Master Thesis

Sarai Lopez Quezada

Enrollment number: [redacted]

Analysis of Individual Speaker Features on Group-level Emotion Recognition from Speech

Master of Science Biomedical Engineering
Faculty of Life Sciences
Department of Biomedical Engineering
Hamburg University of Applied Sciences
in cooperation with the Cognitive Systems Lab (CSL) of the University of Bremen

Sarai Lopez Quezada

Enrollment number: [redacted]

Analysis of Individual Speaker Features on Group-level Emotion Recognition from Speech

Master thesis submitted for examination in Master´s degree
in the study course *Master of Science Biomedical Engineering*
at the Department of Biomedical Engineering
at the Faculty of Life Sciences
at Hamburg University of Applied Sciences
in cooperation with the University of Bremen

First examiner:
Prof. Dr. Thomas Schiemann (HAW Hamburg)
Second examiner:
Prof. Dr.-Ing. Tanja Schultz (Universität Bremen)
Supervisors:
Dr.-Ing. Zhao Ren (Universität Bremen)
Marvin Borsdorf, M.Sc. (Universität Bremen)
Rathi Adarshi Rammohan, M.Tech (Universität Bremen)

Submitted on: 15. May 2025

Kurzzusammenfassung

Group Emotion Recognition (GER) ist von entscheidender Bedeutung sowohl für das Verständnis sozialer Dynamiken als auch für das zwischenmenschliche Verhalten und für die Verbesserung von Mensch-Computer-Interaktionen. Group-Emotions umfassen die von den einzelnen Mitgliedern empfundenen Emotionen und die Kontextfaktoren der Gruppe, die den emotionalen Zustand der Gruppe beeinflussen. Systeme für GER, die auf Sprachsignalen basieren, sollten sich daher auf Informationen stützen, die sowohl aus dem Gruppenkontext als auch aus dem individuellen Kontext stammen. Die Verwendung von Features, die aus Bildern von Einzelpersonen extrahiert wurden, in Kombination mit gruppenbasierten Features hat sich als wirksam für die Verbesserung der GER erwiesen. Daher lässt sich annehmen, dass die Einbeziehung individueller Informationen von Sprechenden zur Verbesserung der speech-based GER führen kann. In dieser Arbeit soll untersucht werden, wie verschiedene Arten von akustischen Features (cepstral, spectral, prosodic, temporal und Sound-Quality-Features) der einzelnen Sprechenden zur Emotionserkennung der Gruppe beitragen. Um dies zu erreichen, wird ein vortrainiertes Modell zur Speech-Separation verwendet. Dadurch werden die einzelnen Stimmen der Speech-Mixture isoliert. Anschließend werden Features aus der Speech-Mixture (Mixture-Features) und der Sprache der einzelnen Sprechenden (individual Features) extrahiert. Es werden verschiedene Experimente mit Support Vector Machines (SVMs) und Fully-Connected Neural Networks (FCNNs) für speech-based GER durchgeführt. Die Modelle werden für die Erkennung von drei Group-Emotiuon-Classes, namentlich *Positiv*, *Neutral*, und *Negativ* trainiert und auf speaker-dependant (bekannte Sprechende) und speaker-indipendent (unbekannte Sprechende) Daten getestet. Die verwendeten Sprachdaten bestehen aus Speech-Mixtures, die aus Videos der VGAF-Datenbank gewonnen wurden. Die SVM- und FCNN-Modelle werden separat auf verschiedenen Feature-Sets trainiert, die Mixture- und individual Features kombinieren. Die Ergebnisse zeigen eine signifikante Verbesserung der Leistung des FCNN-Modells in speaker-indipendent Test-Szenarien, wenn das Modell eine Kombination aus spectral-individual Features und Mixture-Features als Trainingsdaten verwendet (Makro-F1-Score = 65.56%), verglichen mit der Verwendung von ausschließlich Mixture-Features (Makro-F1-Score = 53.48%). Sound-Quality-, temporal und cepstral Features der einzelnen Sprechenden zeigen ebenfalls Verbesserungen bei den GER-Werten, die von den Modellen erreicht werden, wenn sie in Kombination mit den Mixture-Features für das Training verwendet werden.

Abstract

Group Emotion Recognition (GER) is crucial for understanding social dynamics and inter-human behavior, enhancing human-computer interactions. Group emotion encompasses the emotions felt by the individual members and group context factors that shape the emotional state of the group. Employing features extracted from images of the individuals in combination with group-based features has been proven to be effective for improving GER. Consequently, there is a potential for involving individual speaker information to improve speech-based GER. This work aims to analyze how different types of acoustic features (cepstral, spectral, prosodic, temporal, and sound quality features) from the individual speakers contribute to the emotion recognition of the group. To achieve this, a pre-trained model for speech separation is employed for isolating the speech of each speaker in the mixture. Then, features are extracted from the speech mixtures (mixture features) and the individuals' speech (individual features). Various experiments are conducted using Support Vector Machines (SVMs) and Fully-Connected Neural Networks (FCNNs) for speech-based GER. The models are trained for recognition of three group emotion classes, namely *Positive*, *Neutral*, and *Negative*. They are tested in speaker-dependent, i.e. known speakers, as well as speaker-independent, i.e. unknown speakers, scenarios. The speech data employed consists of speech mixtures obtained from videos of the VGAF database. The SVM and FCNN models are trained separately on different feature sets that combine mixture and individual features. The results show a significant improvement in the performance of the FCNN model in speaker-independent scenarios when the model uses a combination of spectral-individual features and mixture features as training input (macro F1-score = 65.56%), compared to using only mixture features (macro F1-score = 53.48%). Sound quality, temporal, and cepstral features from the individual speakers also demonstrate improvements in the GER scores achieved by the models when used in combination with the mixture features for training.

Acknowledgments

I deeply thank my supervisors from the University of Bremen, Marvin, Rathi, and Zhao, for guiding me, answering my questions, and introducing me to the world of emotion recognition in speech. Their time, patience, and motivation have been fundamental to my progress. I have profound respect and admiration for them.

Thanks to Prof. Schiemann for helping me to carry this project out and for introducing me to the MLAG team, where they always helped me by answering my questions and sharing their knowledge in machine learning.

This would not have been possible without the invaluable support of my beloved husband, Lucas. Thank you for taking care of me and ensuring my well-being, especially during the most difficult moments. Sharing my life with you is the most wonderful gift life has given me.

To my mother, Claudia, and grandmother, Juanita, thank you for guiding me, motivating me to move forward, and allowing me to dream big. All my achievements are thanks to you both.

Thanks to Biggi and Ralf, my dear parents-in-law, for their support and motivation. Thank you for always being there for me.

Thanks to Oma Schmidt for taking care of me and providing me with a supply of chocolates and snacks for my study days.

Thanks to my relatives and friends who supported me on this path and motivated me to keep going. Thank you for making me feel their presence even from afar.

Contents

List of Figures

List of Tables

List of Acronyms

F_0 Fundamental Frequency. 41

AI Artificial Intelligence. 13

ANNs Artificial Neural Networks. 20

ASR Automatic Speech Recognition. 38, 53

BSS Blind Source Separation. 49

CM Confusion Matrix. 27, 76

CNNs Convolutional Neural Networks. 26

ComParE Computational Paralinguistic ChallengE. 36, 39, 42, 69

DL Deep Learning. 14

DNNs Deep Neural Networks. 23

DNNs Deep Neural Networks. 14

ECG electrocardiogram. 35

EmotiW Emotion Recognition In The Wild Challenge. 36, 46, 61, 69

ER Emotion Recognition. 35, 45, 52

FCNNs Fully Connected Neural Networks. 23

FN False Negatives. 27

FP False Positives. 27

GER Group Emotion Recognition. 10

GNNs Graph Neural Networks. 26

HNR Harmonic-to-noise ratio. 42

HSFs High-level statistical functions. 39

LDA Linear Discriminant Analysis. 52

LLDs Low-level descriptors. 39

LSTM Long Short-Term Memory Neural Networks. 27

MFCC Mel Frequency Cepstral Coefficients. 39

MIT Massachusetts Institute of Technology. 35

ML Machine Learning. 13, 45

MLP Multi Layer Perceptron. 22

NNs Neural Networks. 16, 20

OpenSMILE open-source Speech and Music Interpretation by Large-space Extraction. 43, 69, 70

PCA Principal Component Analysis. 52

RASTA Relative Spectral Transform. 40

RBF Radial Basis Function. 19, 73

ReLu Rectified Linear Unit. 25

RMS Root Mean Square. 41

RNNs Recurrent Neural Networks. 27

SER Speech Emotion Recognition. 10, 36, 42, 43, 57, 69

spk-dep speaker-dependent. 89

spk-ind speaker-independent. 89

STFT Short-Time Fourier Transform. 49

SVMs Support Vector Machines. 16

TN True Negatives. 27

TP True Positives. 27

TSE Target Speaker Extraction. 49, 57

UA Unweighted Accuracy. 57

UAR Unweighted Average Recall. 29, 76

VAD Voice Activity Detection. 51, 62

VGAF Video Group Affect database. 47, 61, 69

WA Weighted Accuracy. 57

1 Introduction

Emotions are a fundamental aspect of human experience, influencing cognition, perception, learning, communication, and decision-making. Humans express emotions through various channels, including gestures, facial expressions, voice, and actions. Extending the understanding of emotions to computers can enhance human-computer interaction.

The voice is a powerful tool for conveying emotions. Studies have shown that emotions like anger and happiness are characterized by high pitch and speech rate, while sadness is marked by lower pitch and slower speech rate [1]. Individuals can alter the way they react and behave —including the way they speak— based on their perception of the emotions of those around them [2]. Speech emotion recognition (SER) is a field of study that focuses on the automatic recognition of human emotions from speech signals. Multiple approaches on SER have been proposed involving conventional statistical-based methods and machine learning-based methods.

While most SER systems focus on individual-level emotion recognition [3, 4, 5], it is valuable to analyze emotions at a group level. Humans have a natural tendency to seek out and thrive in social environments. Thus, analyzing group emotions is beneficial for understanding social dynamics and human behavior [6].

Group emotion recognition (GER) is a research field that focuses on the identification of group emotions by analyzing data. Group emotion encompasses both the emotions felt by individual members and group context factors that shape the emotional state of the group. Most GER systems utilize video and image data for recognizing the group emotion [7, 8, 9, 10, 11, 12, 13], while only a few employ audio data [14, 15, 16, 17]. GER from speech can be considered as SER applied at a group level, focusing on analyzing speech signals to identify the emotion of a group of people [14].

Speech-based GER systems could revolutionize real-life applications beyond the already explored areas of crowd monitoring, surveillance, and student participation analysis. These systems are pivotal in developing voice assistants capable of interacting with multiple individuals simultaneously and adapting their responses based on recognized emotions. Furthermore, GER from speech can be used to enhance video conferences, virtual meetings, and group calls by providing more personalized interactions based on the group emotion of the participants.

GER systems employ *top-down approaches* and *bottom-up approaches* for recognizing the group emotion [18, 19]. In bottom-up approaches, the emotions of individual group members are aggregated to estimate the group's overall emotion. Therefore, the features analyzed are extracted at an individual level. In top-down approaches, the group is treated as a homogeneous entity, so the features analyzed are extracted from a group context rather than an individual context for GER.

Typically, the speech signals analyzed in group-level SER consist of audio data in the form of mono-sound files (one audio channel) containing the mixture of speech sources. This mixture, often referred to as a *speech mixture*, includes the voices of all speakers combined into a single audio signal. The systems that have been developed for GER from speech typically use a top-down approach for recognizing the group emotion, which means that the features analyzed are extracted directly from the speech mixtures. Meanwhile, bottom-up approaches for group-level SER have not yet been given enough relevance. Bottom-up approaches require the extraction of features of the individual speakers. However, this cannot be done straightforwardly when the voices of the speakers are within a speech mixture. It is necessary to isolate the voice of each speaker first to be able to extract features from the individual speakers. Separating different voices from a speech mixture is a challenging task, and there is an entire research field called *speech separation* dedicated to it.

Given that group emotion encompasses both the group context and the emotions experienced by individual members, it is imperative to study speech-based GER using features derived from both the group as a whole and the individual speakers. Therefore, this work integrates the aforementioned top-down and bottom-up approaches into a hybrid approach for GER from speech using acoustic features extracted from both the speech mixture and the individual speakers, facilitated by a speech separation system.

The goal is to analyze how the features of the individual speakers contribute to the recognition of the group emotion. To achieve this, different experiments were performed implementing machine learning models trained on features from the speech mixtures and the individual speakers for GER. The same parameters are calculated from the speech mixtures and the separated speeches. The parameters calculated from the speech mixture (*mixture features*) represent the features extracted at the group context, while the features from the separated speeches represent the individual speaker features, hereafter referred to as *individual features.* Following, the individual features are grouped into five categories. Various feature sets are then created by concatenating the mixture features with each category of individual features. These feature sets are employed to train the two machine learning models for GER. The performance of each model is tested and compared to a baseline, which consists of the same model trained only on mixture features. It is hypothesized that the models trained on both mixture and individual features outperform the baseline. Finally, a significance test is applied to validate the results. This method aims to identify which categories of individual features contribute to the recognition of group emotion and to analyze the influence of each category.

