithms tracked pixel displacements in the lip region to extract motion-related features, like in [63], where the Horn-Schunck algorithm was used to compute variance and integral features, effectively cap-

turing articulatory patterns over time. ASMs and AAMs went beyond simple motion tracking by modeling the shape and appearance of lips through landmark points and texture mapping, respectively. ASMs represented the lips as a set of landmark points, and variations in the positions of these points were used as features, while AAMs combined shape and texture information, allowing for the extraction of detailed lip features. ACMs, or 'Snakes', were used to derive geometric features, specifically the contours of the lips, in order to track lip movements regardless of pose or different lighting conditions.

Based on Eigenfaces, the concept of Eigenlips was introduced in 1994 [64] by Bregler and Konig and became widely used during the 2000s. Eigenlips, like Eigenfaces, use PCA to capture the principal modes of lip shape variation. They were extracted from a dataset of lip images, and the coefficients of these models served as features.

HMMs were used for the modeling and classification of lip features as well, especially in lip-reading applications like the ones described in [65], [66], where the Discrete Cosine Transform (DCT) and the Discrete Wavelet Transform (DWT) were used for feature extraction. The DCT split the image into parts of differing importance taking into account the visual quality of the image, by transforming it from the spatial domain to the frequency domain. The DWT separated the data into different frequency components and then studied each component with resolution matched to its scale. These features were given as inputs to estimate the parameters of the HMM, which captured their temporal variations.

A comparison of model-based methods like AAMs and transform-based methods, such as PCA and DWT, was made in [67], showing that the latter approach achieved better performance. Nevertheless, both of these methods formed the foundation for leveraging lip movements in ASR systems.

The use of lip-based features in speaker recognition emerged later, building on the foundations established in speech recognition research. Researchers began adapting these techniques to capture speaker-specific patterns in lip movement, focusing on subtle articulatory variations that could distinguish individuals. GMMs were used to model the distribution of lip features, while SVMs classified speaker identities by separating them based on their lip movement characteristics [68], [69]. These methods demonstrated that lip-based features could effectively complement audio-based approaches in speaker recognition tasks.

These traditional approaches were effective in modeling the temporal and statistical aspects of lip movements and offered robust solutions for various applications, but as tech-

nology evolved, researchers shifted towards more data-driven and deep-learning-based approaches, which have shown even better performance and generalization capabilities. The most prominent deep-learning methods for lip feature extraction and processing are CNNs and Recurrent Neural Networks (RNNs) [70], as well as their combination [71], [72]. Similar to face recognition, CNN architectures such as VGG and Inception, were particularly effective in capturing texture and shape information from static lip frames. While RNNs are not commonly used for general face recognition, they can be valuable in specific cases where temporal information, such as lip movements, is essential for the task at hand, as is the case with lip feature extraction for lip-reading. Unlike CNNs, which excel at capturing spatial features from static images, RNNs are designed to model sequential and temporal dependencies in data. This capability makes RNNs, such as LSTMs [73], [74] and Gated Recurrent Unit (GRU) networks [75], a great tool for ASR, since the network learns to associate specific lip movements with phonemes or words through backpropagation and gradient descent during training.

Hybrid CNN-LSTM models have been a significant step forward in lip processing, by combining spatial and temporal feature extraction into a unified, end-to-end system. LipNet [76] is a deep-learning architecture designed for lip-reading, which combines convolutional and recurrent layers to process lip movement data and has achieved state-of-the-art results in lip-reading tasks. CNN components process spatial features from lip images, while LSTM layers capture the sequential dependencies between frames. In [77], the authors utilized residual networks in combination with Bidirectional GRUs to directly extract and model features from the raw audio and labial data. This integration allowed hybrid models to achieve state-of-the-art performance in lip-based speaker recognition, as they effectively model both the static and dynamic aspects of lip movements.

Recent advancements have also focused on incorporating attention mechanisms to further enhance lip-based speaker recognition. In [78], the authors propose an attention-based pooling mechanism that effectively captures and combines relevant visual features over time, improving the model's ability to recognize speech from silent videos. This approach allows the model to focus on the most informative lip features during speech, leading to more accurate recognition. Transformer-based architectures, with their self-attention mechanisms [79], have also been effectively used for modeling long-range dependencies in lip movement patterns, as demonstrated in [80], where a Transformer was used to pre-process the features, highlighting

the important long-term temporal patterns, before passing them to a Temporal Convolutional Network (TCN). Transformers have, overall, significantly advanced lip-reading and speaker verification by improving feature extraction, temporal modeling, and multimodal fusion with their ability to extract fine-grained spatiotemporal features, and enhance cross-modal fusion [81], aligning audio and visual streams dynamically.

From early handcrafted techniques like Optical Flow, ASMs, and AAMs to modern deep-learning-based approaches including CNNs, RNNs, and attention mechanisms, lip-based speaker recognition has seen substantial progress. The combination of spatial and temporal modeling has not only advanced the feature extraction process from lip-related data but also established the lip modality as a vital branch in performance improvement of multimodal speech recognition systems.

2.3 Audio-Visual Fusion Techniques

Evidently, both acoustic and visual based approaches produce great results in speaker recognition. This has driven many researchers to combine the two modalities in hopes of addressing the challenges associated with audio-visual speaker recognition, such as dealing with noise, channel mismatches, synchronization, lighting conditions, and variations in speaker pose [82], [83]. There are several fusion strategies used to combine audio and visual data, whether that be the face or the lips, depending on the specific application and the nature of the data. The most common fusion strategies can be categorized as early (or feature-level) fusion, mid-level fusion, and late (or decision-level) fusion.

2.3.1 Early Fusion

In early fusion, features from both audio and visual modalities are combined at an early stage, typically before the main feature analysis. This approach allows the system to take advantage of the inherent correlations between the two modalities at the feature level. Audio and visual features might be concatenated [84], [85], summed [86], or averaged [87] into a single feature vector, which is then fed into models for further analysis. For instance, [84] concatenates MFCCs with Eigenfaces, creating new speaker representations that capture complementary information from both modalities and are then exported to a neural network framework for recognition.

Despite its advantages, early fusion is sensitive to synchronization issues and modality-specific noise. If one modality is degraded, the fused representation may introduce noise into the pipeline, reducing overall performance. To address these challenges, researchers often preprocess the data to ensure temporal alignment and filter out noise before fusion.

2.3.2 Late Fusion

In late fusion, the audio and visual data are separately processed, and their results are combined at a later stage, typically after feature extraction and individual modality analysis [88]. This approach allows for flexibility in processing each modality independently and can be useful when the two modalities are not directly comparable or when there is a need to handle missing or unreliable data from one modality. For instance, in noisy environments where audio quality is compromised, the system can rely more heavily on visual data for decision-making, and vice versa. Techniques like voting, weighting, or averaging [83], [88], [89], as well as various neural networks, have been employed to fuse the scores computed for each individual modality.

2.3.3 Mid-Level Fusion

Mid-level fusion [90] involves combining information from both modalities at an intermediate stage of processing. This typically includes learning a shared embedding space where data from both modalities are mapped to a common representation. This shared space makes it possible for the system to compensate for any important missing or degraded data from one stream, by using cues from the other. An early implementation of this approach was proposed by Potamianos et al. in [91], where Hierarchical LDA (HiLDA) was introduced, in order to enhance the inter-class and intra-class discrimination of the audio-visual features. In recent years deep-learning models such as CNNs for visual data and RNNs for audio data can be combined to extract multimodal embeddings, just like in, previously mentioned, LipNet, VFNet, introduced in [55], or DeepLip, which combined a CNN with a TDNN model in [16].

2.3.4 Hybrid Fusion

Hybrid fusion strategies involve combining multiple fusion techniques to tackle complex problems. In this case, early fusion may be employed for some aspects of the problem, while

late fusion is applied to others, making it easier to integrate necessary information from various stages of the processing pipeline [13], [92]. As mentioned in Subsection 2.2.2, attention mechanisms play a crucial role in cross-modal fusion. By dynamically weighting or focusing on different parts of the data at various time steps, they manage to prioritize their most discriminative parts. For instance, an attention module might emphasize lip movements during periods of noisy audio, or rely more on vocal characteristics when visual data is partially occluded. This adaptive use of attention mechanisms enables the system to select the most relevant modality for a given task, enhancing its overall performance.

It's important to mention that in applications where audio and visual data are synchronized in time, like speaker recognition, temporal alignment techniques should be used to align the two modalities so that they can be fused more effectively.

The choice of fusion strategy depends on the specific task, the nature of the data, and the requirements of the application. Researchers often experiment with different fusion techniques to determine which one works best for a particular problem. Regardless of which fusion approach is chosen, it is clear that the combination of more than one modalities can produce more robust and accurate performance in speaker recognition systems, making them reliable for real-world applications.

Chapter 3

Data and Preprocessing

In this chapter, we focus on the data used to conduct our experiments, as well as the preprocessing steps required to transform them into a suitable format for each main processing pipeline. In detail, the Lombard Grid database is analyzed in Section 3.1, while the appropriate splitting into training and test sets is discussed in Section 3.2. Lastly, in Section 3.3 we present the three different ways we handle the preprocessing of the audio, the face, and the lip stream respectively.

3.1 The Lombard Grid Database

The Lombard Grid database is an extension of the GRID corpus, containing audio-visual data based on the Lombard effect, and is primarily used for speech perception. Specifically, the dataset includes 54 speakers, 30 female and 24 male. Each speaker has 100 utterances (50 Lombard and 50 plain), yielding 5,400 utterances in total. It is organized in three directories, which contain synchronized audio recordings, videos of the front view, and videos of the side view of the speakers, respectively. The audio recordings are in .wav file format, with a sampling rate of 16kHz, while the videos are in .mov file format, with a 480p resolution (720×480), and a frame rate of 25 fps. In any case, their filenames are arranged as follows:

`SPKR_COND_UTTERANCE.wav|.mov` - e.g. `s8_p_sbbi9p.wav`, where:

- SPKR = s2 to s55 (s1 is not available due to technical issues)
- COND = l or p, where `l` $\Rightarrow$ Lombard, `p` $\Rightarrow$ plain
- UTTERANCE = 6-character Grid utterance code, e.g., `pgag6a`, which means *"place*

green at g 6 again"

If a sentence is spoken incorrectly, the filename follows this format:
`SPKR_COND_[WRONG_UTTERANCE]_WRONG_[CORRECT_UTTERANCE].wav`,
e.g: `s8_2_38_8_r_lrwizp_WRONG_lrbizp.wav`

This database was chosen because it offers a balanced gender distribution (30 female and 24 male speakers), with variation in facial hair and skin tones, as well as variations in pose and lighting. It also inherently shows voice variability since it includes both Lombard and plain speech. Note that Lombard speech is characterized by an increase in vocal effort, including higher intensity, pitch, and articulation changes, in response to noise. The quality of the videos was an important criterion as well, given that the extraction of lip regions requires high-resolution frames in order to yield informative features.