The primary objectives of this thesis are:

- To develop recognition models for group emotion based on mixture features and individual features from real-life speech data.
- To test the performance of the GER models using speech data for unknown and known speakers.
- To identify and analyze the influence of the different groups of acoustic features extracted from individual speakers on the recognition of the group emotion.

The remainder of this thesis is structured as follows: Section 2 introduces the relevant concepts necessary to understand the methodology, including machine learning models, emotion theories, SER, GER, speaker diarization, and speech separation. Section 3 details the methods employed in this study, encompassing the pre-processing of the database, the definition of experiments, and the evaluation of the models' performance. Section 4 presents the results and discussion, providing a thorough analysis of the findings. The conclusion of this work is then articulated in Section 6, summarizing the key insights and contributions of this study.

2 Theoretical background

2.1. Machine learning

In today's rapidly advancing technological landscape, terms like *artificial intelligence* (AI) and *machine learning* (ML) have become widely used and highly relevant. These terms are often used interchangeably, leading to confusion about their distinct roles and definitions. Although they are related, each term refers to a different concept.

AI encompasses all systems that simulate intelligent behavior, including complex algorithms and approaches based on logic or probabilistic reasoning [20]. For example, rule-based expert systems use predefined rules to make decisions or solve problems.

ML is a subfield of AI that has seen explosive growth in recent years. It involves constructing systems that learn to make decisions by fitting mathematical models to observed data [20]. ML algorithms can generally be divided into three categories: supervised, unsupervised, semi-supervised and reinforcement learning.

– *Supervised learning* involves building models where both input data and expected output data (labels) are available. In this type of learning, models learn to map these inputs to their corresponding outputs. The goal is to learn a mapping function that generalizes well to unseen data. For example, training a regression model with data on age and years of children to predict their height [20].

– *Unsupervised learning* involves constructing models from input data that is not labeled, meaning there is "no supervision." The goal is to identify patterns and understand the structure of the data. It is particularly useful for exploratory data analysis and preprocessing [20]. An example of this is a clustering model created with a k-means clustering algorithm. The algorithm partitions data into k clusters, where each data point belongs to the cluster with the nearest mean.

– *Semi-supervised learning* lies between supervised and unsupervised learning. It utilizes a small amount of labeled data together with a larger set of unlabeled data. In this type of learning, models can use patterns in the unlabeled data to make their decisions. The assumption is that the unlabeled data can provide valuable information about the data distribution, which can be used to enhance the model's generalization [20]. Semi-supervised learning is especially valuable when labeling data is expensive or time-consuming.

– *Reinforcement learning* introduces an intelligent agent that learns to take actions based on rewards, aiming to choose the set of actions that lead to the highest rewards on average [20]. For instance, teaching a robot to win a chess game by giving it rewards every time it captures a piece, or giving a reward at the end of the game if it wins. This way, the robot learns which sequence of actions constitutes a strong or weak play.

Currently, cutting-edge methods in these four areas rely on *deep learning* (DL). See Figure 2.1). DL is a subfield of ML that focuses on learning hierarchical representations of data through complex algorithms called *deep neural networks* (DNNs). These are particularly powerful for tasks such as image and speech recognition, natural language processing, and more [20].

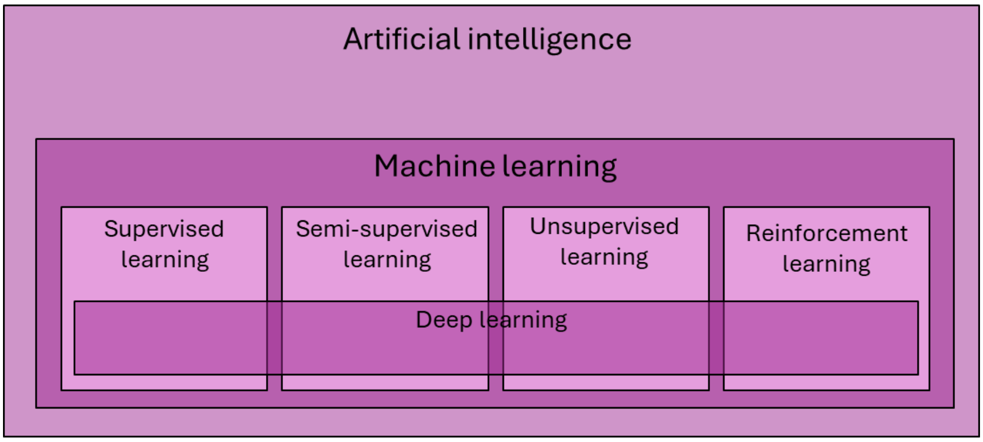


Figure 2.1: Artificial intelligence, machine learning, and deep learning. Adapted from [20].

This work exclusively employs supervised learning models. Consequently, discussions on unsupervised learning, semi-supervised learning, and reinforcement learning are beyond the scope of this research. The following section delves into the supervised learning models that will be employed in the present work, highlighting their capabilities and applications.

2.1.1. Supervised learning models

A *supervised learning model* defines a relationship between one or more inputs to one or more outputs. A simple example could be a model predicting the price of a house as an output, using the number of rooms and the walking distance to the closest subway station as inputs. The model can be interpreted as a mathematical equation where the output "y" is related to the input "x". This equation has parameters ϕ that, when set up correctly, define the particular relationship of "x" with "y" ($y = f(x, \phi)$).

The process of determining the parameters that accurately represent the relationship between inputs and outputs is known as *model training*. This involves using a learning algorithm that processes sets of inputs along with their expected outputs, referred to as *training data.* When the inputs are passed through the equation, the output is calculated, and this process is named *inference.* The algorithm iteratively adjusts the model's internal parameters to minimize the difference between the predicted outputs and the actual outputs for each set of inputs. To clarify, algorithms are systematic procedures or formulas designed to solve problems or execute tasks. They consist of step-by-step instructions that guide the learning process. Models are the representations of the patterns or relationships learned from the data. They are the resultant outputs produced by algorithms after being trained on the data.

The goal is for the model to learn from the training set to accurately predict outputs for new, unknown inputs. Generally, if the model performs well on the training dataset, it is likely to make accurate predictions for new, unseen data where the true output is unknown [20].

To evaluate the performance of a model, it is essential to test it on a separate dataset known as *test data*, which the model has not encountered during training. If the model performs well on the test data, it is considered ready for deployment. Additionally, a third dataset, called *validation data*, is often used during training to fine-tune the model. While the training dataset is used to fit the model, the validation dataset is employed to tune the model's parameters before its performance is assessed on the test dataset.

– **Loss**

The degree of mismatch between the predictions and the actual outputs used in the training process can be quantified with the *loss* (L). This is a scalar value that evaluates how poorly the model predicts the outputs of the training data from their respective inputs. It quantifies the difference between the predicted values and the true values, providing a numerical value that the learning algorithm aims to minimize during the training.

The loss helps to adjust the model's parameters to minimize errors and improve performance. When training the model, the goal is to identify the parameters that minimize the loss function [20].

Loss functions exist for the different tasks models can perform (regression, classification, etc.). Even for the same task, different loss functions can make the model learn in distinct ways because they prioritize and penalize errors differently, which influences the optimization process.

The recognition of emotion is typically perceived as a classification task. Thus, only loss functions for classification are introduced here. A short description of the common loss functions is given below [21].

a) **Cross-entropy loss**

Cross entropy describes the difference between the predicted output and the expected output. The predicted output of the classifier is a probability value between 0 and 1 for each class.

The cross-entropy value gets smaller as the probability distributions of the predicted and expected outputs get closer. The formula is shown in Formula 2.1, and it is used in multi-class classification problems.

$$L = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i)] \tag{2.1}$$

Where n is the number of classes, y_i is the expected probability for the class, and $\hat{y}_i$ is the predicted probability for that class.

b) **Binary Cross-Entropy (BCE) / Log loss**

BCE is a special case of cross-entropy used for binary classification (i.e., two classes). The terms BCE and Log loss are interchangeable in practice. The formula is shown in Formula 2.2.

$$L = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{2.2}$$

Where y_i is the actual label (0 or 1) of the i-th observation, and $\hat{y}_i$ is the predicted probability of the i-th observation being in class 1.

c) **Hinge loss**

The Hinge loss is designed for binary classification tasks. It create a margin around the decision boundary, promoting robust separation between classes. The formula is observed in Formula 2.3.

$$L = \max(0, 1 - y \cdot \hat{y}) \tag{2.3}$$

Where $y_i \in \{-1, 1\}$ is the actual label of the i-th observation, and $\hat{y}_i$ is the predicted label of the i-th observation.

The smaller the loss gets after minimizing the loss function, the more accurately the parameters of the model predict the training outputs from the training inputs.

A wide variety of supervised learning models have been developed, ranging from classical statistical models to modern deep learning architectures. In the next sections, Section 2.1.2 and Section 2.1.2.1, the two relevant types of ML models for the present study are introduced: support vector machines (SVMs) and neural networks (NNs). Other common ML models for classification are briefly described, in Section 2.1.3.

2.1.2. Support vector machines

SVM is a supervised machine learning algorithm originally developed for binary classification [22]. Nevertheless, it has been adapted to successfully handle multi-class classification and regression tasks. While it can handle regression problems, SVM is particularly well-suited for classification tasks.

First, it is necessary to introduce the concept *hyperplane*, which refers to flat affine subspaces that separate a space or hyperspace by half. A hyperplane is represented as a line in a 2-dimensional space, or as a plane in a 3-dimensional space. Nonetheless, the term hyperplane is typically only used when the hyperplane exists in a space with four or more dimensions.

SVMs aim to find the optimal hyperplane, also referred as *hard margin*, which is the one that maximizes the perpendicular distances to the closest points of different classes. In other words, a SVM helps finding the hyperplane that is as far away as possible from the nearest data points of each class, maximizing the margin (the perpendicular distance between the hyperplane and the closest data points). Ensuring a clear separation between the classes. The data points that are closest to the decision boundary are known as *support vectors* [23, 24, 25]. See Figure 2.2.

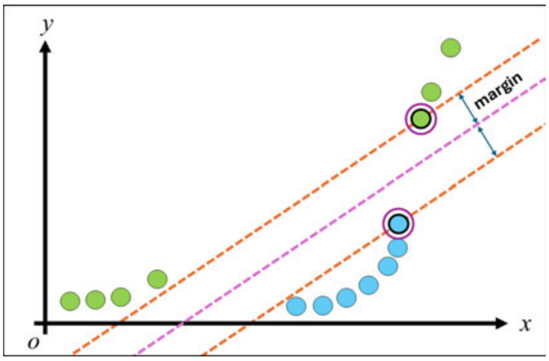


Figure 2.2: The margin is the gap between the boundary that separates the classes and the nearest data points. The support vectors are the key data points that define this boundary. By maximizing the margin, SVMs aim to achieve better separation and classification of the data.

Inference is made in a similar way to other ML models. The input is a set of inputs **X** for each sample, multiplied by a set of parameters known as *weights* **W**, and finally adding another parameter called *bias* **b**. Weights are parameters in a model that determine the influence of input features on the output. They are adjusted during training to minimize the loss function and improve model accuracy. Bias terms are additional parameters that allow the model to adjust the output independently of the input features.

The linear combination of weights and input features predicts the output $\mathbf{y}$, as observed in equation (2.4).

$$\mathbf{W}\cdot\mathbf{X} + \mathbf{b} = \mathbf{y} \tag{2.4}$$

The *support vectors* are crucial to determine the position and orientation of the boundary, these datapoints are the ones used by the algorithm to optimize the weights of the model, which makes it different from NNs, where all datapoints are used for optimizing the model's parameters.

The algorithm penalizes the model for every misclassification. If a data point is classified correctly and within the margin, there is no penalty ($loss = 0$); on the other hand, if a data point is classified incorrectly, the loss increases proportionally to the distance of the datapoint to the boundary (distance of violation).

In some scenarios, the separation of data is challenged due to the presence of outliers (see Figure 2.3). In such cases, the concept of a *hard margin* that perfectly separates the data is not feasible. To address this, a *soft margin* is used.

A soft margin allows for some misclassifications or violations of the margin, which improves generalization and increases the model's robustness against outliers. However, it is important to note that with the soft margin framework, a wider margin comes at the cost of more misclassifications, while a smaller margin results in a model that is less tolerant to outliers [24, 23].