3.2 Data Splitting Strategy

The dataset is divided into training and test sets using an 80%-20% ratio, meaning that each speaker's samples are split so that 80% of their recordings are allocated to the training set, while the remaining 20% are allocated to the test set. While the same 54 speakers are present in both the training and test sets, there is no overlap in the individual samples used for training and testing, which is crucial for preventing overfitting due to data leakage.

The same split was performed on all three folders (audio, frontal visual, and side visual), to make sure the samples are aligned across modalities, so that the final fusion can be seamlessly executed and provide meaningful results.

It should be noted that a small number of the video recordings in the database were corrupted and were therefore manually removed. This resulted in a training set of 54 speakers with approximately 80 samples per speaker, and a test set of 54 speakers with approximately 20 samples per speaker, shown in Table 3.1.

Given that the database is not very large, the training set is also used for 10-fold cross-validation. This approach is employed to make sure hyperparameter tuning and performance evaluation during training are carried out, without the need for an additional validation set. Each fold is constructed by stratifying samples to ensure that all speakers are well-represented in each subset, making the model more robust.

Table 3.1: Dataset partition

Subsets	# Speakers	# Samples
Training	54	4316
Testing	54	1080

The test sets used for the evaluation are created with a random selection of 20,000 trial pairs, which include 4,000 target (same speaker) and 16,000 non-target (different speakers) pairs. The 1:4 ratio is used to mimic real-world scenarios, where the comparisons between different speakers are more common than those between the same person, and the detection of impostors is highly important.

3.3 Preprocessing

Preprocessing is a crucial step in the development of a robust audio-visual model, as it directly impacts the accuracy and efficiency of downstream tasks. Both audio and visual data require some processing in order to be transformed into meaningful representations that can be used by the main processing pipeline. Each modality calls for unique handling, since they are subject to different limitations and contain distinct discriminative information. Proper preprocessing can significantly increase the quality of the data, as well as ensure alignment between modalities, thereby improving the overall performance of the model.

3.3.1 Audio Preprocessing

The audio stream does not require significant preprocessing, since the model used for this modality has the ability to extract high discriminative features directly from raw data. Thus, the only steps we took before feeding the audio files to the model was to split them into segments of fixed length at a 16kHz sampling rate with a 15ms window to ensure smooth transitions between overlapping frames.

In case the segment was less than the specified fixed length, it was wrapped with padding, instead of zero-padding, to prevent artificial silence and maintain the natural characteristics of the speaker's voice (see Figure 3.1). The fixed length is determined by the duration of input segments given by the user, which in our case is 200 frames, corresponding to 2 seconds.

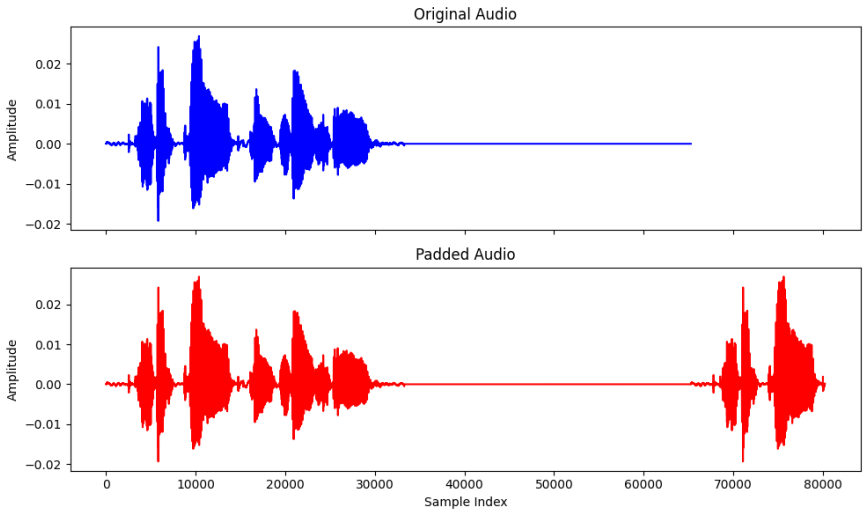


Figure 3.1: Padded waveform in relation to original raw waveform

Some additional processing is done by the ECAPA-TDNN model to transform the waveform into the Mel frequency domain, but that will be further discussed in Subsection 4.1.1.

3.3.2 Face Preprocessing

Unlike the audio stream, the face stream requires quite a bit of preprocessing, even when we use an end-to-end system. Face detection is a vital step in this pipeline, since it detects and extracts the faces present in visual data, be it videos or images, as the name suggests. Due to computational limitations, the video recordings we possess cannot be used in their raw format for live detection, so we instead extracted 4 (random) frames from each video, in order to form a "new" face database. The choice of 4 frames was directly determined by the available GPU capacity in parallel to the need for sufficient data representation.

For face detection, we used an MTCNN model from a PyTorch implementation of FaceNet, which provided the most accurate detection on our dataset. The MTCNN extracted the face Regions of Interest (ROIs) by finding the facial landmarks and computing the bounding boxes of the faces in the image, while automatically aligning them. All the extracted faces were then resized to 112×112, which is the required input size of the CNN model used for training. Those were, in turn, grouped into folders based on the speaker's label, and wrongly detected faces were manually removed to ensure the data is clean and accurate. An example of the cropped and aligned face ROIs can be seen in Figure 3.2.

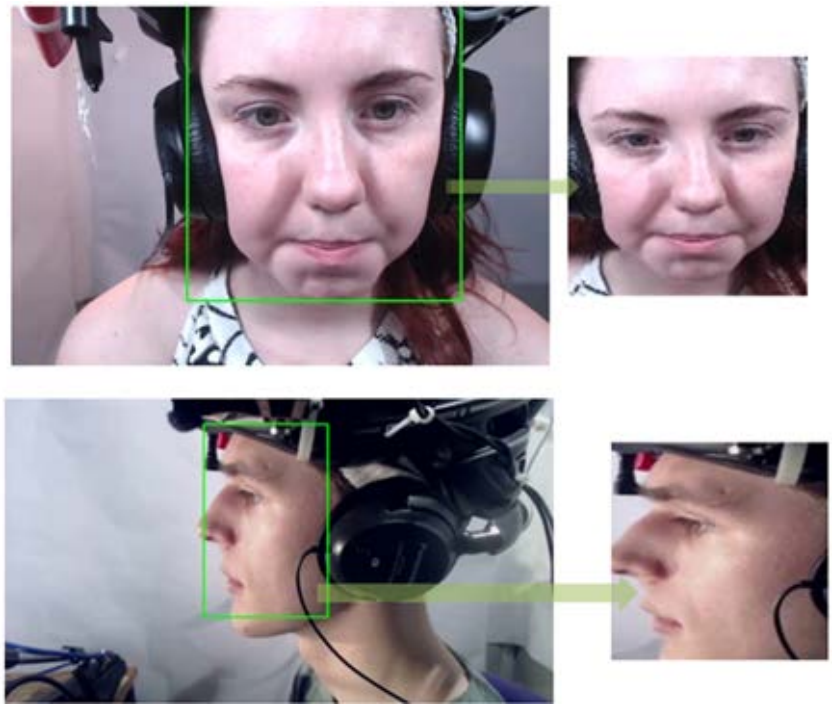


Figure 3.2: Detected face in a frame to cropped and aligned face ROI

After this “new” face database was created, it was split into training and test sets in the same way we divided the audio set. By using the same split as the audio dataset, we make sure there’s consistency across modalities, so the final results can be meaningfuly combined. The same preprocessing steps are applied to both the frontal and the side view.

3.3.3 Lip Preprocessing

Similar to the face modality, lip detection is necessary in the lip modality. Using a face alignment module, we extracted 2D facial landmarks from all the videos in the database, which were then saved in .npz format for easier access and processing. Video files that can’t be processed were logged in a failure file for error resolution, meaning they were manually removed.

The 68-landmark model of Multi-PIE [93] (see Figure 3.3) was preferred since it provides a stable reference for the mouth region across all aligned frames. The landmarks were used to obtain the mean faces both for the front and the side view, which are necessary for the face alignment.

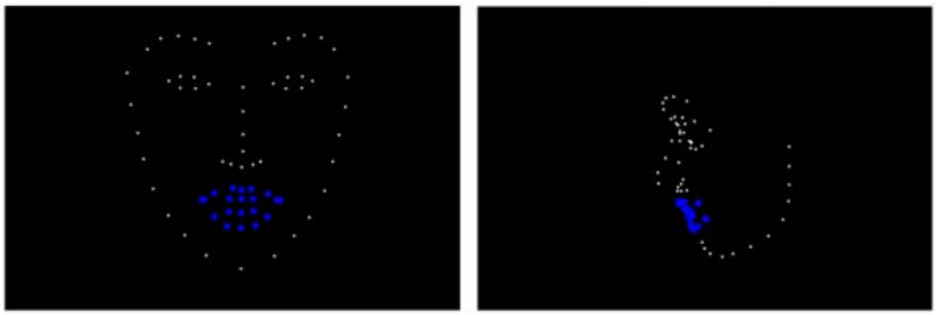


Figure 3.3: 68-landmark face model (Mouth stable points are highlighted in blue)

The mouth ROIs were then cropped, based on the model's stable points (48-68 for the mouth region for both front and side views). They were then converted into grayscale and resized to 96×96 before being saved in .mp4 format as shown in Figure 3.4.

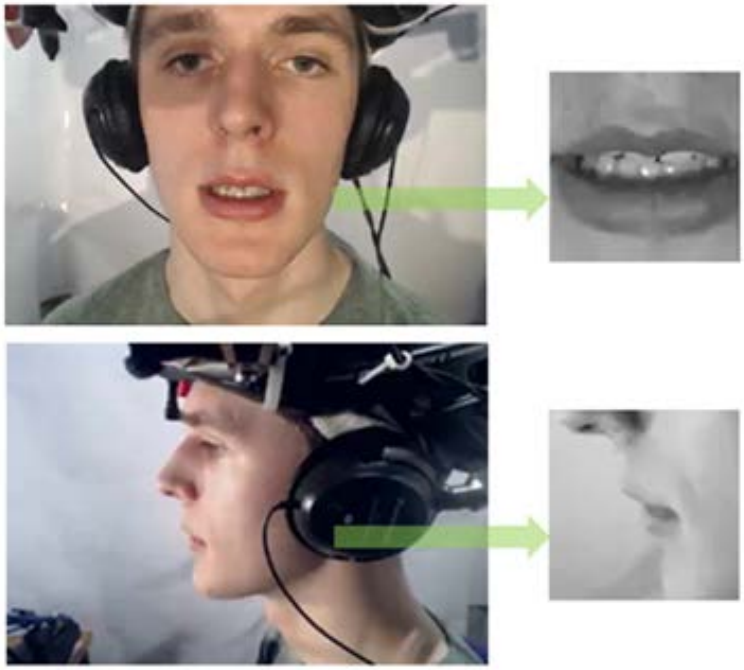


Figure 3.4: Extracted lip ROIs from front and side views

Chapter 4

Methodology

In this chapter, we present an overview of our audio-visual speaker recognition system, which utilizes three independently processed streams—audio, face, and lips— for the successful completion of the task. In particular, Section 4.1, 4.2, and 4.3 describe the models selected for the audio, face, and lip modality respectively, diving into their architectures and training procedures. Section 4.4 presents the performance metrics used to evaluate each stream, as well as the final fused system, while Section 4.5 focuses on the fusion strategies chosen for the effective combination of the three modalities.