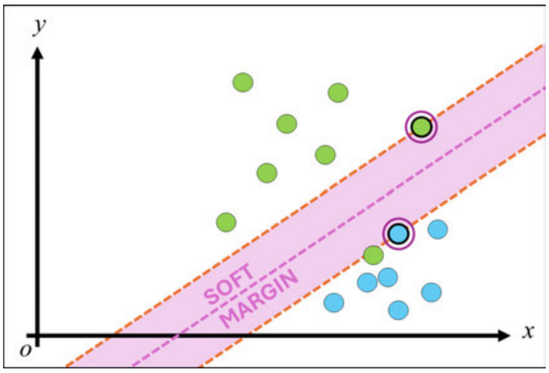


Figure 2.3: Soft margin in SVMs

The real power of SVMs lies in their ability to handle non-linearly separable data (i.e., data that cannot be divided by a straight line). SVMs achieve this by transforming data that is not linearly separable in its original form into a higher-dimensional space where it becomes easier to separate. This transformation facilitates the identification of a decision boundary (hyperplane) even for non-linear data [25, 26]. See Figure 2.4.

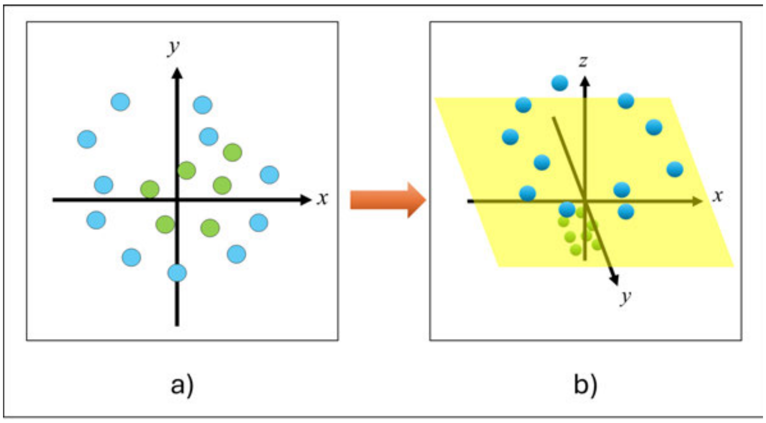


Figure 2.4: a. Original data in 2D space cannot be separated by a hyperplane (one-dimensional line on a plane), b. Data can be separated by a hyperplane (a two-dimensional plane in a space) after being transformed to a 3D space.

SVMs utilize special mathematical functions called *kernels* o map data to a higher-dimensional space. These kernels enable the algorithm to perform necessary calculations while remaining in the original, lower-dimensional space, thereby avoiding the complexity of computations in the higher-dimensional space and making the process more efficient and manageable [25].

Numerous kernels exist and can be customized, but three kernels stand out as the most commonly used [27]:

a) Radial Basis Function (RBF) kernel, also called Gausian kernel, measures the similarity between datapoints based on their Euclidean distance in the feature space and maps the input feature vector to an infinite-dimensional feature space using a Gaussian function. This kernel can handle more complex relationships [23].

b) Polinomial kernel, which transforms data into a polynomial feature space of any order, the higher the degree of the polynomial, the better the kernel captures the affinities in a non-linear dataset [28].

c) Linear kernel is the simplest kernel function, which maps the input space to itself. Computes the dot product between feature vectors. It is used for linear classification. Thus, effective for linearly separable data [23].

Different types of kernel functions exist for SVMs, each kernel function can produce different classification results. There is no universal rule for choosing the best kernel, so it is often beneficial to experiment with various kernels on the training datasets to see which one performs best [26].

In summary, SVMs are extremely powerful and versatile classifiers. Their ability to handle high-dimensional data and their effectiveness in class separation make them valuable tools in the field of machine learning. These makes them worthy contenders against NNs, which will be explored in detail next.

2.1.2.1. Neural networks

Artificial Neural Networks (ANNs), simply referred as NNs, were first introduced in 1943 by McCulloch and Pitts [29], a neurophysiologist and a mathematician. In their paper, they presented a simple computational model of a biological neuron that was able to implement basic Boolean logical operations such as AND, OR and NOT (Figure 2.5).

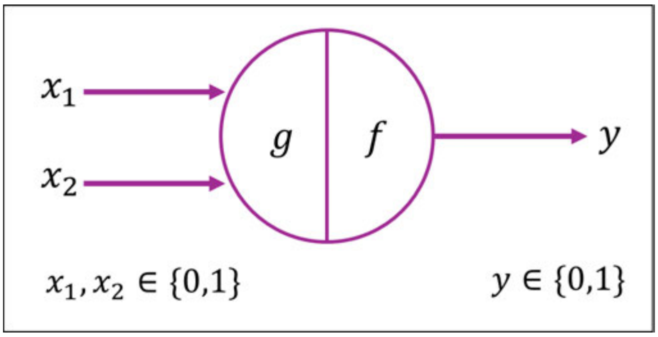


Figure 2.5: Depiction of a McCulloch-Pitts Neuron with Two Inputs: This early model of a neural network demonstrates the basic structure of artificial neurons, including the integration of input signals and a threshold function. Adapted from [25].

In Figure 2.5, the binary inputs x_1 and x_2 are added together by the function $g(x)$, then the value is passed to a threshold $f(g(x))$. If the sum of $x_1 + x_2$ is greater than or equal to the threshold, denoted (θ), then $y = 1$, if not, $y = 0$[30, 25].

With this description, an AND Boolean logic gate can be designed with $\theta = 2$, where the output y is equal to 1 only if both x_1 and x_2 have a value of 1. When $\theta = 1$, an OR logic gate is achieved, meaning that if either x_1 or x_2 is 1, then $y = 1$.

This concept can be generalized to a greater sequence of inputs, x_1, x_2, x_3, $\cdots$, x_i. The NN calculates the sum of the inputs and performs the thresholding that assigns the value of y, giving the mathematical equations below (Equations 2.5, 2.6, and 2.7).

$$g(x) = \sum_{i=1}^{n} x_i \tag{2.5}$$

$$f(z) = \begin{cases} 1, & \text{if } z \geq \theta \\ 0, & \text{otherwise} \end{cases} \tag{2.6}$$

$$y = f(g(x)) = \begin{cases} 1, & \text{if } g(x) \geq \theta \\ 0, & \text{otherwise} \end{cases} \tag{2.7}$$

Although this set the beginning of NNs, the neuron is not learning, only computing. The value of θ had to be manually selected, and the neuron could not learn from the input data to figure out θ. It was not until 15 years later that the first neuron capable of learning was invented by Rosenblatt [31]. He presented the *Perceptron*, an improved version of the McCulloh-Pitts neuron infused with a learning algorithm (Figure 2.6) [30, 25].

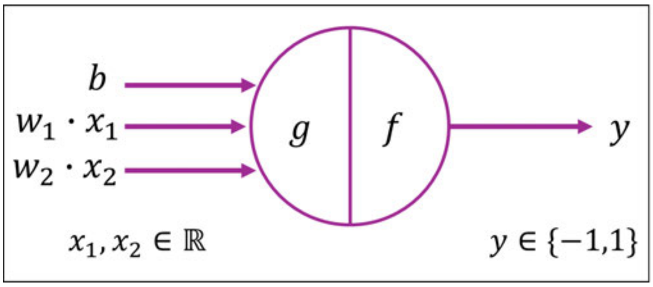


Figure 2.6: Illustration of Rosenblatt's Perceptron with two inputs: This simple neural network model includes weights and a bias term, demonstrating the fundamental components of a Perceptron. Adapted from [25].

The Perceptron differs mainly in that its inputs can take any real number and do not need to be binary. These inputs are multiplied by weights and then summed to generate a weighted sum. Additionally, there is a bias term (b). Both the weights and the bias are real numbers. The output $y \in \{-1, 1\}$ is defined by a thresholding function $f(z)$, known as the *activation function.*

Many activation functions exist, but typically the **Heaviside step function** and the **signum function** are employed for this type of NN. The plots of both functions are shown in Figure 2.7 [30, 25].

Unlike the old model, the Perceptron can be trained to find correct values for the weights and the bias to solve a task. Its mathematical notation is given below in Equations 2.8 and 2.9.

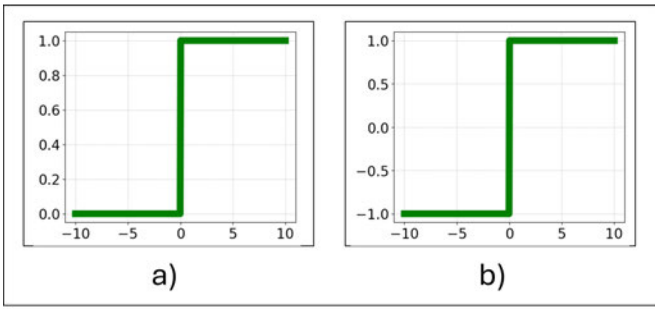


Figure 2.7: Common activation functions for Perceptron model. a) Heaviside step function, b) Signum function.

$$g(x) = \sum_{i=1}^{n}(w_i \cdot x_i) + b \tag{2.8}$$

$$y = f(g(x)) = \begin{cases} +1, & \text{if } g(x) > 0 \\ -1, & \text{otherwise} \end{cases} \tag{2.9}$$

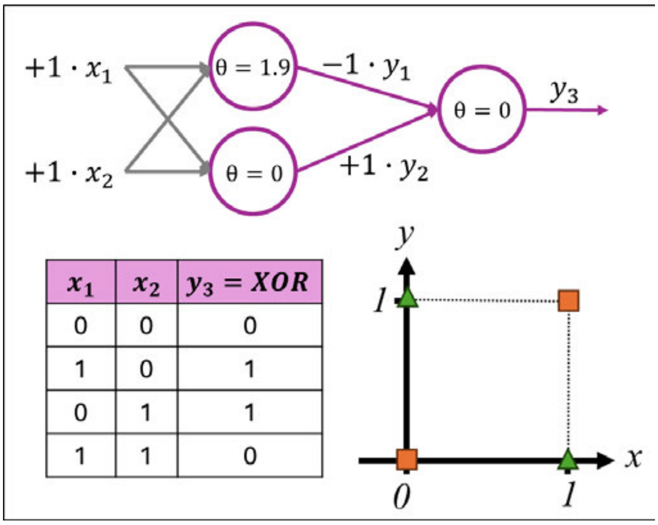
During training, the Perceptron starts with its weights and bias initialized to zero. A set of inputs is provided, a prediction is made, and this prediction is compared against the expected output. The weights are then adjusted to minimize the difference between these two values. Eventually, the Perceptron finds at least one set of values for its weights and bias term that produce the correct outcome [30, 25].

The Perceptron is designed for binary classification of linearly separable data, meaning it cannot learn complex non-linear patterns. This limitation can be overcome by stacking multiple Perceptrons, resulting in a more complex ANN called a *Multi-layer Perceptron* MLP. The MLP is capable of learning complex patterns in the training data and can solve multiclass classification and regression tasks.

For instance, an MLP can solve the XOR logical operation, which a simple Perceptron cannot [30]. An example of this is shown in Figure 2.8, where the outputs of two Perceptrons in the first layer serve as inputs to another Perceptron in the next layer.

In MLPs, the inputs are connected to all neurons in the first layer, also called the hidden layer. The outputs of these neurons are used as inputs for all neurons in the next hidden layer, and this process repeats for each hidden layer until reaching the final layer of neurons, called the output layer. For each hidden layer, a bias term is added, which is also connected to the neurons of the following hidden layer.

Furthermore, all outputs of the hidden layers are multiplied by weights. Each neuron calculates the sum of the weighted inputs and the bias term, then passes the value through an activation function. MLPs are also called *fully connected neural networks*



x_1	x_2	$y_3 = XOR$
0	0	0
1	0	1
0	1	1
1	1	0

Figure 2.8: MLP with parameters adjusted (weights and bias terms) for solving the XOR problem. All bias terms are zero, and the weights are the coefficients next to the variables.

(FCNNs) because all elements of one layer are connected to all elements of the following layer (Figure 2.9).

The number of neurons on the hidden layers can vary. When ANNs have two or more hidden layers, they are called *deep neural networks* (DNNs). Conversely, if they have fewer than two hidden layers, they are referred to as *shallow neural networks.*

The output layer should contain a number of neurons corresponding to the expected outputs. For example, if the NN is solving a classification problem for four classes, the output layer should contain four neurons, one for each class. For binary classification, one neuron in the output layer is often sufficient and considered more optimal due to fewer parameters needing adjustment.

The next question to address is how to adjust the parameters of an NN, or in other words, how to train an MLP. The *backpropagation* and *gradient descent* algorithms are typically used for training NNs.

During the training process, the backpropagation algorithm feeds each set of inputs to the network, and each neuron computes an output that is passed to the next layer of neurons. This is known as the *forward pass.* The algorithm then measures the difference between the network's output and the actual output for that set of inputs using a loss function. This difference is referred to as the network's output error.