The choices regarding the development of this speaker verification system were made, bearing in mind the importance of meaningful contribution from each stream so as to make sure the overall system performance is improved.

4.1 Audio Stream

For the implementation of the audio stream we use an ECAPA-TDNN model, which has been proven to outperform traditional TDNN architectures in speaker verification benchmarks [9]. This model incorporates SE-Res2Blocks, which, according to [9], enable it to better model the temporal dependencies of the recording by rescaling the channels based on its global properties. Furthermore, unlike the traditional TDNN, it aggregates feature maps at multiple levels, leveraging the ability of neural networks to extract features of varying complexity at every layer. Finally, the attentive statistics pooling (ASP) layer that is utilized instead of a simple statistics pooling (SP) one, makes the model capable of focusing on important frame-level features for every channel. In summary, the incorporation of channel

attention mechanisms in the traditional TDNN architecture, allow the ECAPA-TDNN model to dynamically extract more relevant speaker-specific features, which renders this a reliable approach for handling the audio modality.

4.1.1 ECAPA-TDNN Model Architecture

Before we dive into the main architecture of the model, we have to mention the preprocessing module that it contains and was briefly introduced in Subsection 3.3.1. In this module, a series of transformations is applied to the raw audio waveform in order to generate the corresponding Mel spectrogram. A pre-emphasis filter is first applied to the signal in order to enhance the higher frequencies and provide a more balanced spectral energy distribution. The pre-emphasis filter is followed by a transformation, which computes the spectrograms in the Mel-frequency domain. The Mel spectrograms are extracted from each audio file in 80 Mel bins with a sampling rate of 16kHz and a 512-point Fast Fourier Transform (FFT) with a Hamming window with a length of 25ms and a 10ms frame shift. A log transformation is then applied, followed by mean normalization, to standardize the spectrogram and reduce the impact of variations in amplitude. These steps ensure the audio data is in an optimal form to be used as model input and can be seen in Figure 4.1

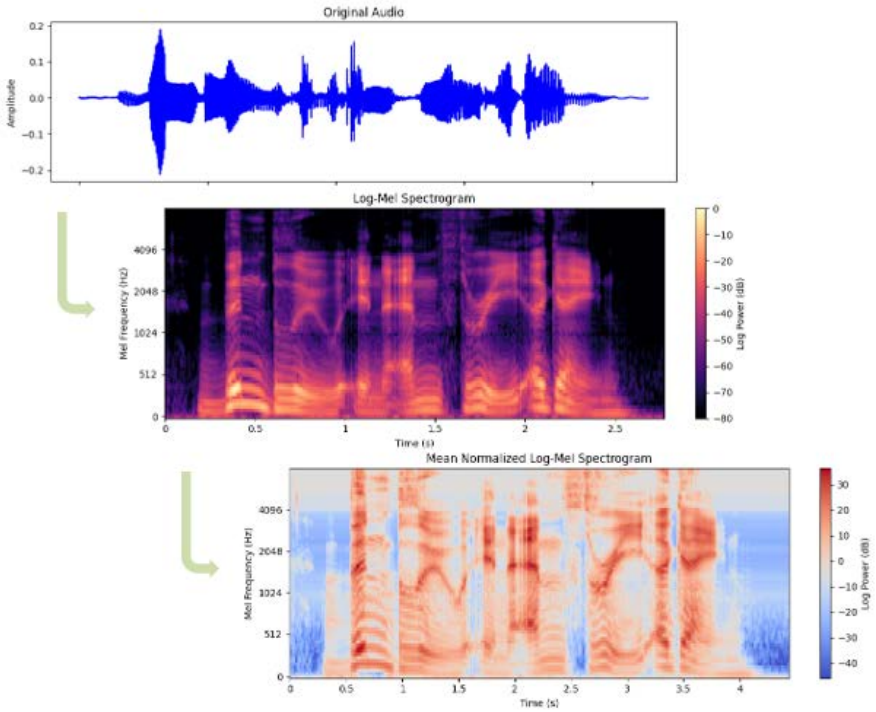


Figure 4.1: Mean-normalized log-Mel spectrogram extraction

Figure 4.2: ECAPA-TDNN speaker model architecture (k is the kernel size and d denotes the dilation size) (Figure taken from [9])

The main architecture of the ECAPA model is shown above, in Figure 4.2. The first layer is comprised of a 1D convolutional layer, combined with a Rectified Linear Unit (ReLU) activation and batch normalization (BN) in order to introduce non-linearity and ensure fast convergence and proper regularization for stabilized training. This layer transforms the 80-dimensional input (80 Mel bins) to C channels using a kernel of size 5, and it is followed by three SE-Res2Blocks.

These blocks, whose configuration is shown in Figure 4.3, incorporate Squeeze-Excitation (SE) residual bottleneck blocks, based on the Res2Net architecture [94]. The first component of an SE-Res2Block is a dense layer with a context of 1 frame, which is used for dimensionality reduction, followed by a Res2 dilated convolutional layer, which enhances the central convolutional layer, allowing for multi-scale feature processing. Another dense layer

is used to restore the original dimension of the features, and an SE-block is used to scale each channel. The whole unit is covered by a skip connection. The residual connection in each SE- Res2Block is defined as the sum of all previous blocks, rather than their concatenation. This choice is particularly important for this module, since it significantly reduces the model parameter count.

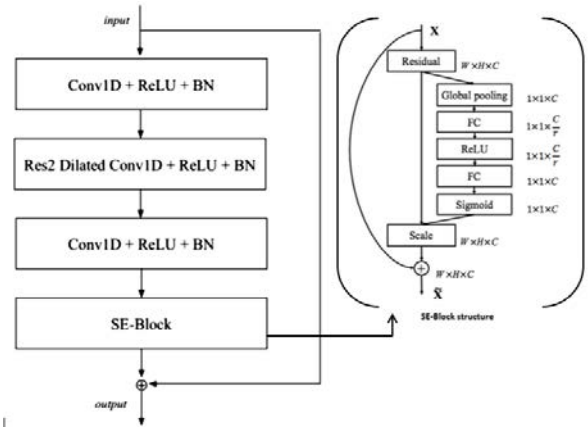


Figure 4.3: The SE-Res2Block of the ECAPA-TDNN architecture. The standard Conv1D layers have a kernel size of 1. The central Res2Net Conv1D with scale dimension s = 8 expands the temporal context through kernel size k and dilation spacing d (Figure adapted from [9] and [95])

Each of the SE-Res2Blocks has an input of C×T dimensionality and utilizes a kernel size of 3, while their dilation rates progressively increase from 2, to 3, and 4, with a scale of 8, to capture multi-scale temporal patterns. The residual connections from these blocks are concatenated to create a combined representation with a dimension of 3×C×T. After this multi-layer feature aggregation (MFA), this representation is projected into 1536 channels, by going through another 1D convolutional layer with a kernel size of 1, before being fed to the attentive statistics pooling layer.

The attentive statistics pooling process, depicted in Figure 4.4, begins with concatenating the MFA features, their mean, and their standard deviation, creating a (4608 × T)-dimensional input that provides the necessary global context. This is passed through the attention mechanism, which is comprised of a 1D convolution with ReLU activation and BN, followed by a Tanh activation and another convolution with a softmax activation. The attention weights generated by the softmax layer are used to calculate the weighted mean and standard deviation vectors that are concatenated to create a 3072-dimensional feature representation.

The weighted mean vector $\tilde{\mu} = [\tilde{\mu}_1, \tilde{\mu}_2, \ldots, \tilde{\mu}_c]$ is calculated by estimating the channel component $\tilde{\mu}_c$ of each utterance as :

$$\tilde{\mu}_c = \sum_{t=1}^{T} a_{t,c} h_{t,c}$$

where $a_{t,c}$ are the attention weights and $h_{t,c}$ the activations of the last frame layer at time step *t* and the weighted standard deviation vector $\tilde{\sigma} = [\tilde{\sigma}_1, \tilde{\sigma}_2, \ldots, \tilde{\sigma}_c]$ is constructed as follows:

$$\tilde{\sigma}_c = \sqrt{\sum_{t=1}^{T} a_{t,c} h_{t,c}^2 - \tilde{\mu}_c^2}$$

where $\tilde{\sigma}_c$ is the channel component.

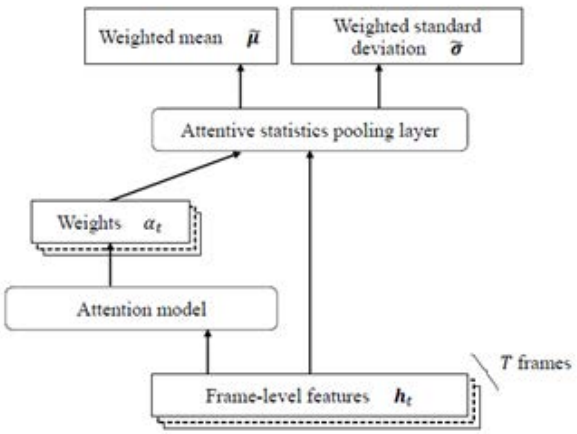


Figure 4.4: Attentive statistics pooling (Figure taken from [96])

Finally, the 3072-dimensional embedding passes through a BN layer, followed by a fully-connected (FC) layer that reduces the dimensionality from 3072 to 192, compressing it into a lower-dimensional space while preserving speaker-specific characteristics. Another BN layer is applied to further refine the embedding, ensuring consistency across different samples. The final output is a 192-dimensional, highly discriminative speaker embedding, ready to be used for the final speaker classification.

4.1.2 Hyperparameters and Training

As shown in Figure 4.2, the ECAPA-TDNN model was trained with Additive Angular Margin Softmax loss (AAM Softmax), which enhances inter-class separability by applying an angular margin to the standard Softmax loss function. The updated loss formula, as introduced in [11] is :

$$\text{Loss} = -\log \frac{e^{s\cos(\theta_{y_i}+m)}}{e^{s\cos(\theta_{y_i}+m)} + \sum_{\substack{j=1 \\ j\neq y_i}}^{N} e^{s\cos(\theta_j)}}$$

where:

θ_j is the angle between the feature vector and the weight vector of class j

y_i is the true class label for sample i

m is the angular margin

s is the scaling factor, controlling the magnitude of logits before Softmax

N is the batch size

The loss, shown in Figure 4.5, was computed with margin m=0.2 and scale factor s=30. The Adam optimizer was used to optimize the model, with an initial learning rate of 0.001 and a weight decay of 2e-5, after a comparative trial with the Stochastic Gradient Descent (SGD) optimizer, and a step learning rate scheduler (StepLR) with 0.97 learning rate decay was used to update the learning rate after every epoch. Finally, the model was trained for a default of 20 epochs, with a batch size of 200, and was evaluated after every epoch.

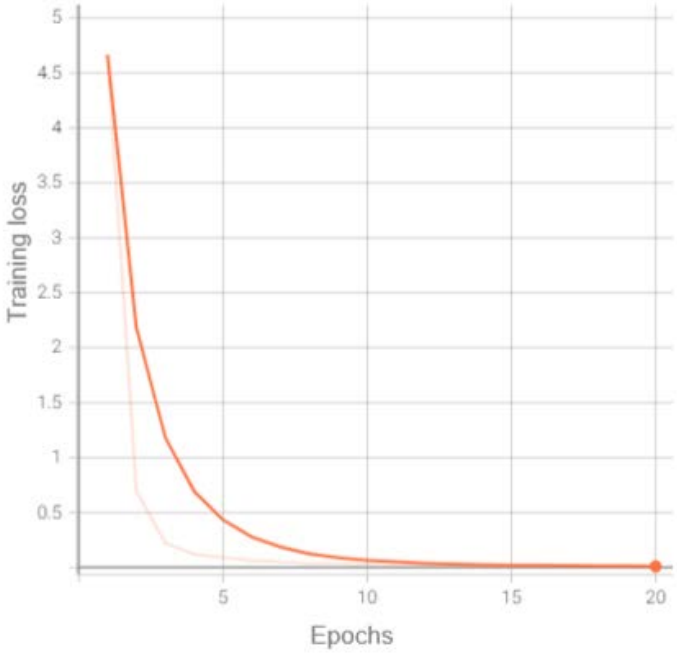


Figure 4.5: Training loss vs. epochs for the ECAPA-TDNN model (The bold line shows the loss with a 0.6 smoothing factor)

4.2 Face Stream

For the face stream we used a PyTorch implementation of the ArcFace system [97], [98], with a MobileFaceNet as the backbone and an ArcFace head for classification. MobileFaceNet is a CNN model, which is a lightweight alternative for the most commonly used ResNet architectures. The high computational cost and the complexity of ResNet models prevents us from using them in this implementation, since the amount of data and the computational power we have at our disposal is limited. As described in [10], MobileFaceNets are MobileNet models optimized for facial recognition. In particular, they are based on the MobileNetV2 architecture, which uses depthwise separable convolutions and inverted residual blocks to reduce computational complexity, while maintaining high accuracy. The use of Parametric Rectified Linear Unit (PReLU) activation instead of ReLU further helps in maintaining a low computational overhead and is shown to improve performance in face verification tasks. Thus, the small trade-off in efficiency and computational cost prompted us to use the MobileFaceNet model for the implementation of this branch of our speaker recognition system. The ArcFace head essentially incorporates an angular margin penalty into the face embeddings produced by the MobileFaceNet, enhancing their discriminative power.

4.2.1 MobileFaceNet Model Architecture

The main component and most important part of a MobileFaceNet is the inverted residual block, which is primarily used in MobileNets. The architecture begins with an initial convolutional block, which applies a 3×3 2D convolution with a stride of 2, followed by BN and PReLU activation. This is followed by a depth-wise convolution that operates on 64 channels, maintaining the spatial dimensions while refining feature representation.

A depth-wise separable convolution module with 64 input and output channels is then applied, with a 3×3 kernel, a stride of 2, and padding of 1, with 128 groups for grouped convolution. This reduces spatial dimensions while learning localized features. We refer to it as a Depth-Wise block and, as shown in Figure 4.6, it consists of a point-wise convolution, a depth-wise convolution, and a linear projection layer.

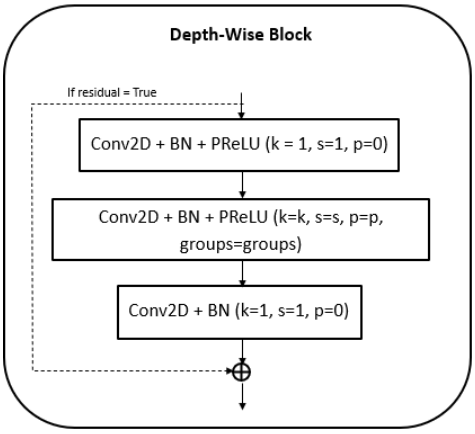


Figure 4.6: Depth-wise separable convolution module (k: kernel size, p: padding, s: stride and groups: number of channel groups for grouped convolution, where groups=[number of input channels] for depth-wise convolution) (If residual=True, the Depth-Wise block is used as part of a Residual block, so a residual connection is added at the final output as seen in Figure 4.7)

Right after the depth-wise separable convolution, a Residual block, which stacks 4 Depth-Wise blocks, each with 64 channels, is applied. A Residual block essentially consists of multiple Depth-Wise modules, connected through their residual connections, to learn hierarchical features. Its structure is shown in Figure 4.7.

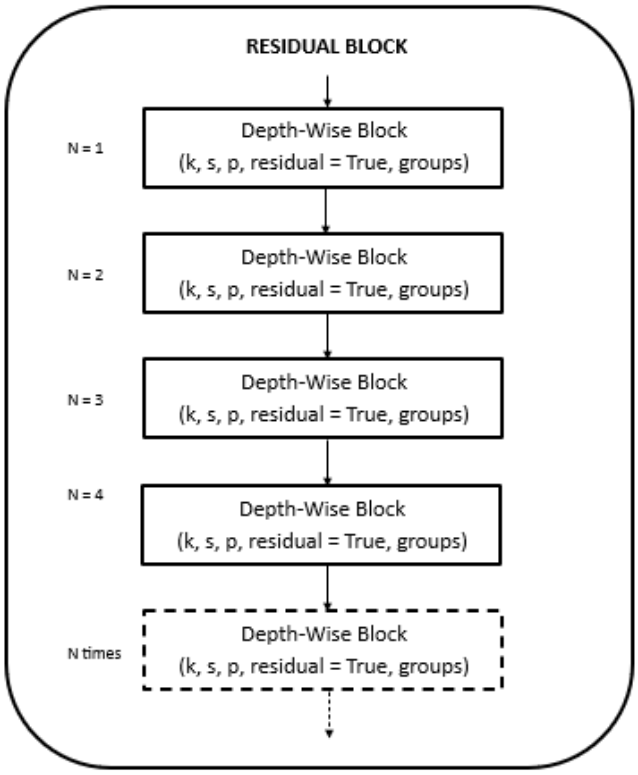


Figure 4.7: Residual block structure

The pattern of Depth-Wise block followed by a Residual block is repeated 2 more times, with 6 and 2 Depth-Wise blocks respectively, progressively increasing the number of channels from 64 to 512, as shown in Figure 4.8, thus reducing spatial resolution through strided convolutions.

Finally, a 1×1 2D convolution expands the feature depth to 512 channels, followed by a global depth-wise convolution with a 7×7 kernel, replacing the global average pooling layer used in MobileNetV2. The global depth-wise convolution treats units of the feature map with different importance, better integrating spatial information. The output of that layer is then flattened and passed through an FC layer, reducing it to the desired embedding size of 512. A BN layer ensures stability before applying L2 normalization to generate the final face embedding and ensure unit-length feature vectors.

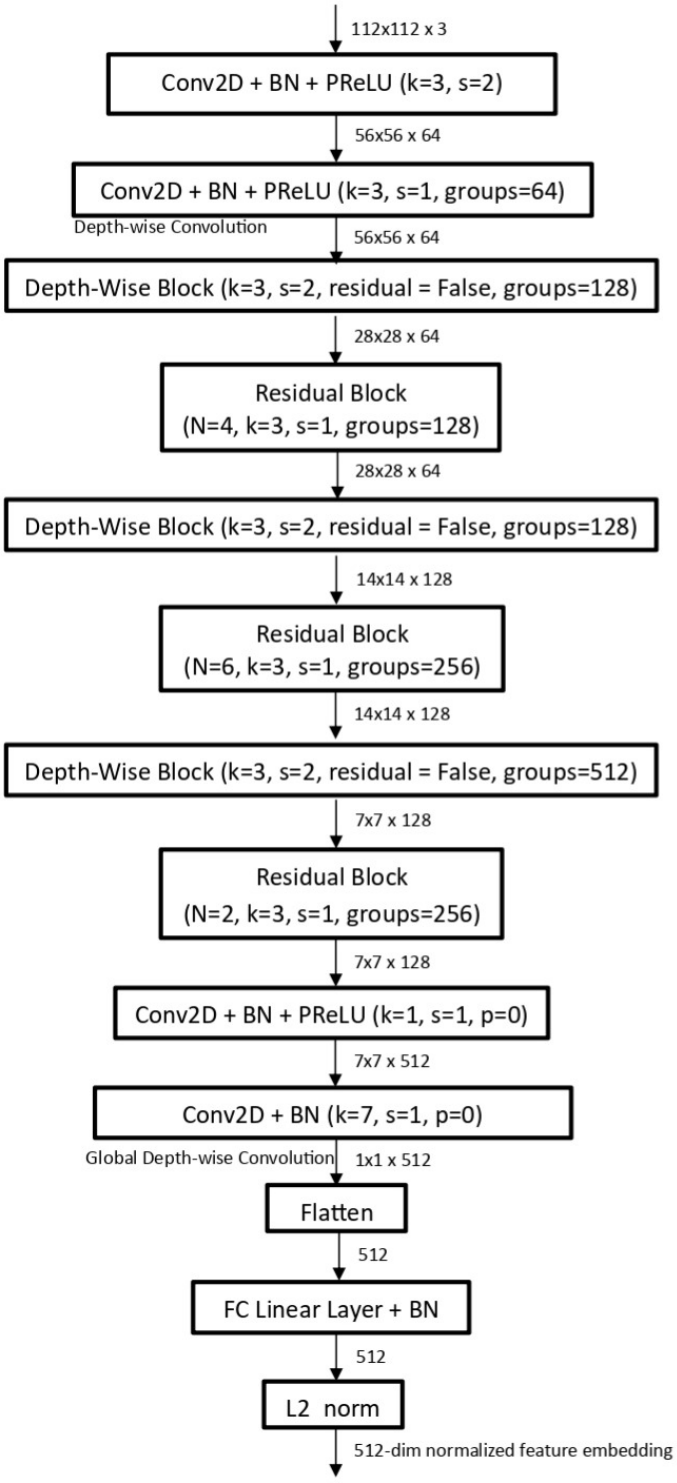


Figure 4.8: MobileFaceNet architecture

The extracted feature vectors are fed to the ArcFace head, pictured in Figure 4.9, which learns the 54 class centers during training and uses them to output adjusted feature logits. Firstly, a learnable weight matrix with dimensions 512×54, where each column represents a class-specific weight vector is randomly initialized and normalized. The cosine of the angle

θ between the embedding and the class vectors is calculated with a matrix multiplication between the embeddings and the normalized kernel (weight matrix), and an angular margin m=0.5 is then added to the cosine values with the use of the trigonometric identity:

$$\cos(\theta + m) = \cos\theta \cdot \cos m - \sin\theta \cdot \sin m$$

The adjusted cosine values are then scaled by the factor s=30 and the final logits are produced.