Starting from the last hidden layer, the backpropagation algorithm calculates the contribution of each neuron to the network's output error. It then determines how much of these error contributions originated from each neuron in the previous hidden

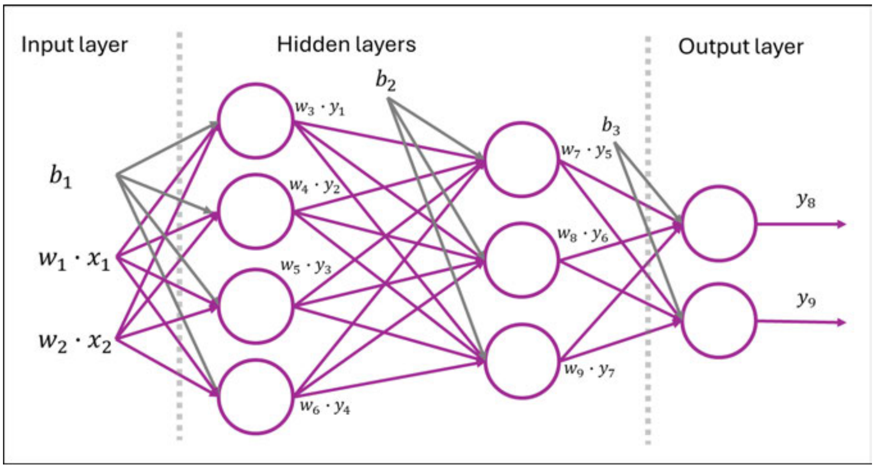


Figure 2.9: Multi-layer Perceptrons are also known as fully connected neural networks. MLP with an input layer, two hidden layers and an output layer, fully connected.

layer, repeating this process for each hidden layer until reaching the input layer. This procedure, known as the *reverse pass*, helps identify the contribution of each connection in the neural network to the overall error. The error gradient is propagated backward through the network, which is the reason for the algorithm's name.

Finally, using the previously computed error gradients, the network's weights are adjusted through a gradient descent step to minimize the error (loss). This process is repeated for each set of inputs in the training dataset, with each repetition counted as an iteration. The goal of backpropagation is to converge to the global minimum of the loss function with each iteration [30, 20].

Each weight is updated according to the calculation in Equation 2.10.

$$w_i^* = w_i - -\alpha \left(\frac{\partial L(\theta)}{\partial w_i} \right) \tag{2.10}$$

Where w_i^* represents the updated weight. w_i is the current weight. $\frac{\partial L(\theta)}{\partial w_i}$ is the gradient of the error determined by the loss function L with respect to the weight w_i. The gradient is the derivative of the loss function, defining the direction of the step necessary to reach the global minimum. Finally, α is the learning rate, which defines how big or small the step will be for each iteration.

The learning rate is a non-trainable parameter; therefore referred to as a hyperparameter. It is typically in the range between 0 and 1 and needs to be chosen carefully. A small learning rate makes the model take tiny steps with each iteration, converging slowly, and training the model will require multiple epochs until it has reached the minimum loss. An *epoch* refers to a complete pass through the training data. If the learning rate is too small, the loss can get stuck in a local minimum, missing the optimal weights. Conversely, a high learning rate can cause the model to overshoot optimal weights, making the loss bounce back and forth, and risking never converging.

The gradient descent algorithm needs a slope to determine which direction to move in to reduce errors. For the algorithm to work, it is necessary to introduce activation functions where a gradient could be calculated because step functions do not have a slope. Therefore, the **sigmoid function** (Formula 2.11) was introduced, which has a smooth curve and a well-defined non-zero derivative. The backpropagation algorithm can also work with other activation functions besides this one. Two other popular choices are the **tanh function** (Formula 2.12) and the **ReLu function** (Formula 2.13); these can be visualized in Figure 2.10.

Sigmoid function:

$$\sigma(z) = 1/(1 + \exp(-z)) \tag{2.11}$$

Hyperbolic tangent (tanh) function:

$$\tanh(z) = \frac{\exp(z) - \exp(-z)}{\exp(z) + \exp(-z)} \tag{2.12}$$

Rectified linear unit (ReLU) function:

$$\mathrm{ReLU}(z) = \max(0, z) \tag{2.13}$$

Typically, for multi-class classification the output layer is very often modified by replacing the individual activation functions of the neuron for a shared **softmax function**, in formula 2.14, that converts the output into predicted probabilities, this means that the outputs are now values between 0 and 1, and they add up to 1. This way, the outputs of the model can be easily interpreted; the class predicted for each set of inputs is the output with the highest probability [30].

Softmax function:

$$\mathrm{softmax}(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^{N} \exp(z_j)} \tag{2.14}$$

Once the predictions are done, the model's classification performance can be measured. To achieve that, evaluation metrics are employed. These help assess how well a model is performing as well as identify areas where the model needs improvement. Section 2.1.4 describes common metrics used in ML for assessing classification models' performance.

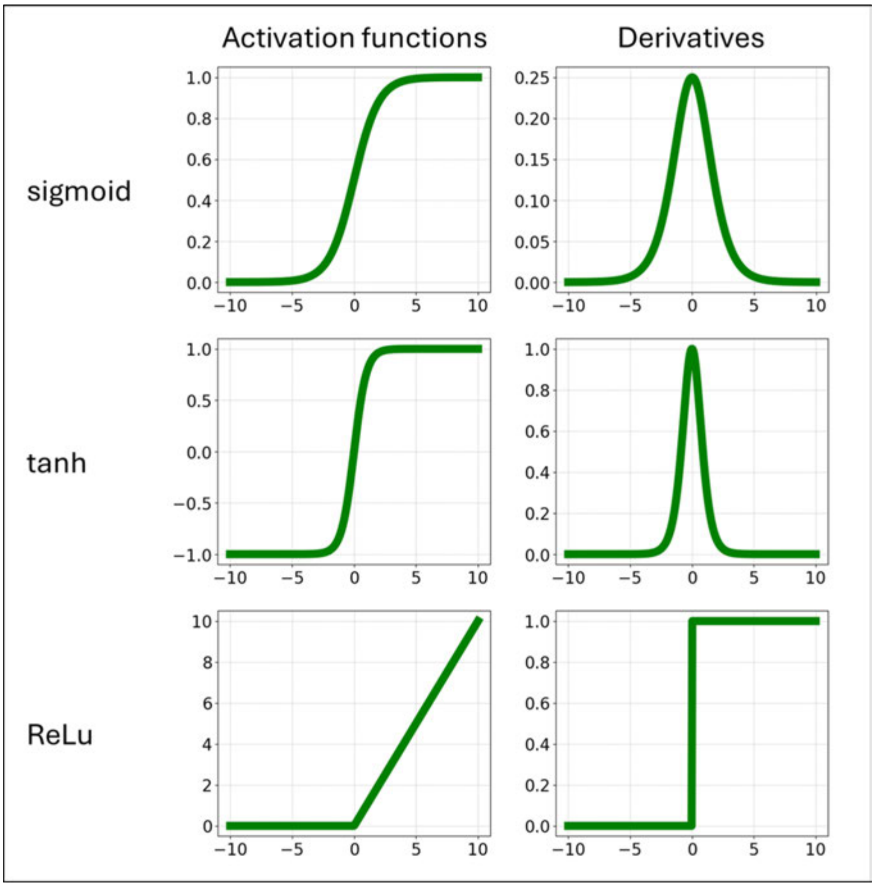


Figure 2.10: Activation functions for Multi-layer Perceptrons.

2.1.3. Common deep learning models used in SER

Below is a brief overview of the classification models that have seen frequent application in SER tasks.

- **Convolutional Neural Networks (CNNs):** CNNs are specialized NNs designed for processing structured grid-like data such as images or time-frequency representations of audio. The architecture typically includes multiple convolutional and pooling layers, followed by fully connected layers for classification. CNNs can achieve state-of-the-art performance but require substantial computational resources and large amounts of labeled data for training.
- **Graph Neural Networks (GNNs):** GNNs are designed to work with graph-structured data, where data points (nodes) are connected by edges. They use a message-passing mechanism to aggregate information from neighboring nodes, allowing the network to capture complex relationships and dependencies within the graph.

- **Recurrent Neural Networks:** RNNs are designed for sequential data, such as time series or natural language. They have connections that loop back on themselves, allowing information to persist across time steps. This enables RNNs to capture temporal dependencies and patterns in sequences. However, standard RNNs can suffer from issues like vanishing and exploding gradients, making it difficult to learn long-term dependencies.
- **Long Short-Term Memory Neural Networks (LSTM):** LSTMs are a type of RNN designed to address the limitations of standard RNNs. LSTMs use special units called memory cells that can maintain information over long periods. These cells have gates that control the flow of information, allowing the network to learn long-term dependencies without suffering from vanishing gradients.
- **Transformer-Based Models:** Transformers are a type of neural network architecture that use self-attention mechanisms to weigh the importance of different parts of the input sequence, allowing them to capture long-range dependencies more effectively than RNNs or LSTMs. Transformers consist of encoder and decoder layers, each with multiple attention heads and feedforward layers.

Each model presents different strengths and trade-offs depending on the nature of the data, feature representations, and the complexity of the target task. Deep learning has become a common approach in many speech processing tasks, including speech recognition, speaker identification, and emotion recognition, due to its ability to model complex, high-dimensional data effectively [32, 33, 34, 35].

2.1.4. Evaluation metrics

Multiple evaluation metrics exist in ML for assessing the performance of models. In ML, there are multiple metrics to choose from to assess the models, typically, the more metrics calculated, the better. Nevertheless, not all metrics are suitable for all models, things like the distribution of labels in the dataset (balanced/imbalanced), the type of task regression, classification, among others.

Next, four of the most common metrics used in multiclass-classification tasks are introduced [36].

a) Confusion matrix (CM)

CM is a table that helps with visualizing the counts of True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) for each class.

- **True Positives (TP)** are the correctly predicted positive instances.
- **False Positives (FP)** are negative instances incorrectly predicted as positive.
- **True Negatives (TN)** are instances that are correctly classified as negative.

- **False Negatives (FN)** are instances that are actually positive and were incorrectly classified as negative.

The CM for binary classification is shown in table 2.1.

Table 2.1: Confusion matrix for binary classification

	Predicted Positive	**Predicted Negative**
Actual Positive	TP	FN
Actual Negative	FP	TN

In a multi-class classification problem, classes cannot be directly represented as positive or negative values. Therefore, the counts of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) cannot be computed straightforwardly due to the involvement of multiple labels. The approach involves calculating the TP, TN, FP, and FN values for each class individually.

For example, in a multi-class classification with three classes A, B, and C, the calculations for class A would be as follows:

- TP: the number of instances labeled as A, that were correctly predicted as A.
- TN: the number of actual instances of A, that were incorrectly predicted as B or C.
- FP: the number of instances of B and C, that were incorrectly predicted as A.
- FN: the number of instances of A, that were incorrectly predicted as A.

In a similar way, the values can be calculated for classes B and C. To complement the example, the confusion matrix is presented in Table 2.2.

Table 2.2: Confusion matrix for multi-class classification, example for three classes.

	Predicted A	**Predicted B**	**Predicted C**
Actual A	TP_A	$FP_{A,B}$	$FP_{A,C}$
Actual B	$FP_{B,A}$	TP_B	$FP_{B,C}$
Actual C	$FP_{C,A}$	$FP_{C,B}$	TP_C

Where:
- TP_i are the correctly predicted instances of class i.
- $FP_{i,j}$ are the instances of class i incorrectly predicted as class j.

b) Accuracy

Accuracy measures the proportion of correct predictions made by the model out of the total number of predictions. Its mathematical definition is shown in Formula 2.15.

$$\text{Accuracy} = \frac{\text{number of correct classifications}}{\text{total classifications}} \tag{2.15}$$

Accuracy ranges from 0 to 1, it can also be expressed as percentage. When the number of instances in each class is roughly equal (balanced), accuracy provides a good measure of overall performance. However, for imbalanced datasets (i.e., the labels are not evenly distributed) the accuracy can be misleading.

c) Recall and unweighted-average recall UAR

UAR is the average of the recall values for each class. A recall score measures the model's ability to correctly predict positives from actual positives. In multi-class classification the recall score is calculated for each class individually, then the average from all recall scores is obtained. When the average is weighted according to the distribution of labels this is known as *Weighted Average Recall*, and the classes with more labels will influence the value the most. In these study, the performance of the model will be done by treating all classes equally, to compensate the imbalanced distribution of labels. Thus an *Unweighted Average Recall* (UAR) score is used. The formulas to calculate the recall for each class, and the unweighted-average of the recalls are presented in Formulas 2.16 and 2.17.

- Recall for each class:

  $$\text{Recall}_i = \frac{TP_i}{TP_i + FN_i} \tag{2.16}$$

  Where i represents each class. TP are the correctly predicted positive instances for class i, and FN the incorrectly predicted negative instances for same class i.

- Unweighted average of recalls:

  $$\text{UAR} = \frac{1}{N}\sum_{i=1}^{N} \text{Recall}_i \tag{2.17}$$

  Where N is the total number of classes and i represents each class.

The range of the recall values is between 0 and 1, the higher the value, the better the performance of the model. The values can be represented as percentages, as well.

d) Macro F1–score

The F1 score is the harmonic mean of precision and recall, combining both in one value. This metric is especially useful in imbalance datasets, since a low value in either precision or recall for impacts significantly the F1–score. The harmonic mean gives more weight to lower values, ensuring that the F1 score is high only when both precision and recall are reasonably high. The value is in the range between 0 and 1.