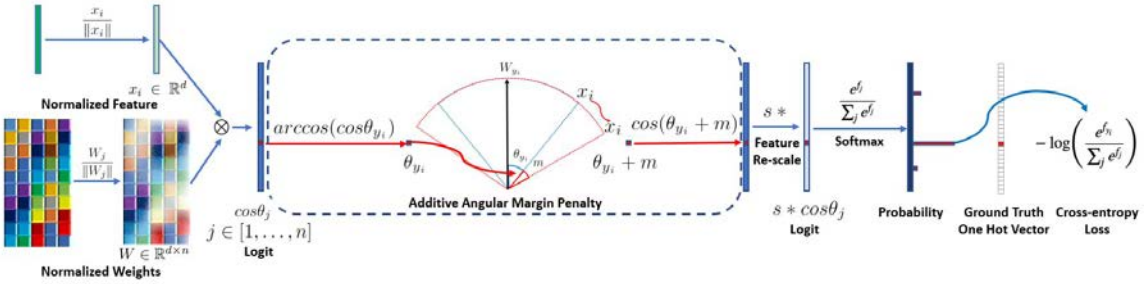


Figure 4.9: ArcFace Head architecture (Figure taken from [11])

4.2.2 Hyperparameters and Training

The training of the face model was performed using cross-entropy loss:

$$L = -\frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{N} y_{ij}\log\hat{y}_{ij}$$

where M is the batch size

N is the number of classes

y_{ij} is a one-hot encoded label (1 if sample i belongs to class j, 0 otherwise)

$\hat{y}_{ij}$ is the predicted probability for class j, obtained via Softmax:

$$\hat{y}_{ij} = \frac{e^{f_j}}{\sum_k e^{f_k}}$$

where f_j are the ArcFace logits.

An SGD optimizer was used to update the model parameters during training, with an initial learning rate of 0.0005 and a momentum of 0.9 to accelerate convergence.

We had to separate BN from the rest of the model's parameters, since they are not learned through backpropagation, and applying weight decay to them could degrade the performance. The optimizer, therefore, applied a 4e - 5 weight decay on all the non-BN parameters except for the last one, which, along with the ArcFace kernel, had a weight decay of 4e - 4.

The model was trained for a default of 20 epochs, with a batch size of 128 and was evaluated every epoch using 10-fold cross-validation. A custom scheduler was used to reduce the learning rate by a factor of 10 at epochs 12, 15, and 18. The training loss can be seen in Figure 4.10 for both views.

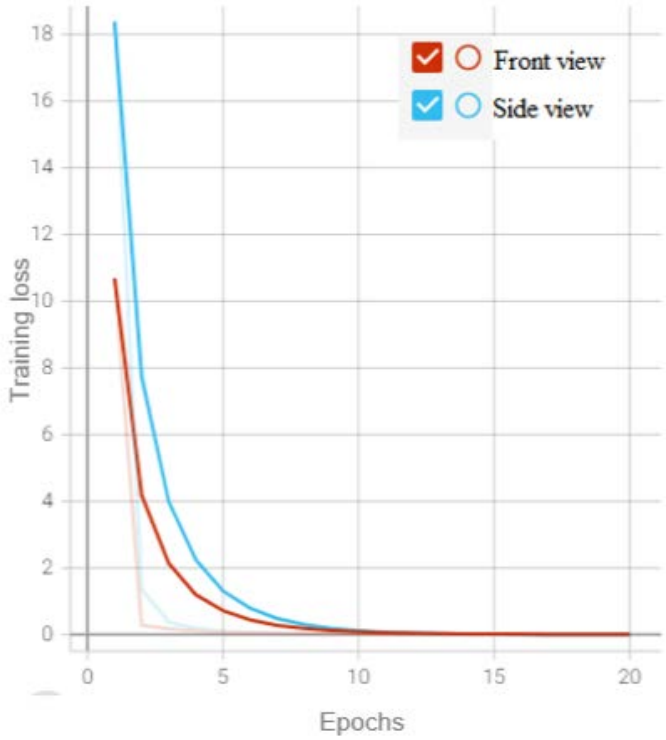


Figure 4.10: Training loss vs. εpochs for the MobileFaceNet model (The bold line shows the loss with a 0.6 smoothing factor)

4.3 Lip Stream

An MCNN was used in combination with an attentive statistics pooling decoder to create the lip model and extract 192-dimensional feature embeddings. MCNNs were specifically designed to classify time series [12], which makes them ideal for modeling lip sequence variations in shape, motion, and articulation over time. They are made up of multiple convolutional layers, each operating at a different scale in the time and frequency domains, effectively learning both fine-grained temporal details and broader spatial features. In order to increase the discriminative power of these features, we incorporated an attentive statistics pooling layer. This attention mechanism assigned greater weight to different temporal and spatial characteristics of the extracted features, making sure the most important aspects of the lip sequence

were emphasized. So overall, the fact that an MCNN takes into account that time series often have features at different time scales makes it a very reliable and robust lip model, especially in combination with an efficient attention mechanism like attentive statistics pooling.

4.3.1 MCNN Model Architecture

First, the grayscale lip sequences, with dimensions B×C×T×H×W, where B the batch size, C the channels, T the video frames, H the height and W the width, are fed to a front-end 3D convolutional layer, whose structure can be seen in Figure 4.11, to extract a 64-channel feature map. This front-end manages to extract spatiotemporal information from the video by stacking a 3D convolution with a 5×9×9 kernel, a 1×5×5 stride, and a 2×3×3 padding, a BN layer, a PreLU activation and a final 3D max-pooling layer with a 1×3×3 kernel, a 1×2×2 stride, and a 0×1×1 padding, reducing the spatial dimensions while preserving temporal resolution. A dropout mechanism with a probability of 0.5 is applied directly after to prevent overfitting. This representation is then reshaped by merging the batch and frame dimensions into a 2D output with shape (B×T)×64×10×10.

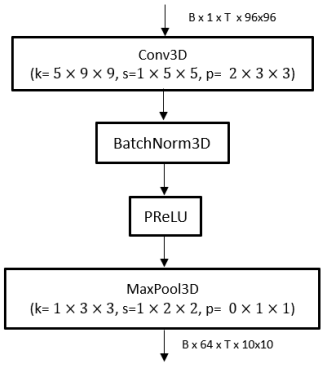


Figure 4.11: Front-end 3D convolutional module

The reshaped feature map is used as input to a ResNet10 backbone, which generates 256-channel embeddings. The 10-layer architecture is chosen in place of the 18 layers used in [15] due to memory constraints and is shown in Figure 4.12. The embeddings then pass through another dropout layer with 0.5 probability before being passed through a global average pooling layer to further reduce the spatial dimensions to (B×T)×256×1×1.

Finally, the feature map is again reshaped into B×T×256 before being fed to a TCN to model the temporal dependecies of the lip sequence. The TCN is made up of two temporal convolutional layers, each with a hidden size of 256 and a kernel size of 5. A dropout rate

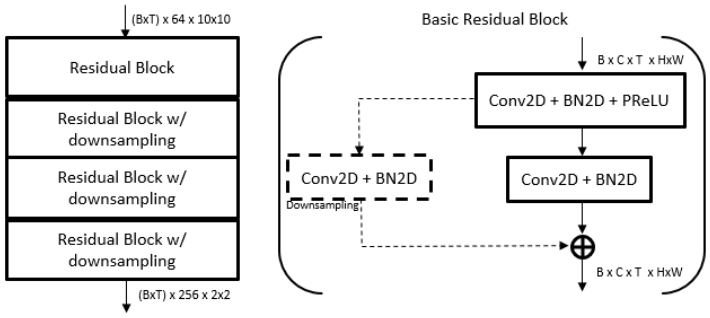


Figure 4.12: ResNet10 architecture and basic Residual Block structure

of 0.1 is applied at this stage to further enhance generalization. Its structure is shown in Figure 4.13.

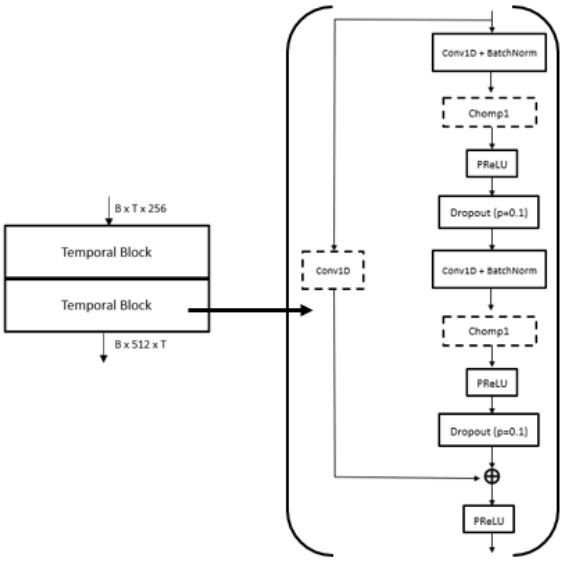


Figure 4.13: TCN and temporal block structure

The exracted features tensor, which now contains both spatial and temporal information, is passed to a final Attentive Statistics Pooling decoder which generates the final 192-dimensional lip embeddings. The attention mechanism has already been described in Subsection 4.1.1 and is shown in Figure 4.4. It should be noted that we do not leverage the global context in this implementation. The entire pipeline of the lip model is depicted in Figure 4.14.

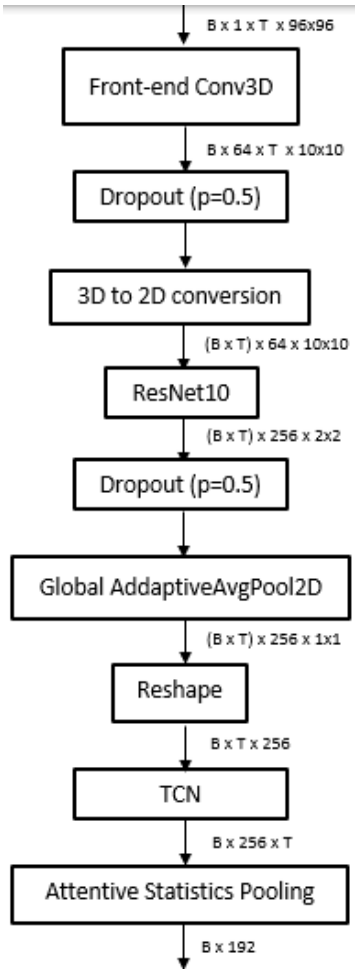


Figure 4.14: Lip Model architecture (MCNN + Attentive Statistics Pooling)

4.3.2 Hyperparameters and Training

We trained the lip model in a similar manner to the audio model, using the AAM Softmax, as described in Subsection 4.1.2. For the calculation of the loss, shown in Figure 4.15, we used a margin m=0.2 along with a scaling factor s=30.