As expected, in multi-class classification analysis this value is calculated for each class individually. When the F1–scores of all classes are averaged it is referred as 'macro-averaging'.

In order to calculate the individual F1 scores, precision and recall scores for each classes are required. Recall was already described above and its formula is shown in Formula 2.16. Precision is a measure that assess how many of the instances classified as positive are actually positive (Formula 2.18).

- Precision for each class:

$$\text{Precision}_i = \frac{TP_i}{TP_i + FP_i} \tag{2.18}$$

Where i represents each class.

With both, precision and recall scores, the F1 scores can be obtained for each class through the formula 2.19.

- F1–score for each class i :

$$\text{F1}_i = 2 \times \frac{\text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \tag{2.19}$$

Then the average of the F1–scores are calculated with formula 2.20.

- Macro F1–score i :

$$\text{Macro F1–score} = \frac{1}{N} \sum_{i=1}^{N} \text{F1}_i \tag{2.20}$$

The values are in the range of 0 to 1, but can be represented as percentages, as well.

2.2. Emotion theories

While everyone has an understanding of what emotion is, articulating its exact meaning in words is challenging for most. Some might describe it as a feeling, others as a body response to certain stimuli, and some even use it as a synonym for "mood". Despite being a core concept in psychology, there is still no consensus among researchers on its precise definition. The complexity of emotions makes them difficult to define, which in turn complicates their classification and understanding [37, 38].

In the work of Hartmann [37], a total of 72 definitions of the word *emotion* from various publications were reviewed to identify the key features of emotions. Through their research, they developed their own definition of emotion:

> *"Emotion can be defined as a bounded episode in the life of an individual that is characterized as an emergent pattern of synchronization between changing states of different subsystems of the organism (which are considered as components of the emotion), preparing adaptive action tendencies to relevant events as defined by their behavioral meaning (as determined by recurrent appraisal processes), and thus having a powerful impact on behavior and experience."*

Although the issue of defining the key attributes of emotions remains unresolved, the definition of emotion presented in the study encompasses all the ideas necessary to build this concept. Emotions drive and shape thoughts and actions, preparing individuals to respond to environmental challenges and directing attention to signals that indicate important needs and desires [39, 40]. Understanding the theoretical foundations of emotion is essential for developing effective emotion recognition systems. This section explores the key theories of emotion that have shaped the understanding of how emotions are experienced and expressed. There is no single standard for modeling emotions. Researchers often define emotions either as discrete constructs or continuous dimensions.

2.2.1. Discrete emotions

One of the most influential theories in the study of emotions is Paul Ekman's theory of basic emotions [42]. Ekman proposed that there are a limited number of basic emotions that are universally recognized across different cultures and are associated with distinct facial expressions and physiological responses, suggesting that they have a biological basis. These basic emotions are anger, disgust, fear, happiness, sadness and surprise. They have been widely used in affective computing.

Discrete constructs with more categories have been proposed in other studies, where the basic emotions are combined to create "secondary emotions", creating more comprehensive emotion category sets. For example, Robert Plutchik's *"wheel of emotions"*

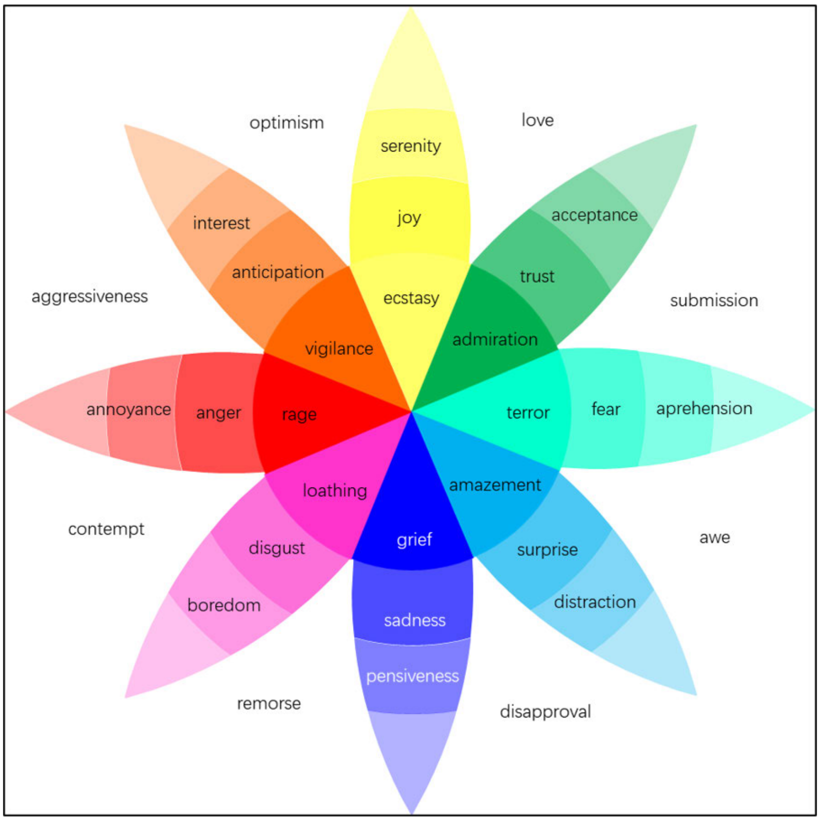


Figure 2.11: Plutchik's wheel of emotions. Adapted from [41].

[43], observed in figure 2.11, suggests that basic emotions can combine to form more complex emotions, similar to how primary colors mix to create other colors. This model highlights the dynamic and interconnected nature of emotional experiences.

2.2.2. Dimensional emotions

Despite being widely used, discrete constructs still fail to capture the complexity of some emotional states encountered in everyday communication. Dimensional constructs, on the other hand, define emotions as points in a space characterized by continuous dimensions.

The most common dimensions observed are valence, arousal, and dominance [44, 45]. The valence dimension represents the level of pleasantness or unpleasantness and is also called "the dimension of pleasure." For instance, both anger and fear are unpleasant emotions and are classified on the negative valence side, while joy is on the positive valence side as a pleasant emotion.

The arousal dimension indicates the level of activation or energy that the emotion produces. For example, anger and excitement have high activation or high arousal, whereas tiredness and depression have low arousal.

The dominance dimension depicts the feeling of control over the environment, with lower dominance indicating a more submissive emotion. For instance, while both anger and fear are unpleasant emotions, anger has higher dominance, and fear is a more submissive emotion, thus classified as low dominance.

In the dimensional approach, emotions are not independent of each other; instead, they are systematically analogous [40]. One prominent dimensional construct is the *circumplex model* [46, 47], which uses arousal and valence as perpendicular axes on a two-dimensional plane to represent emotional states with continuous values. Arousal ranges from low to high, with neutral in between, and valence ranges from negative to positive, also with neutral in between. Figure 2.12 illustrates this model.

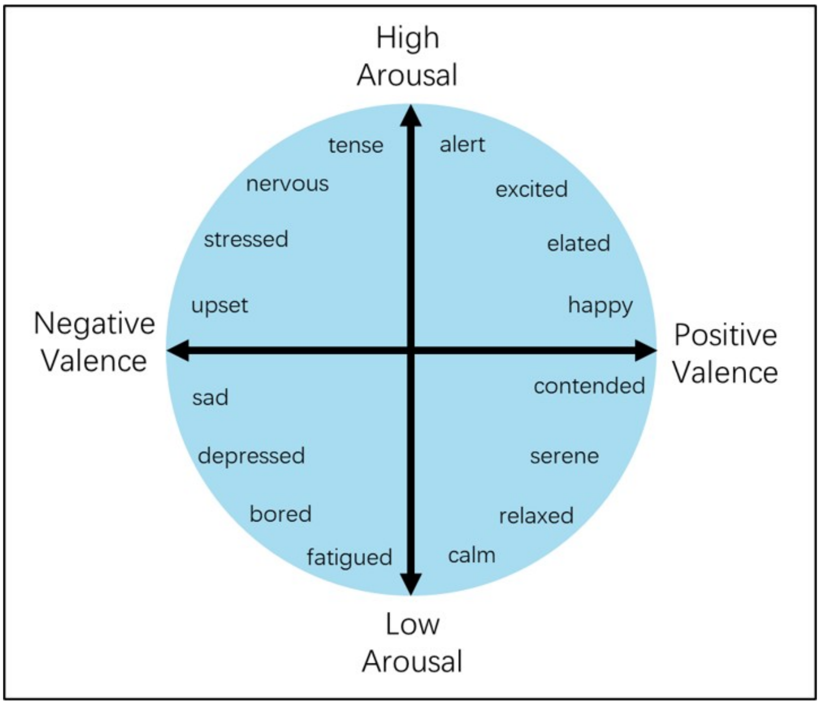


Figure 2.12: Circumplex emotion model. Adapted from [47].

Continuous constructs allow for a more nuanced representation of emotional states, capturing the complexity of human emotions.

2.2.3. Appraisal emotions

Finally, the hybrid theory, or appraisal theory, can be viewed as an extension of the dimensional construct [48]. This theory explains how emotions are influenced by evaluations (or appraisals) of events. Rather than being automatic responses to stimuli, this theory proposes that emotions result from the interpretation and assessment of situations.

The concept of the appraisal construct was significantly developed by psychologist Magda Arnold in 1960 [49]. According to her, appraisal is a straightforward, instantaneous, and instinctive process that does not initially necessitate identifying the object being evaluated.

Appraisal constructs help explain why different people can have different emotional reactions to the same event. For example, if a clown walks into a party, some people might find this amusing and laugh (positive appraisal), while others might feel scared (negative appraisal) based on their past experiences and interpretations of clowns.

While this theory offers a detailed and flexible framework for understanding emotions, it is less frequently adopted in automatic emotion recognition systems due to its complexity and the challenges associated with measuring cognitive appraisals. Instead, models like the "Circumplex model" and Ekman's discrete emotion categories are more commonly used because they are simpler, easier to implement in computational systems, and widely supported by labeled datasets for emotion recognition.

In the next section, the most common methods used to recognize emotions automatically will be presented.

2.3. Automatic emotion recognition

Emotions play a key role in human interaction, as they help us understand and connect with one another. Humans express emotions in various ways, such as through facial expressions, body language, voice, and actions. This emotional communication is essential for building relationships and fostering empathy. Given the importance of emotions in our interactions, it is a natural outcome to extend this ability of recognizing and interpreting emotions to computers and machines [40, 14, 50].

The field of **affective computing** aims to develop systems and devices capable of recognizing, interpreting, processing, and simulating human emotions. It also encompasses the creation and interaction with machine systems that respond to and influence emotions. This research field integrates insights from various disciplines, including psychology, physiology, engineering, sociology, mathematics, computer science, education, and linguistics, to achieve its objectives [51].

The concept of affective computing originated with Rosalind Picard, a professor at the Massachusetts Institute of Technology (MIT) Media Lab. In 1995, Picard published a seminal paper titled *"Affective Computing"*, and in 1997, she expanded on these ideas in her book of the same name. Her work laid the foundation for the field, emphasizing the importance of giving machines emotional intelligence to improve human-computer interactions [52, 53]. Since then, the field has grown to include various applications, such as Emotion Recognition (ER), emotion synthesis, and the development of conversational agents such as Siri[1] and Alexa[2], which can simulate emotional responses [54].

The motivation behind affective computing is to create systems that can understand and respond to human emotions, thereby enhancing user experiences and making interactions with technology more natural and intuitive. This involves using various sensors and advanced computational algorithms to detect and interpret emotional cues from facial expressions, speech, body language, and physiological signals.

Multiple methods exist for recognizing emotions automatically. Since emotions trigger changes not only in the psychological state but also in the physiological state of the organism, it is possible to detect emotions from signals emitted by the human body. Among the signals currently used for this purpose are physical signals like facial expressions, speech, gestures, and body postures. Even text is considered a physical signal, according to [55]. Additionally, physiological signals are analyzed for ER, such as electroencephalogram, electrocardiogram (ECG), electromyogram, galvanic skin response, respiration, skin temperature, photoplethysmography, and eye tracking [56, 57, 58].

[1] Siri is a virtual assistant created by Apple, designed to perform tasks such as sending messages, setting reminders, and answering questions through voice commands.

[2] Alexa is a virtual assistant developed by Amazon, capable of voice interaction, music playback, setting alarms, and providing real-time information.

Speech is a quick, natural, and likely the simplest method of communication between humans. It stands out as the most suitable signal for emotion recognition, not only because it is easier to collect, but also because it requires less storage compared to image or video data used for facial or body posture analyses. For the latter two, participants typically have concerns such as embarrassment or protecting their confidentiality due to the exposure of their appearance, which does not occur with speech analysis. Additionally, speech conveys information beyond emotion, such as age, gender, and language of the speaker. It can even reflect their way of thinking and cultural context [54, 48].