The SGD optimizer was preferred for updating the model parameters during training, with an initial learning rate of 0.0001, a weight decay of 0.0001, and a 0.9 momentum. Nesterov acceleration was also utilized to improve convergence.

The model was trained for 20 epochs, with a 96 batch size, and it was evaluated using 10-fold cross-validation. The learning rate was updated by a StepLR scheduler after every epoch with a decaying factor of 0.97.

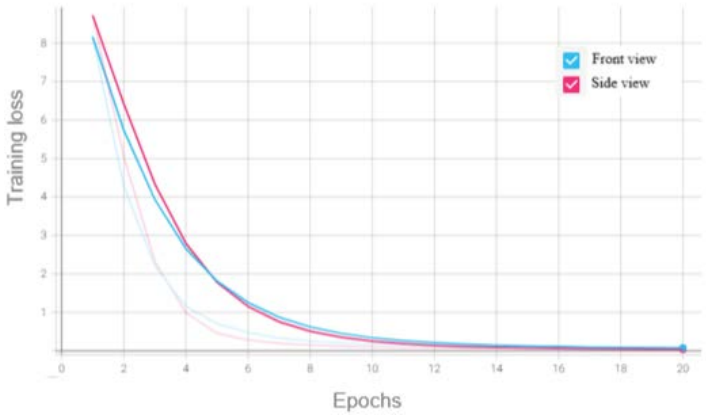


Figure 4.15: Training loss vs. epochs for the MCNN model (The bold line shows the loss with a 0.6 smoothing factor)

4.4 Evaluation Protocol and Performance Metrics

The evaluation process for the proposed audio-visual speaker verification system follows the same overall structure across all three modalities. The pipeline begins with loading the trial pairs created at the dataset splitting stage described in Section 3.2, so the feature embeddings can be extracted.

In the case of the audio and lip streams, the extraction of the embeddings is done in the same way. A global embedding is extracted from the entire sequence in order to identify long range patterns, while a local embedding is extracted from short overlapping consecutive segments of the sequence in order to capture short-term dependencies.

The face stream does not require this kind of handling, meaning one embedding corresponds to one face.

In order to determine whether or not the two speakers of a pair are the same person, we calculate the cosine similarity of their embeddings:

$$\cos\theta = \frac{A \cdot B}{\|A\|\|B\|}$$

where $A \cdot B$ is the dot product of the two vector embeddings A and B, and $\|A\|$, $\|B\|$ are the magnitudes of the vectors.

The scores computed for the global and local embeddings of the audio and lip data are averaged in order to obtain the final score for each sequence. This makes the evaluation more robust by incorporating both long-term and short-term characteristics of the data. The resulting cosine scores are stored in score files corresponding to each modality, so they can later be used for their score fusion.

It should be noted that the process is executed independently for the front and side view.

System performance is measured with two main metrics, Equal Error Rate (EER) and the minimum Detection Cost Function (minDCF), which are the standard in speaker verification tasks.

The EER indicates the point where the False Acceptance Rate (FAR) is equal to the False Rejection Rate (FRR). FAR represents the number of wrongly classified negative pairs, meaning an impostor was accepted even though they are not the person they claim to be. FRR respectively represents the number of wrongly classified positive pairs, meaning a person was rejected even though they are who they claim to be. A low FAR is highly important in verification tasks, since they are mainly used in security applications.

The minDCF provides a weighted evaluation of classification performance based on predefined cost parameters. A DCF is calculated by:

$$\text{DCF}(\vartheta) = C_{\text{miss}} \times P_{\text{target}} \times \text{FAR}(\vartheta) + C_{\text{fa}} \times (1 - P_{\text{target}}) \times \text{FRR}(\vartheta)$$

where C_{miss} is the cost of a miss (falsely rejecting a genuine speaker),

P_{target} is the prior probability of a genuine speaker,

C_{fa} is the cost of a false alarm (falsely accepting an impostor), and

ϑ is the threshold at which the FAR and FRR are calculated.

Since DCF is dependent on the decision threshold, the minDCF is obtained by selecting the threshold that minimizes the cost function.

Receiver Operating Characteristic (ROC) curves are also plotted to visualize the trade-off between true positive and false positive rates at various decision thresholds. The area under the ROC curve (AUC-ROC) quantifies the classifier's overall ability to distinguish between genuine and impostor samples, with a higher AUC indicating better discrimination.

4.5 Fusion

The fusion of the independently trained branches was performed at the score level, as shown in Figure 4.16, with the use of simple averaging:

$$y_{\text{pred}} = \frac{\sum_{i=1}^{N} s_i}{N}$$

and weighted averaging:

$$y_{\text{pred}} = \sum_{i=1}^{N} (w_i \times s_i)$$

where s_i are the scores of each branch, N is the number of independent streams, and w_i are the corresponding weights, which sum up to 1.

In order to find the optimal fusion weights, we employed an optimization-based approach that minimizes the EER. To achieve this, we used the Bayesian optimization method [99]. This approach makes sure the weights are automatically tuned, without manual intervention or computationally costly methods such as grid search.

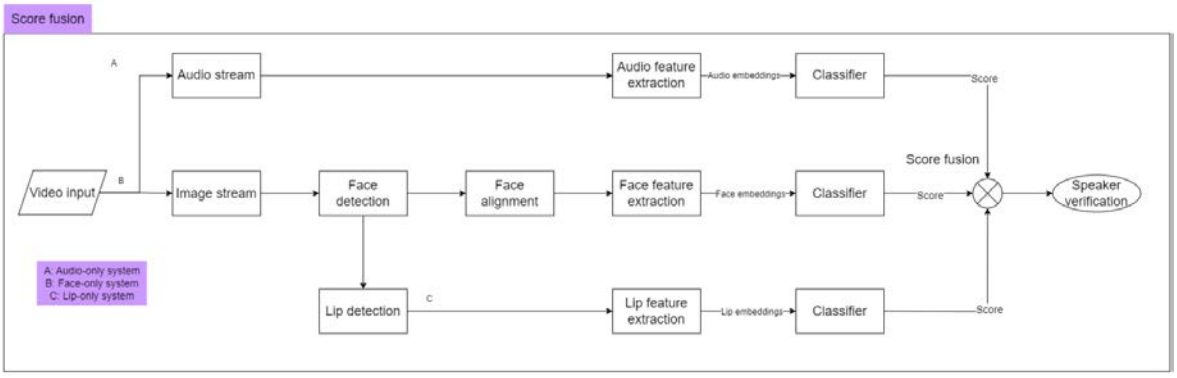


Figure 4.16: Flowchart of score fusion of the three independent modalities

Feature fusion was performed as well, at the evaluation stage, in order to further improve the performance of the face and lip modalities, by mapping the front-view and side embeddings to a common feature space. More specifically, the front-view and side-view features of each speaker of a trial pair were concatenated and L2-normalized, resulting in 1024-dimensional face embeddings, and 384-dimensional lip embeddings respectively. These fused embeddings were then used to calculate the cosine similarity scores between the enrollment and test utterances. A visualization of the feature fusion pipeline is presented in Figure 4.17.

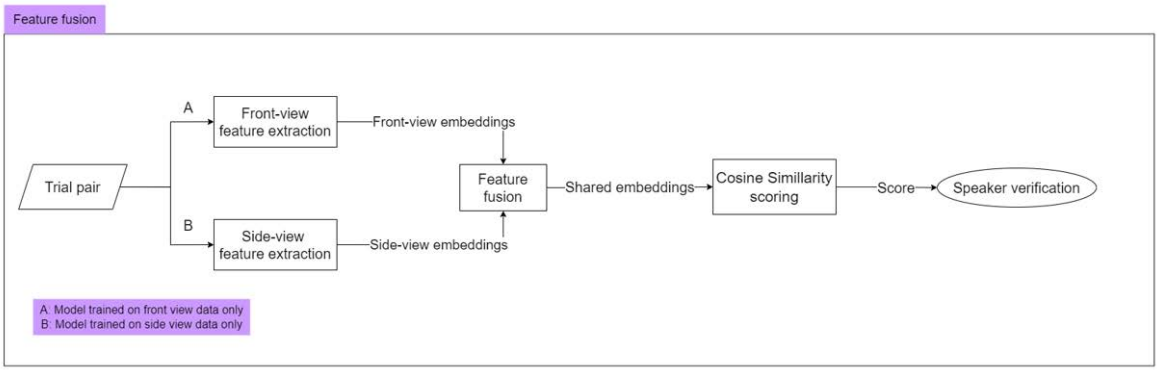


Figure 4.17: Flowchart of the feature fusion of the front and side views of each trial pair

Chapter 5

Experimental Results

In this chapter, we present an evaluation of the three individual modalities in our system in Section 5.1, followed by a performance analysis of their fusion in Section 5.2. The evaluation focuses on three perspectives, front view, side view, and multi-view, with the audio modality remaining consistent across all of them.

5.1 Analysis of Individual Modalities

Starting with the audio modality, the model showed very good performance on the test set, as shown in Table 5.1, with an EER of 0.225% and minDCF of 0.0136, confirming the discriminative power of audio features in verification tasks. The model's performance can also be seen in the ROC curve shown in Figure 5.1.

Table 5.1: Performance metrics of the audio modality

Modality	**EER (%)**	**minDCF**
audio	0.225	0.0136

The face verification models, trained on front view and side view data only, were evaluated on two mixed-view datasets. The first dataset consists of same-view image pairs, while the second one consists of cross-view image pairs.

The performance on the same-view dataset was good for both the front and side models, with respective EERs of 2.450% and 6.025%, as shown in Table 5.2. However, testing on the cross-view pairs showed significantly worse results, reaching an EER of 35.725% for the front view model and 35.575% for the side view one. These results show that, while the

models can correctly verify identities when both images in a pair are from the same view, even if it's a previously unseen view, they do not generalize well across views.

In order to make sure that the model can be accurate regardless of viewing angle, we performed feature fusion of the front-view and side-view embeddings, as descibed in Section 4.5, which led to great performance on the mixed dataset, with an EER of 0.225%. Figure 5.1 shows the ROC curves of each approach.

Table 5.2: Performance of single-view and multi-view face modality systems trained on front and side view data and tested on mixed-view data. Best EER values are highlighted in bold

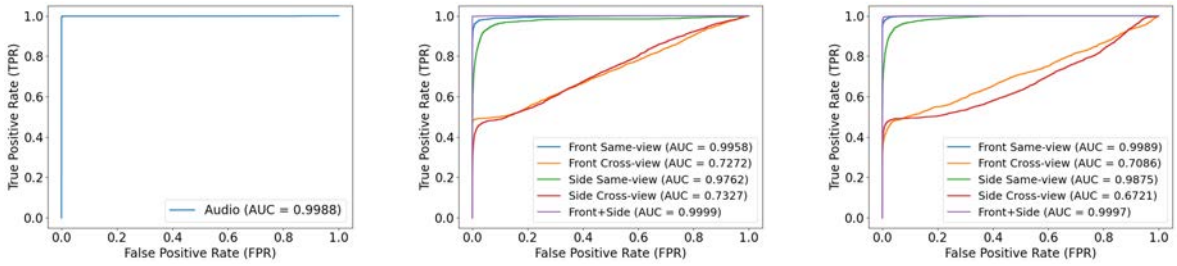
	Front View		**Side View**		**Front+Side View**	
Face	**EER (%)**	**minDCF**	**EER (%)**	**minDCF**	**EER (%)**	**minDCF**
same-view	**2.450**	0.1129	**6.025**	0.3835	**0.225**	0.0002
cross-view	35.725	0.5379	35.575	0.7108	**0.225**	0.0002

The models trained for the lip modality behave in a similar manner. Table 5.3 shows that, when tested on same-view pairs, the front-view model yields an EER of 0.681%, compared to 5.350% observed for the side model. The EER of the front model, tested on the cross-view dataset, rises to 36.668%, and to 41.381% respectively for the side one, indicating that the single-view models' performance is highly dependent on the viewing angle.

Following the same approach used for the face modality, we performed feature fusion of the front and side-view lip embeddings, which drastically improved the verification performance, reaching an EER of 0.575%. Figure 5.1 shows the ROC curves highlighting the performance differences among the lip models.

Table 5.3: Performance of single-view and multi-view lip modality systems trained on front and side view data and tested on mixed-view data. Best EER values are highlighted in bold

	Front View		**Side View**		**Front+Side View**	
Lips	**EER (%)**	**minDCF**	**EER (%)**	**minDCF**	**EER (%)**	**minDCF**
same-view	**0.681**	0.0405	**5.350**	0.3445	**0.575**	0.0018
cross-view	36.668	0.6910	41.381	0.6410	**0.575**	0.0018



(α′) Audio modality (β′) Face modality (γ′) Lip modality

Figure 5.1: ROC curves of the models used for the audio, face, and lip modalities across different view configurations

5.2 Analysis of Fused Modalities

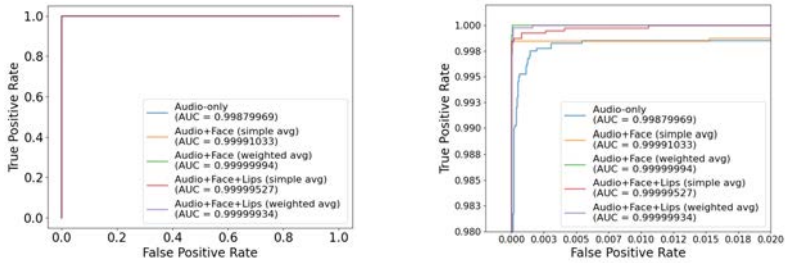
Based on the performance of the uni-modal architectures, we chose the multi-view face and lip systems for the final score fusion of the three individual streams. The score fusion was performed with simple and weighted averaging, where the weights were determined via Bayesian optimization, as described in Section 4.5. The performance metrics of the fused architectures are summarized in Table 5.4, with the corresponding ROC curves shown in Figure 5.2, clearly demonstrating that combining modalities enhances system performance.

Simple averaging fusion already shows better results than the audio-only system. The audio+face configuration reduces the EER from 0.225% to 0.150%, while also significantly lowering minDCF from 0.01360 to 0.00150, confirming that even a basic fusion of complementary modalities can improve verification. The integration of the lip stream produces even better results, achieving an EER of 0.075%, reducing it by 50%, while maintaining the same minDCF value. This indicates that the lip stream contains useful information that can enhance the performance of an already robust verification system.

Table 5.4: Performance metrics of fused systems. Best values are highlighted in bold

Modality	**EER (%)**	**minDCF**
audio-only	0.225	0.01360
audio + face (simple avg)	0.150	0.00150
audio + face (weighted avg)	**0.006**	**0.00100**
audio + face + lips (simple avg)	0.075	0.00150
audio + face + lips (weighted avg)	**0.025**	0.00262

In the case of weighted averaging, we observe even further improvement of the verification results. The audio+face configuration achieves an EER of 0.006% and minDCF of 0.00100, which is the best performance among all tested systems. When the lip stream is added in the pipeline, the EER reaches 0.025% and the minDCF rises to 0.00262. Even though this shows a drop in performance compared to the bi-modal weighted fusion, it still outperforms both systems fused with simple averaging. This highlights the importance of balancing the contribution of each stream during fusion, based on the reliability and discriminative power of the information each modality provides.



(α′) Standard size (β′) Zoomed in

Figure 5.2: ROC curves of the different score fusion configurations

Overall, these findings show that the integration of the lip stream in traditional audio-visual speaker verification systems, is an effective way of enhancing their robustness and accuracy. The proper calibration of each stream's contribution plays a crucial part in getting reliable results as well, especially in scenarios where one modality might be degraded or unavailable.

Chapter 6

Conclusion

6.1 Summary

This thesis studied the combination of audio, face, and lip modalities for speaker verification, aiming to enhance overall performance of standard bimodal audio-visual systems. Independent models were trained and evaluated for each stream before being fused, using simple and weighted averaging, at the score level. The experiments conducted showed that the fusion of the audio, face, and lips modalities significantly enhanced the performance of the system, achieving an EER of 0.025% and minDCF of 0.00262. Overall, this study confirms that lip features offer additional discriminatory power and can enhance accuracy and robustness of audio-visual speaker verification systems.

6.2 Future Work

Several future directions can be explored to further improve the integration of the lip modality in speaker verification systems. The use of a larger dataset in combination with data augmentation and deeper feature extraction architectures, such as Transformer models, may lead to better feature representation and overall improved generalization and robustness. Multiple temporal branches with different kernel sizes could also be utilized in the MCNN model, in order to extract both short-term and long-term temporal information from the lip sequences. Lastly, exploring more fusion strategies, such as feature fusion at the training stage, where speaker embeddings are co-learned, could optimize the contribution of each modality dynamically, based on the quality of the data.

Bibliography

[1] Nilu Singh, Alka Agrawal, and RA Khan. Voice biometric: A technology for voice based authentication. *Advanced Science, Engineering and Medicine*, 10(7-8):754–759, 2018.

[2] Laurynas Dovydaitis, Tomas Rasymas, and Vytautas Rudžionis. Speaker authentication system based on voice biometrics and speech recognition. In *Business Information Systems Workshops: BIS 2016 International Workshops, Leipzig, Germany, July 6-8, 2016, Revised Papers 19*, pages 79–84. Springer, 2017.

[3] Alexa amazon wiki. `https://en.wikipedia.org/wiki/Amazon_Alexa#Alexa_Voice_Service`. Access Date: 08-03-2025.

[4] Amy Neustein and Hemant A Patil. *Forensic speaker recognition*, volume 1. Springer, 2012.

[5] Seyed Omid Sadjadi, Craig Greenberg, Elliot Singer, Lisa Mason, and Douglas Reynolds. The 2021 NIST speaker recognition evaluation. *arXiv preprint arXiv:2204.10242*, 2022.

[6] David Snyder, Daniel Garcia-Romero, Gregory Sell, Alan McCree, Daniel Povey, and Sanjeev Khudanpur. Speaker recognition for multi-speaker conversations using x-vectors. In *ICASSP 2019-2019 IEEE International conference on acoustics, speech and signal processing (ICASSP)*, pages 5796–5800. IEEE, 2019.

[7] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, Inception-ResNet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[8] Najwa Alghamdi, Steve Maddock, Ricard Marxer, Jon Barker, and Guy J Brown. A corpus of audio-visual Lombard speech with frontal and profile views. *The Journal of the Acoustical Society of America*, 143(6):EL523–EL529, 2018.

[9] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *arXiv preprint arXiv:2005.07143*, 2020.

[10] Sheng Chen, Yang Liu, Xiang Gao, and Zhen Han. MobileFaceNets: Efficient CNNs for accurate real-time face verification on mobile devices. *arXiv preprint arXiv:1804.07573*, 2018.

[11] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019.

[12] Zhicheng Cui, Wenlin Chen, and Yixin Chen. Multi-scale convolutional neural networks for time series classification. *arXiv preprint arXiv:1603.06995*, 2016.

[13] Jesús Villalba, Bengt J Borgstrom, Saurabh Kataria, Magdalena Rybicka, Carlos D Castillo, Jaejin Cho, L Paola García-Perera, Pedro A Torres-Carrasquillo, and Najim Dehak. Advances in cross-lingual and cross-source audio-visual speaker recognition: The JHU-MIT system for NIST SRE21. In *Odyssey*, pages 213–220, 2022.

[14] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[15] Meng Liu, Kong Aik Lee, Longbiao Wang, Hanyi Zhang, Chang Zeng, and Jianwu Dang. Cross-modal audio-visual co-learning for text-independent speaker verification. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[16] Meng Liu, Longbiao Wang, Kong Aik Lee, Hanyi Zhang, Chang Zeng, and Jianwu Dang. Deeplip: A benchmark for deep learning-based audio-visual lip biometrics. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 122–129. IEEE, 2021.

[17] Irwin Pollack, James M Pickett, and William H Sumby. On the identification of speakers by voice. *The Journal of the Acoustical Society of America*, 26(3):403–406, 1954.

[18] Sandra Pruzansky. Pattern-matching procedure for automatic talker recognition. *The Journal of the Acoustical Society of America*, 35(3):354–358, 1963.

[19] Wm A Hargreaves and John A Starkweather. Recognition of speaker identity. *Language and Speech*, 6(2):63–67, 1963.

[20] Bishnu S Atal. Automatic speaker recognition based on pitch contours. *The Journal of the Acoustical Society of America*, 52(6B):1687–1697, 1972.

[21] PD Bricker, R Gnanadesikan, MV Mathews, STPA Pruzansky, PA Tukey, KW Wachter, and JL Warner. Statistical techniques for talker identification. *Bell System Technical Journal*, 50(4):1427–1454, 1971.

[22] James E Luck. Automatic speaker verification using cepstral measurements. *The Journal of the Acoustical Society of America*, 46(4B):1026–1032, 1969.

[23] Thomas F Quatieri, R Bob Dunn, and Douglas A Reynolds. On the influence of rate, pitch, and spectrum on automatic speaker recognition performance. In *Proc. ICSLP 2000*, pages vol–2, 2000.

[24] John S Bridle and Michael D Brown. An experimental automatic word recognition system. *JSRU report*, 1003(5):33, 1974.

[25] Ronald W Schafer and Lawrence R Rabiner. Digital representations of speech signals. *Proceedings of the IEEE*, 63(4):662–677, 1975.

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

[27] Lawrence Rabiner and Biinghwang Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, 3(1):4–16, 1986.

[28] Naftali Z Tisby. On the application of mixture AR hidden Markov models to text independent speaker recognition. *IEEE Transactions on Signal Processing*, 39(3):563–570, 1991.

[29] Richard C Rose and Douglas A Reynolds. Text independent speaker identification using automatic acoustic segmentation. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 293–296. IEEE, 1990.

[30] Douglas Alan Reynolds. *A Gaussian mixture modeling approach to text-independent speaker identification*. PhD thesis, Georgia Institute of Technology, 1992.

[31] Douglas A Reynolds and Richard C Rose. Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing*, 3(1):72–83, 1995.