2.3.1. Speech emotion recognition

Speech emotion recognition (SER) is a subfield of affective computing that focuses on identifying emotions from the voice. The history of SER starts long before Picard's work. As mentioned in [59] the origins of SER date back to a century ago with Blanton's work *"The voice and the emotions"* [60] where he wrote "The effect of emotions upon the voice is recognized by all people", although his publication was made in 1915, the first patent in the field of SER occurred 60 years later, in the late 1970s [61]. Nevertheless, it has been during the past two decades that the field has matured significantly, thanks to the advancements in ML and DL techniques [40, 48, 34, 62, 55].

Over the past few years, multiple research challenges have been established to advance the field of SER. These challenges allow researchers to compare their affect recognition systems with benchmark performances [63], typically providing a database, specific feature sets, and baseline results. Among these are the ten editions of the Emotion Recognition In The Wild Challenge (EmotiW) [64, 65, 66], which cover multimodal emotion recognition from audio-visual data; and the INTERSPEECH Computational Paralinguistic ChallengE (ComParE) editions from 2013 to 2023, which focus on tasks of high relevance for affective and behavioral research [67, 68].

In general, there are three key steps in the process of recognizing emotion from speech: collection of speech data (speech database), selection and extraction of features from speech, and automatic classification of emotion from the extracted features [69, 48]. This process can be illustrated in Figure 2.13.

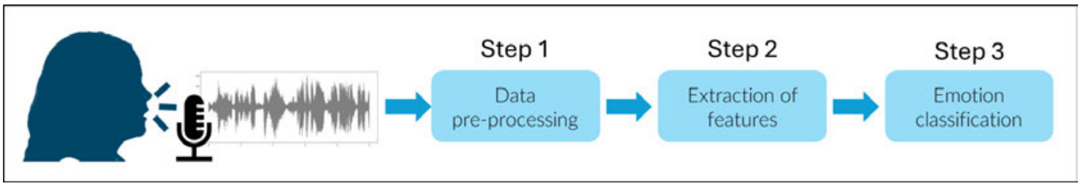


Figure 2.13: Steps of SER. Adapted from [70].

2.3.1.1. Collection of speech data

To achieve a good analysis, high-quality data is essential. A reliable database is crucial for developing SER. The importance of the database in the entire SER process should not be overlooked; if the database is incomplete, of low quality, or contains erroneous data, it can lead to incorrect conclusions. The success of emotion classification through SER largely depends on the labeled data. Databases for SER are generally divided into three categories [71, 40]:

–Acted (simulated) speech emotion databases:

In this type of database, the audio is recorded in controlled environments, typically a soundproof studio, where a certain number of actors repeat the same phrase using different intonations to express the targeted emotion. The meaning of the phrase is not relevant, but rather the way it is expressed. This type of database is the most widely used, as data collection is done in a controlled and standardized manner, offering high-quality, noise-free data, thus allowing for easy comparison of results [72, 71, 69]. The disadvantage is that systems trained with this type of data are not the best at detecting emotions in real-life speech, where the environment is not controlled, the speakers variability is higher, and emotions can be more subtle compared to acted speech where emotions are clearer or even exaggerated [73, 40].

–Natural (spontaneous) speech emotion databases:

Data in this type of database consists of recordings where speakers have expressed themselves naturally and spontaneously. In other words, real-life speech is used, without a script or the expectation/forcing of an emotion. Collecting and distributing this type of data is not easy, as it requires ethical, financial, and legal considerations [74].

These databases are typically built from recordings obtained from talk shows, call-center recordings, podcasts, and social media [74, 40] where genuine interactions occur.

The main problem with this type of data is that the emotions need to be labeled by third parties, which makes the process error-prone, since the annotators' perception of emotion is subjective and does not always match the actual emotion of the subjects [73]. Additionally, the emotion span is usually very short, or the emotion changes within a few seconds, which makes the process time-consuming and expensive.

Another problem faced by Natural Speech databases is that they usually do not have the best audio quality, as the audios were recorded under different settings and uncontrolled environments, often containing background noise and unclear voices, adding more sources of variation to the system. However, this type of data is useful for training systems that attempt to recognize emotions from real-life speech, where random sources of noise are present, and it is impossible to control all variables.

An advantage of natural speech emotion databases is their great speaker variability, which means the speakers vary in terms of quantity, language, age, and gender. Databases with these extensive variations in speakers and recording conditions are also known as “in-the-wild” databases [8, 7]. These are difficult to create and are therefore not as widely used as acted speech databases. However, they allow emotion recognition systems to appropriately generalize emotions that occur in real-life speech [73].

–Induced (elicited) speech emotion databases:

In this type of database, the speaker is stimulated to provoke a targeted emotion in them. To achieve this, the subject is placed in a simulated situation. For example, in [75], subjects had to interact through their voice with a multimodal dialogue system. Their interactions were recorded, and it was observed how the subjects experienced various emotional states such as anger/irritation, surprise, joy, and helplessness while interacting with the system.

Induced emotions, although not obtained in a real-life context, are quite realistic. This type of speech occurs spontaneously as it is not scripted. However, the data lacks the diversity of speakers and environments found in natural-speech databases [76].

2.3.1.2. Feature selection and feature extraction

For machines to be able to recognize human emotions, quantitative data is needed. Feature extraction from speech implies obtaining measurable parameters from a speech signal. These parameters characterize the signal and allow the identification of specific patterns in the voice that are associated with different emotions. Typically, in SER, acoustic and linguistic features from speech are analyzed. While acoustic features contain information on *how* the speaker conveyed the message, linguistic features disclose information on *what* is said in the message.

As mentioned before, emotions affect the body’s physiology, causing tension and autonomic arousal in muscles involved in phonation (voice production) and speech articulation. These physiological changes are reflected in the voice and can be measured using purely acoustic features[77]. For example, a person experiencing happiness will be characterized by high mean pitch and high speech rate [1].

Although the fusion of acoustic-linguistic features has been found to be effective in SER for categorical and dimensional emotions, most achieving state-of-the-art results still rely on manually hand-crafted acoustic features. One reason for this could be the added complexity to SER systems when linguistic features are evaluated, as a word-by-word transcription is required to extract semantic information, which consumes time and resources. When transcription is not done manually, an ASR system is typically used for this task. Additionally, the text must still be processed, making the entire

SER process slower. Meanwhile, acoustic features can be extracted directly from the original audio signal, eliminating the need for additional processing steps [48]. As will be observed in Section 3, for the present work, only acoustic features will be used. Thus, only these types of features will be described.

There is no agreed-upon standard for building an acoustic feature vector for SER. However, it is common to categorize features according to their temporal structure, parametric structure, and acoustic properties [69, 48, 78]. Authors organize and select features in their feature sets according to their considerations. Below, some common groups of features used in acoustic feature sets for SER are described.

According to their temporal structure, features can be categorized into two fixed classes: *segmental features*, where parameters are extracted from short segments of the signal (usually 25-50 ms long), and *suprasegmental features*, where the entire utterance duration is used for calculating the features.

Based on their parametric structure, two classes are used in the SER community:

- *Low-level descriptors (LLDs):* These are features extracted from every short-segment and help modelling temporal and spectral information.
- *High-level statistical functions (HSFs):* Also known as **functionals**, these include statistical features that derive from LLDs, like maximum, minimum, and mean values. These capture the dynamics among frames.

Functionals include statistical parameters from LLDs and are therefore considered suprasegmental features. As reviewed in [69], researchers prefer the use of suprasegmental features as input vectors for their SER systems because they have proven to be more effective in identifying emotions than segmental features. Segmental features are converted into suprasegmental forms through statistical processing.

The classification by acoustic properties is more complex, with no common standard. Researchers define their groups of features and the features to be included in each group [77, 69, 71, 79, 80]. Nevertheless, efforts have been made to set a benchmark for feature selection and extraction for SER [77, 62]. In [62], they present a set of 6373 acoustic features, and these are classified into five general groups: prosodic, cepstral, spectral, sound quality, and temporal features. This set, referred to as the *ComParE 2013 feature set*, has been used in multiple SER studies [81, 63, 5, 66]. The set is composed of functionals extracted from 65 LLD parameters. Depending on the parameter, either 39, 46, or 54 statistical functionals are obtained. Among the functionals obtained are: mean values, root-quadratic mean, flatness, standard deviation, skewness, kurtosis, temporal centroid, percentiles, quartiles, peaks and valleys, mean of peak amplitudes, minimum, maximum, rising slopes, and falling slopes. In the following, a general description of the features is presented according to the group they are classified into.

1. **Cepstral features:** These are derived from the spectrum of the speech signal and include Mel Frequency Cepstral Coefficients (MFCC).

- **MFCC 1-14**: MFCCs capture the vocal tract characteristics. They represent the short-term power spectrum of a sound signal, capturing the essential characteristics of speech. MFCCs are obtained by taking the Fourier transform of a windowed signal, mapping the powers of the spectrum onto the mel scale, and then taking the logarithm and applying the discrete cosine transform. The mel scale is designed to approximate the human ear's response to different frequencies, making it more suitable for audio processing tasks. This process results in 14 coefficients per frame that effectively represent the timbral aspects of the audio signal.

2. **Spectral Features:** These are obtained by converting the time-domain signal to the frequency domain, representing the distribution of energy across different frequencies.

    - **RASTA-Style Filtered Audio Spectrum, bands 1-26 (0 - 8kHz)**: These features involve dividing the audio spectrum into 26 bands within the frequency range of 0 to 8 kHz and applying RASTA filtering (Relative Spectral Transform) to each band separately. The result is a set of values, one for each band, representing the energy in each frequency band after RASTA filtering. RASTA filtering aims to reduce the impact of short-term noise variations and remove constant spectral coloration, such as that caused by different recording environments or channels.

    - **Spectral Energy 250 - 650 Hz, 1k - 4kHz**: Spectral energy refers to the amount of energy present in specific frequency bands of an audio signal.

    - **Spectral Psychoacoustic Sharpness**: Psychoacoustic sharpness is a measure of the perceived sharpness or brightness of a sound. It is influenced by the distribution of spectral energy, particularly the presence of high-frequency components.

    - **Spectral Centroid**: Measures the "center of mass" of the spectrum, calculated as the weighted mean of the frequencies present in the signal.

    - **Spectral Entropy**: Provides a measure of how spread out the spectral energy is.

    - **Spectral Flux**: The difference of the spectra of two consecutive frames, indicating voice dynamics.

    - **Spectral Harmonicity**: The amount of spectral power concentrated at harmonic frequencies (multiples of the fundamental frequency).

    - **Spectral Kurtosis**: Measures how "peaked" or "flat" the spectral distribution is.

- **Spectral Roll Off Point (0.25, 0.50, 0.75, 0.90)**: The spectral roll-off point is a measure that indicates the frequency below which a certain percentage of the total spectral energy is contained (25, 50, 75, 90%). For example, the 0.25 roll-off point is the frequency below which 25% of the spectral energy is found.
- **Spectral Skewness**: Measures the asymmetry of the spectral distribution around the centroid.
- **Spectral Slope**: Linear regression slope of the logarithmic power spectrum within two given bands.
- **Spectral Variance**: Measures the spread of the spectral energy around the mean frequency.

3. Prosodic features: These features represent aspects of speech that relate to its rhythm, melody, and intonation.

   - **Sum of RASTA-Style Filtered Audio Spectrum**: This feature involves applying RASTA filtering to the audio spectrum and then summing the filtered values. The sum provides a single value representing the overall energy in the filtered spectrum.
   - **Sum of Auditory Spectrum (Loudness)**: This represents the overall loudness of an audio signal, which is calculated by summing the energy across the entire auditory spectrum. Loudness is a perceptual measure that correlates with the human experience of how “loud” a sound is, and it is influenced by both the intensity and the frequency content of the sound.
   - **RMS Energy**: Root Mean Square Energy is a measure of the average power or loudness of an audio signal over time. It is calculated by squaring the amplitude values, averaging them over a specified period, and then taking the square root of the result.
   - **Zero-Crossing Rate**: Represents the rate at which a signal changes from positive to negative or vice versa. It is particularly useful for distinguishing between voiced and unvoiced speech sounds.
   - **Fundamental Frequency (F_0)**: The fundamental frequency of the voice, also denominated pitch. It is determined by the rate of vibration of the vocal folds.

4. *Sound Quality Features:* These features assess the overall quality of the speech signal, including aspects like clarity and noisiness.

   - **Jitter Delta, Jitter Local**: Jitter Delta measures the variation in the fundamental frequency from one cycle to the next. Jitter local measures the short-term variations in the fundamental frequency. It is calculated as the average absolute difference between consecutive periods, normalized by the average period length.

- **Log HNR**: Logarithm of Harmonics-to-Noise Ratio is a measure of the harmonicity of a signal, indicating the ratio of periodic (harmonic) components to noise. It is calculated using the logarithm of the ratio of the energy of the harmonics to the energy of the noise.
- **Shimmer Local**: This measures the short-term variations in the amplitude of the signal between pitch periods. It is calculated as the average absolute difference between the amplitudes of consecutive periods, normalized by the average amplitude.
- **Probability of Voicing**: This indicates the likelihood that a given segment of the audio signal contains voiced speech. It is calculated using features such as the fundamental frequency and energy levels.

5. *Temporal Features:* These features capture the timing aspects of speech, such as the rate of speech and pauses.
   - **Ratio of non-zero F0 values**: Gives an estimate of how much of the signal is voiced.
   - **Mean length of voiced segments**: The average duration of segments in the audio signal that are classified as voiced.
   - **Max length of voiced segments**: The longest duration of a continuous voiced segment in the audio signal. It provides insights into the longest uninterrupted speech segments.
   - **Min length of voiced segments**: This is the shortest duration of a continuous voiced segment in the audio signal.
   - **Standard deviation of the voiced segments length**: This measures the variability in the lengths of voiced segments. A higher standard deviation indicates more variability in the duration of voiced segments.

In Table 2.3.1.2, the number of features is broken down by group. The LLDs and their first-order delta coefficients (differential) were smoothed by a moving average filter.

The ComParE feature set encompass parameters that are frequently analyzed in SER works. However, it is important to recall that each author organizes and selects the features according to their own considerations, this could mean choosing only one group of features, merging groups, removing or adding features to the groups, or even using other categories [77, 69, 71, 79, 80, 26]. As mentioned in [82], “the golden set of acoustic features has yet to be found, or simply does not exist.” Some authors use several hundred features per utterance, while others use only a few, making it difficult to compare results among SER works.

Hand-crafted acoustic features can be manually measured from audio signals, nevertheless, the use of automatic feature extraction tools is widely supported by research in SER. Tools like the open-source Speech and Music Interpretation by Large-space Extraction (OpenSMILE) toolkit [83], PRAAT Software [84], and the Python package Librosa [85] are commonly used for extracting acoustic features for SER. Automatic feature extraction ensures the consistency of the parameters extracted, allowing reproducibility. Besides, they can extract a wide range of features that are crucial for emotion recognition and help in handling large datasets [77].

In recent years, DL has enabled the automatic learning of hierarchical feature representations directly from raw or minimally processed input data. These learned representations, often referred to as *deep representations*, are optimized end-to-end to improve task performance, contrasting with traditional hand-crafted features.

DNNs, especially convolutional and recurrent architectures, can learn multi-layered representations of the input data. Each layer transforms its input into a more abstract representation. These deep representations enhance the model's ability to capture intricate patterns in the data [86, 18, 87, 88]. However, these representations can be difficult to interpret and understand. Unlike hand-crafted features, which are designed based on domain knowledge, deep representations are often seen as “black boxes” [89].

This study focuses on the analysis of hand-crafted features. Although deep representations have demonstrated significant advancements in various fields, the research aims to explore the effectiveness and nuances of traditional acoustic features.

Table 2.3: Groups of Features of the ComParE 2013 feature set. Δ represents the first-order derivatives. Adapted from [63].

Groups of features	**Features**	**Functionals**	**Subtotal**	**Total of features**
Cepstral features	MFCC 1 – 14	54	14*54 =756	1400
	ΔMFCC 1 - 14	46	14*46 =644	
Prosodic features	Sum of RASTA-Style Filtered Auditory Spectrum	54	1*54 =54	478
	ΔSum of RASTA-Style Filtered Auditory Spectrum	46	1*46 =46	
	Sum of Auditory Spectrum (Loudness)	54	1*54 =54	
	ΔSum of Auditory Spectrum (Loudness)	46	1*46 =46	
	RMS Energy	54	1*54 =54	
	ΔRMS Energy	46	1*46 =46	
	Zero-Crossing Rate	54	1*54 =54	
	ΔZero-Crossing Rate	46	1*46 =46	
	F0 (Fundamental Frequency)	39	1*39 =39	
	ΔF0 (Fundamental Frequency)	39	1*39 =39	
Sound Quality features	Jitter Delta, Jitter Local	39	2*39 =78	390
	ΔJitter Delta, ΔJitter Local	39	2*39 =78	
	Log HNR (Harmonic-to-Noise Ratio)	39	1*39 =39	
	ΔLog HNR (Harmonic-to-Noise Ratio)	39	1*39 =39	
	Shimmer (Local)	39	1*39 =39	
	ΔShimmer (Local)	39	1*39 =39	
	Probability of Voicing	39	1*39 =39	
	ΔProbability of Voicing	39	1*39 =39	
Spectral features	RASTA-Style Auditory Spectrum, Bands 1-26 (0-8 kHz)	54	26*54 =1404	4100
	ΔRASTA-Style Auditory Spectrum, Bands 1-26 (0-8 kHz)	46	26*46 =1196	
	Spectral Energy (250-650 Hz), (1k-4kHz)	54	2*54 =108	
	ΔSpectral Energy (250-650 Hz), (1k-4kHz)	46	2*46 =92	
	Spectral Psychoacoustic Sharpness	54	1*54 =54	
	ΔSpectral Psychoacoustic Sharpness	46	1*46 =46	
	Spectral Centroid	54	1*54 =54	
	ΔSpectral Centroid	46	1*46 =46	
	Spectral Entropy	54	1*54 =54	
	ΔSpectral Entropy	46	1*46 =46	
	Spectral Flux	54	1*54 =54	
	ΔSpectral Flux	46	1*46 =46	
	Spectral Harmonicity	54	1*54 =54	
	ΔSpectral Harmonicity	46	1*46 =46	
	Spectral Kurtosis	54	1*54 =54	
	ΔSpectral Kurtosis	46	1*46 =46	
	Spectral Roll Off Point (0.25), (0.5), (0.75), (0.90)	54	4*54 =216	
	ΔSpectral Roll Off Point (0.25), (0.5), (0.75), (0.90)	46	4*46 =184	
	Spectral Skewness	54	1*54 =54	
	ΔSpectral Skewness	46	1*46 =46	
	Spectral Slope	54	1*54 =54	
	ΔSpectral Slope	46	1*46 =46	
	Spectral Variance	54	1*54 =54	
	ΔSpectral Variance	46	1*46 =46	
Temporal features	Ratio of non-zero F0 values	1	1*1 =1	5
	Mean length of the voiced segments	1	1*1 =1	
	Max length of the voice segments	1	1*1 =1	
	Min length of the voiced segments	1	1*1 =1	
	Standard deviation of the voiced segment length	1	1*1 =1	

2.3.1.3. Emotion classification from speech

Multiple approaches have been developed for automatic ER from speech at the individual level. Traditionally, conventional ML methods have been employed, but DL approaches are becoming increasingly popular. According to the review made by Atmaja et al. [48], four of the most common SER classifiers are the SVM, MLP, CNN, and LSTM NN. In Section 2.7, some of the most relevant studies on SER using those classifiers are described.

2.4. Group-level emotion recognition

Recognizing emotions at the group level presents significantly more complexity than individual emotion recognition. The concept of a *group* can vary widely, ranging from small gatherings to large crowds, raising the fundamental question: does a *group emotion* truly exist?

Multiple studies in psychology dive deep into how the emotions of the individuals can affect the emotions of the people around them, creating a concept of a collective emotion [90, 91, 92, 93]. For example, in a football game, the euphoria and feelings of the spectators are affected collectively depending on the performance of the teams they are supporting.

In the definition of group emotion, a generally accepted view is that group emotion can be analyzed from a *bottom-up approach*, and *top-down approach* [94]. The bottom-up approach suggests that the group emotion originates from the individuals and their emotions, while the top-down approach highlights that the group emotion is exhibited at a group level and is felt by the individual members. In effect, group emotion encompasses both the emotions felt by the individual members and group-level factors that shape the emotional state of the group.

In computational affective analysis, researchers have adopted the bottom-up and top-down approaches to develop methods for automatic GER [19, 18].

- **Bottom-up approach (individual context)**: These methods focus on analyzing the individual emotional expressions of the group members and aggregate them to infer the overall group emotion.
- **Top-down approach (group context)**: These methods treat the group as a homogeneous entity, using contextual and environmental cues to determine the group's emotional state, independent of individual expressions.

The goal of GER is to classify group emotions as precisely as possible from data. With the introduction of challenges as the "Emotion Recognition In The Wild Challenge (EmotiW)", the area of automatic group-emotion recognition has gained attention among researchers and has been boosted with the advancement of ML algorithms [18]. Examples of these are introduced in Section 2.7 "State-of-the-art". This placement is intentional to provide a comprehensive overview of the latest advancements and methodologies in the field.

Despite growing research interest, there is a lack of databases specifically designed for group emotion recognition, especially when compared to individual-level SER datasets. Typically, "in-the-wild" data is used for building group emotion databases, which has the main advantage of representing real-life group interactions. However, this type of data requires annotators, who give the emotion labels based on their perceived emotion, which introduces a degree of subjectivity.Group emotion annotations, with few exceptions, are typically categorized along coarse valence-based labels: *Positive*, *Neutral*, and *Negative* [19].

The majority of group emotion datasets are based on images or videos, but other types of databases also exist that employ other data modalities. Image-based datasets often consist of photos collected from online sources such as Flickr[3] or Google Images. Examples of them are the GAFF 2.0 [95], GAFF 3.0 [96], HAPPEI [97] and GroupEmoW [13] databases. Video-based datasets include recordings in both controlled environments and real-world scenarios. The Video Group Affect database (VGAF), for instance, consists of videos collected from YouTube of small groups and large crowds interactions [50]. With the popularity of social media, text-based group emotion datasets have also emerged, in which posts are labeled based on the emotions they convey. Audi-only group emotion databases are rare. According to a review made in 2023 by Veltmeijer et al. [19], only two audio-only GER databases were found; these were built using audio clips from cheering (Joy), rioting (Anger), and background noise (Neutral). An even rarer type of database is databases using physiological data, like the database presented in [98], where photoplethysmography and electrodermal activity are reported for each individual. These types of databases are even more challenging to create, but they have the advantage that the annotation of emotion labels is self-reported.

The review of GER methods made by Veltmeijer et al. [19] highlights three main challenges faced by GER systems. 1) The accurate emotion detection is complex and can be biased by subjective labeling. 2) The varying number of individuals and diverse scenes require robust and generalized features, and 3) integrating diverse features in multimodal models and achieving end-to-end learning is difficult. Furthermore, they mention the importance of using different metrics, rather than accuracy, for evaluating GER systems. Since accuracy can be highly influenced by imbalanced datasets, they encourage the use of metrics such as UAR and F1-score that offer a more comprehensive evaluation of the GER models.

According to Lee and Kim [6], GER can be used in real-life applications such as the monitoring of crowds, surveillance, and participation analysis of students in online classes. They suggest that future research on GER should address several key areas: Firstly, methods need to be developed that are flexible in analyzing varying group sizes, such as networks that automatically detect and adjust to group size. Secondly, the presence of different emotional subgroups within a group should be investigated, focusing on detecting and analyzing subgroups with emotions that deviate from the average. Thirdly, incorporating temporal analysis to track changing emotions can help predict emotional patterns and train security personnel in simulations. Lastly, enhancing robustness to non-emotional data variations and exploring hybrid approaches combining multiple data modalities, like images and audio, is crucial. They final-

[3] Flickr is an online photo management and sharing application, known for its large collection of Creative Commons-licensed images. `https://www.flickr.com/`

ize by mentioning that creating unbiased datasets is vital for developing real-world, applicable frameworks.

2.5. Speech separation

Humans naturally excel at tracing, segregating, and recognizing the speech of interest in such environments, a phenomenon known as selective hearing [99]. This ability allows them to focus on a specific speech source while ignoring noise and interfering speakers. However, achieving similar capabilities in machines remains a significant challenge. Speech separation is the process of isolating individual speech signals from a mixture of multiple speakers, commonly referred to as the *"cocktail-party problem"*. This problem arises in real-world communications where multiple speakers talk simultaneously, accompanied by other sounds, background noises and reverberation, as well. The cocktail-party problem was first introduced by Cherry [100] and has since been a focal point of research in the speech processing domain [99, 101, 102, 103].

Speech separation is categorized based on the number of channels or microphones used for recording the audio signal. Single-channel speech separation is more challenging compared to multi-channel. Multi-channel approaches can exploit spatial information from multiple microphones to distinguish between different sound sources based on their locations. This spatial information helps in separating the sources more effectively [104]. Single-channel speech separation relies solely on spectral aspects of speech, such as pitch continuity, harmonicity, common onsets, etc. [104]. Figure 2.14 illustrates the task of a single-channel speech separation process.