[32] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing*, 10(1-3):19–41, 2000.

[33] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2010.

[34] David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur. Deep neural network embeddings for text-independent speaker verification. In *Interspeech*, pages 999–1003, 2017.

[35] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust DNN embeddings for speaker recognition. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5329–5333. IEEE, 2018.

[36] Daniel Garcia-Romero, David Snyder, Gregory Sell, Alan McCree, Daniel Povey, and Sanjeev Khudanpur. X-vector DNN refinement with full-length recordings for speaker recognition. In *Interspeech*, pages 1493–1496, 2019.

[37] Hossein Zeinali, Lukas Burget, and Jan Cernocky. Convolutional neural networks and x-vector embedding for DCASE2018 acoustic scene classification challenge. *arXiv preprint arXiv:1810.04273*, 2018.

[38] Qihang Xu, Mingjiang Wang, Changlai Xu, and Lu Xu. Speaker recognition based on long short-term memory networks. In *2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP)*, pages 318–322. IEEE, 2020.

[39] Jahangir Alam, Abderrahim Fathan, and Woo Hyun Kang. Text-independent speaker verification employing CNN-LSTM-TDNN hybrid networks. In *International Conference on Speech and Computer*, pages 1–13. Springer, 2021.

[40] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

[41] Ismail Shahin, Ali Bou Nassif, Nawel Nemmour, Ashraf Elnagar, Adi Alhudhaif, and Kemal Polat. Novel hybrid DNN approaches for speaker verification in emotional and stressful talking environments. *Neural Computing and Applications*, 33(23):16033–16055, 2021.

[42] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer. End-to-end text-dependent speaker verification. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5115–5119. IEEE, 2016.

[43] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI'81: 7th international joint conference on Artificial intelligence*, volume 2, pages 674–679, 1981.

[44] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[45] Juergen Luettin, Neil A Thacker, and Steve W Beet. Speaker identification by lipreading. In *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96*, volume 1, pages 62–65. IEEE, 1996.

[46] Lourdes Ramirez Cerna, Guillermo Camara-Chavez, and D Menotti. Face detection: Histogram of oriented gradients and bag of feature method. In *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition*

(IPCV), page 1. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp)., 2013.

[47] Md Abdur Rahim, Md Najmul Hossain, Tanzillah Wahid, and Md Shafiul Azam. Face recognition using local binary patterns (LBP). *Global Journal of Computer Science and Technology*, 13(4):1–8, 2013.

[48] Chengjun Liu and Harry Wechsler. A Gabor feature classifier for face recognition. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 270–275. IEEE, 2001.

[49] David Dean and Sridha Sridharan. Dynamic visual features for audio-visual speaker verification. *Computer Speech & Language*, 24(2):136–149, 2010.

[50] Ralph Gross, Jie Yang, and Alex Waibel. Growing Gaussian mixture models for pose invariant face recognition. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 1, pages 1088–1091. IEEE, 2000.

[51] William M. Campbell, Joseph P. Campbell, Terry P. Gleason, Douglas A. Reynolds, and Wade Shen. Speaker verification using support vector machines and high-level features. *IEEE Transactions on Audio, Speech, and Language Processing*, 2007.

[52] Abhishek Bansal, Kapil Mehta, and Sahil Arora. Face Recognition Using PCA and LDA Algorithm. In *2012 Second International Conference on Advanced Computing & Communication Technologies*, pages 251–254, 2012.

[53] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[54] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[55] Ruijie Tao, Rohan Kumar Das, and Haizhou Li. Audio-visual speaker recognition with a cross-modal discriminative network. *arXiv preprint arXiv:2008.03894*, 2020.

[56] Omkar Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *BMVC 2015-Proceedings of the British Machine Vision Conference 2015*. British Machine Vision Association, 2015.

[57] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[58] Vijay Kumar B G, Gustavo Carneiro, and Ian Reid. Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[59] Juergen Luettin, Neil A Thacker, and Steve W Beet. Active shape models for visual speech feature extraction. *Speechreading by Humans and Machines: Models, Systems, and Applications*, pages 383–390, 1996.

[60] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.

[61] Timothy F. Cootes, Gareth J. Edwards, and Christopher J Taylor. Active appearance models. *IEEE Transactions on pattern analysis and machine intelligence*, 23(6):681–685, 2001.

[62] Sung Joo Lee, Kang Ryoung Park, and Jaihie Kim. A comparative study of facial appearance modeling methods for active appearance models. *Pattern Recognition Letters*, 30(14):1335–1346, 2009.

[63] Satoshi Tamura, Koji Iwano, and Sadaoki Furui. Multi-modal speech recognition using optical-flow analysis for lip images. *Journal of VLSI signal processing systems for signal, image and video technology*, 36:117–124, 2004.

[64] Christoph Bregler and Yochai Konig. ”Eigenlips” for robust speech recognition. In *Proceedings of ICASSP’94. IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2. IEEE, 1994.

[65] G. Potamianos, H.P. Graf, and E. Cosatto. An image transform approach for HMM based automatic lipreading. In *Proceedings 1998 International Conference on Image Processing. ICIP98*, 1998.

[66] N Puviarasan and Sengottayan Palanivel. Lip reading of hearing impaired persons using HMM. *Expert Systems with Applications*, 38(4):4477–4481, 2011.

[67] Iain Matthews, Gerasimos Potamianos, Chalapathy Neti, and Juergen Luettin. A comparison of model and transform-based visual features for audio-visual LVCSR. In *IEEE International Conference on Multimedia and Expo, 2001. ICME 2001.*, pages 825–828. IEEE Computer Society, 2001.

[68] H Ertan Cetingul, Yücel Yemez, Engin Erzin, and A Murat Tekalp. Discriminative analysis of lip motion features for speaker identification and speech-reading. *IEEE Transactions on Image Processing*, 15(10):2879–2891, 2006.

[69] Ayaz A Shaikh, Dinesh K Kumar, Wai C Yau, MZ Che Azemin, and Jayavardhana Gubbi. Lip reading using optical flow and support vector machines. In *2010 3rd international congress on image and signal processing*, volume 1, pages 327–330. IEEE, 2010.

[70] Alex Sherstinsky. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena*, 404, 2020.

[71] Patel Dhruv and Subham Naskar. Image classification using convolutional neural network (CNN) and recurrent neural network (RNN): A review. *Machine learning and information processing: proceedings of ICMLIP 2019*, pages 367–381, 2020.

[72] Amit Garg, Jonathan Noyola, and Sameep Bagadia. Lip reading using CNN and LSTM. *Technical report, Stanford University, CS231 n project report*, 2016.

[73] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[74] Michael Wand, Jan Koutník, and Jürgen Schmidhuber. Lipreading with long short-term memory. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6115–6119. IEEE, 2016.

[75] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[76] Yannis M Assael, Brendan Shillingford, Shimon Whiteson, and Nando De Freitas. LipNet: End-to-end sentence-level lipreading. *arXiv preprint arXiv:1611.01599*, 2016.

[77] Stavros Petridis, Themos Stafylakis, Pingehuan Ma, Feipeng Cai, Georgios Tzimiropoulos, and Maja Pantic. End-to-end audiovisual speech recognition. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 6548–6552. IEEE, 2018.

[78] KR Prajwal, Triantafyllos Afouras, and Andrew Zisserman. Sub-word level lip reading with visual attention. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 5162–5172, 2022.

[79] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[80] Alexandros Koumparoulis and Gerasimos Potamianos. Accurate and resource-efficient lipreading with EfficientNetV2 and Transformers. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8467–8471. IEEE, 2022.

[81] R Gnana Praveen and Jahangir Alam. Audio-visual speaker verification via joint cross-attention. *arXiv preprint arXiv:2309.16569*, 2023.

[82] C. Neti, G. Potamianos, J. Luettin, I. Matthews, H. Glotin, D. Vergyri, J. Sison, A. Mashari, and J. Zhou. Audio-visual speech recognition. Final workshop report, Center for Language and Speech Processing, The Johns Hopkins University, Baltimore, MD, USA, 2000.

[83] Lea Schönherr, Dennis Orth, Martin Heckmann, and Dorothea Kolossa. Environmentally robust audio-visual speaker identification. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pages 312–318. IEEE, 2016.

[84] Chenxi Yu and Lin Huang. Biometric recognition by using audio and visual feature fusion. In *2012 International Conference on System Science and Engineering (ICSSE)*, pages 173–178. IEEE, 2012.

[85] Yufei Wang. Efficient audio-visual speaker recognition via deep multi-modal feature fusion. In *2021 17th International Conference on Computational Intelligence and Security (CIS)*, pages 99–103. IEEE, 2021.

[86] Zhifang Wang, Erfu Wang, Shuangshuang Wang, and Qun Ding. Multimodal biometric system using face-iris fusion feature. *J. Comput.*, 6:931–938, 2011.

[87] Angela Schörgendorfer. Extended confidence-weighted averaging in sensor fusion. Master's thesis, Technische Universität Wien, May 2006.

[88] Andrew Senior, Chalapathy V Neti, and Benoit Maison. On the use of visual information for improving audio-based speaker recognition. In *International Conference on Auditory-Visual Speech Processing*, 1999.

[89] Engin Erzin, Yücel Yemez, and A Murat Tekalp. Multimodal speaker identification using an adaptive classifier cascade based on modality reliability. *IEEE Transactions on Multimedia*, 7(5):840–852, 2005.

[90] Leda Sarı, Kritika Singh, Jiatong Zhou, Lorenzo Torresani, Nayan Singhal, and Yatharth Saraf. A multi-view approach to audio-visual speaker verification. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6194–6198. IEEE, 2021.

[91] G. Potamianos, J. Luettin, and C. Neti. Hierarchical discriminant features for audio-visual LVCSR. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (ICASSP)*, 2001.

[92] Zhiyong Wu, Lianhong Cai, and Helen Meng. Multi-level fusion of audio and visual features for speaker identification. In *Advances in Biometrics: International Conference, ICB 2006, Hong Kong, China, January 5-7, 2006. Proceedings*, pages 493–499. Springer, 2005.

[93] Gross Ralph, Matthews Iain, Cohn Jeffrey, Kanade Takeo, and Baker Simon. Multi-PIE. *Image and vision computing*, 28(5):807–813, 2010.

[94] Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr. Res2Net: A new multi-scale backbone architecture. *IEEE transactions on pattern analysis and machine intelligence*, 43(2):652–662, 2019.

[95] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[96] Koji Okabe, Takafumi Koshinaka, and Koichi Shinoda. Attentive statistics pooling for deep speaker embedding. *arXiv preprint arXiv:1803.10963*, 2018.

[97] DeepInsight. Arcface: Torch implementation. `https://github.com/deepinsight/insightface/tree/master/recognition/arcface_torch`. Accessed: 08-03-2025.

[98] TreB1eN. Insightface: Pytorch implementation. `https://github.com/TreB1eN/InsightFace_Pytorch/tree/master`. Accessed: 08-03-2025.

[99] Peter I Frazier. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.