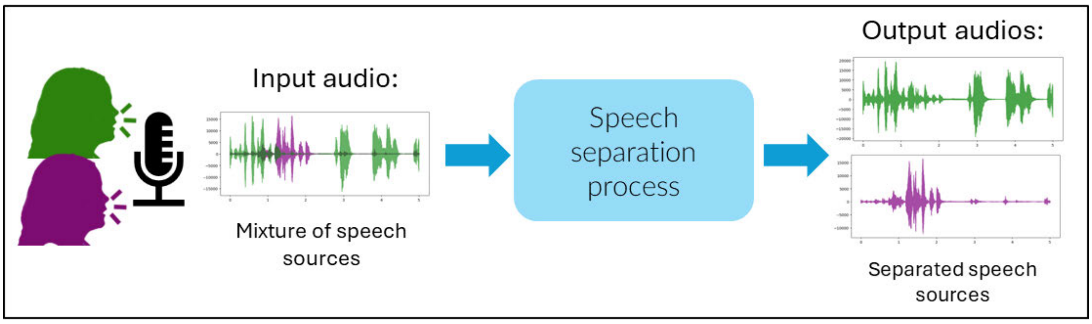


Figure 2.14: Visualization of speech separation on a mixture of two speech sources.

There are two dominant methods for approaching the cocktail party problem: blind source separation (BSS) and target speaker extraction (TSE). The term "blind" means there is no available information about the mixing function of the signals or source signals. The purpose of BSS is to separate all sources in a mixture in one step, while TSE aims to extract the speech signal of a target speaker only, removing the rest of the interferences [105].

DNNs are at the forefront of speech separation and can be broadly categorized into time-frequency domain approaches and time-domain approaches. Time-frequency approaches convert the mixture waveform into the time-frequency domain using the short-time Fourier transform (STFT), separate time-frequency features for each source, and reconstruct the source waveforms by applying the inverse STFT. Time-domain approaches perform end-to-end separation by directly modeling the mixture waveform with an encoder-decoder framework [106]. End-to-end models are particularly beneficial for tasks where the main focus is not speech separation, but can significantly benefit from it, like in speech emotion recognition.

Separating speech is essential for applications such as speech recognition, speaker identification, and audio processing. Since overlapping speech frequently happens when speakers are in intense emotional states, such as during a heated argument or an enthusiastic conversation, where the usual practice of taking turns is disrupted, speech separation is also needed in GER [103]. Implementing speech separation as a preprocessing step for speech-based GER is beneficial for isolating the speech of the individual speakers, and in this way, features from the individual speakers can be extracted for the GER.

2.6. Speaker diarization

Speaker diarization is the process of identifying “who spoke when” in multi-speaker audio data. The term “diarize” means to record events in a diary. Similarly, speaker diarization involves recording or logging speaker-specific information within an audio signal. This process can determine not only the number of speakers in an audio signal, but also measures the duration of each speaker’s speech, the start and end times of a speaker’s speech segment, and determines the order in which they speak. Figure 2.15 visualizes an example of speaker diarization, where three speakers have been identified in a raw audio signal.

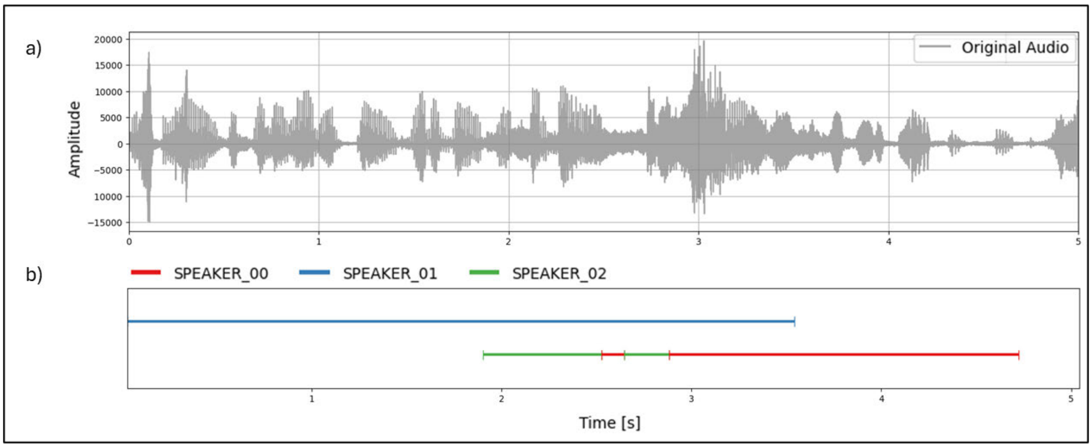


Figure 2.15: Example of speaker diarization in a speech signal with three speakers a. Original audio signal. b. Output of the speaker diarization process (speech segments are identified for each speaker)

Typically, speaker diarization systems are constructed using a combination of various submodules or building blocks, where each one is responsible for executing different tasks. These tasks include, but are not limited to, denoising the signal, speech enhancement, voice activity detection (VAD), segmentation of the audio signal, separation of overlapped segments, clustering of the segments and identification of the speakers. Classical systems involve separate modules that are trained individually. However, in modern approaches, end-to-end solutions are proposed using advanced DL algorithms, where all individual submodules are replaced by one DNN. End-to-end approaches enable global optimization and reduce cascading errors, resulting in robustness against noise and overlapping speech [35].

2.7. State of the art

This section reviews the current advancements and methodologies in the field of speech emotion recognition, with a particular focus on group emotion recognition, the influence of acoustic features, and speech separation techniques. The review is divided into three subsections, each addressing a critical aspect of the research landscape.

2.7.1. Speech emotion recognition

Most research in SER has been conducted at the individual level. Therefore, it is appropriate to introduce studies that have made use of common classifiers for SER, such as SVMs and NNs [107, 48].

The most widely used classifier for SER tasks is the SVM. It has been employed for categorical emotion recognition, as well as for dimensional ER [77, 107, 48]. Gao et al. [108] uses a linear kernel SVM. They compute pitch, intensity, MFCCs, linear spectral pairs, and zero-crossing rate as acoustic features over segments of 20-100 ms. After smoothing and normalization, functionals are extracted and fed to the SVM model, and concluded that the SVM model and the features extracted were enough in effectively characterizing and recognizing seven individual emotions (Happy, Boredom, Disgust, Fear, Angry, Neutral, and Sad). Dahake et al. [26] compared several kernel functions to classify emotion utterances. They conclude that the RBF kernel achieves the best overall recognition rate, while the polynomial kernel offers the worst result. M S et al. [109] use a combination of MFCCs, pitch, and energy-based features to classify emotions with a linear kernel SVM, achieving an overall accuracy of 95.83% on their self-created Malayalam emotional database (four classes: Anger, Happy, Neutral, Sad) and total 75% accuracy on the Emo-DB database[110] (classes: Anger, Happy, Neutral Sad). Chen et al. [111] compared an SVM against an MLP. They extract energy and spectral LLDs from the utterances, compute the first and second derivatives of the LLDs, and extract functionals, obtaining 288 features. These features are later reduced via dimensionality reduction methods such as principal component analysis (PCA) and linear discriminant analysis (LDA). They achieve better general performance when using LDA with SVM.

NNs are powerful for combining acoustic and linguistic information for SER [48]. In [4], different NN architectures were tested for classifying emotions. The maximum performance was achieved with an MLP with the following architecture: 33 input neurons (equaling the number of input features), a sigmoid transfer function in the hidden layer, and seven output neurons (one for each emotion: Anger, Disgust, Fear, Joy, Neutral, Sadness, and Surprise), using statistical functionals from the pitch, energy, spectral, and temporal contours. In the study by Getahun and Kebede [112], telephonic conversations recorded in a call center are analyzed for detecting the emotion of each speaker. Their attempt to separate the mixture of speech signals consisted of using a software tool for detecting speaker turns in the dialogues and isolating the segments of speech for each speaker. However, they do not mention separating overlapped speech. They further segmented the speech signal into chunks 1 to 15 seconds long, and each chunk got an emotion label among Anger, Fear, Positive, and Sad. They extracted a total of 170 acoustic features at a chunk level, consisting of prosodic, spectral, and voice quality features. From these, they selected 33 to be fed into an MLP for the SER task. They obtained an average accuracy of 73.4% in classifying the four emotions.

The study by Yenigalla et al. [113] experimented with phonemes (distinct units of sound in a language, such as vowels and consonants) and spectrograms as input parameters to a CNN model for categorical Emotion Recognition (six classes). The combination of both features achieved higher performance than processing each alone.

LSTM networks have dominated the DL classifiers used in ASR and SER [48]. LSTMs are able to map LLDs with emotion labels, as the data is sequenced. The work of Tian et al. [114] uses a combined approach with LSTM and SVM, where both models are trained separately on LLDs and disfluencies, and non-verbal vocalizations. They used a dimensional emotion model consisting of four dimensions: Arousal, Expectancy, Power, and Valence. They tested both models on the IEMOCAP and AVEC2012 databases, with acted speech and spontaneous speech, respectively. The SVM with the LLD achieved the highest weighted-averaged F1-score on the IEMOCAP database (57.5%), while the LSTM achieved the best results on the AVEC2012 database using the disfluences and non-verbal vocalizations as features (weighted-averaged F1-score of 62.0%).

2.7.2. Approaches to group emotion recognition

As reviewed in Section 2.4, in GER analysis, researchers often follow a top-down or a bottom-up approach. There is another approach referred to as *hybrid approach* that combine bottom-up and top-down approaches [19, 18]. This subsection explores the different and recent approaches in GER, highlighting the techniques and models developed to understand and recognize the emotional states of groups.

2.7.2.1. Bottom-up

Bottom-up approaches recognize group emotion from features extracted at the individual level. Multiple studies have focused on this type of approach, primarily using visual features. For instance, Lu and Zhang [115] employed CNNs to recognize the emotions from the individuals' faces, which were then fused to create a group emotion prediction.

In [17], they analysed the emotion of a group of four people while playing a board game. They attached microphones to each of them and recorded their facial expressions, and used them for extracting visual features. From the speech signals, they extracted prosodic information as acoustic features. Since every speaker had their microphone, no speech separation was applied. They derived the group emotion from the individual emotions with a Bayesian network, which is a probabilistic graphical model that represents a set of random variables and their conditional dependencies. They classified three emotions: Normal, Smile, and Surprise. With their method, they achieved an overall average recognition rate of 60.3%.

2.7.2.2. Top-down

Top-down approaches utilize features extracted at a group context for recognizing the group emotion. These type of approaches have been explored for both visual and audio data. In [50], which also introduced the VGAF database, group-context acoustic and visual features are analysed from the videos to predict the group emotion as Positive, Negative, or Neutral. The raw audio signals were utilized to extract the ComParE 2013 acoustic features, which were processed with an FCNN. For the video features, a pre-trained model on images was used to obtain deep representations that were fed into an LSTM network for video inference. The outputs of the FCNN and LSTM networks were concatenated, and the prediction of group emotion was done over the fused features by using softmax and sigmoid activation for classification. The acoustic features alone achieved an overall accuracy of 48% across the three classes on the test data. The results achieved by the fusion of acoustic and video features reduced the accuracy by 0.5%, while the use of visual features alone resulted in a lower accuracy of 42%.

Another top-down approach was done by Petrova et al. [8] where video data is used. The videos were sampled into 10 thousand frames, from these frames (images) the visual features were extracted and used for training different CNN models for recognizing Positive, Negative, and Neutral group emotions. Their approach was self-denominated as "Privacy-safe" since the features are not extracted at the individual level. They experimented with different CNN architectures and achieved the highest scores using a VGG-19 model [116], enhanced with several fully-connected layers, dropout layers, and ReLU activation functions. An accuracy of 59.13% was achieved across the three classes on the test subset of the VGAF database.

Another top-down approach was introduced by Wang et al. [117], who developed a model called the K-injection audiovisual network. This model employs a multi-head cross-attention mechanism to jointly model audio and video data, integrating the description of the video context as a linguistic feature to improve generalization. From the audio signals, they analyzed Mel-spectrograms, MFCCs, pitch, and energy. An overall accuracy of 66.40% was achieved on the test subset of the VGAF database, classifying emotions into three categories: Positive, Negative, and Neutral.

Ottl et al. [14] conducted another top-down approach using speech data. They used various CNN architectures, referred to as Deep Spectrum nets, to extract deep representations from Mel-spectrograms of the audio signals from the VGAF database. These representations were fed into a classifier for group emotion consisting of an NN optimized with stochastic gradient descent. They also extracted the ComParE 2013 feature set from the signals and fed it to the same classifier for comparison. Additionally, another NN was trained with the fusion of the deep representations and ComParE features. They reported a UAR of 53.43% and an accuracy of 52.48% over the three emotion classes (Positive, Neutral and Negative) using only the ComParE 2013 features. And, with the fusion of features, they achieved a UAR of 60.91% and an accuracy of 59.40%.

2.7.2.3. Hybrid

Hybrid approaches combine features extracted at both individual and group contexts for GER. These result from the combination of top-down and bottom-up approaches. In [118], the emotions of students in a classroom are analysed. A multimodal approach was employed, using an LSTM trained on acoustic features, an SVM trained on linguistic parameters, and a CNN model trained on visual information. A decision-level fusion algorithm was applied on the three different modal schemes to integrate the classification results and deduce the overall group emotions of the students. The acoustic and linguistic features were obtained from mixtures of speech signals. The fusion of all modalities achieved an overall accuracy of 76.2% on the seven emotions classified (Happy, Anger, Disgust, Fear, Surprise, Neutral, Sad). While the speech modality alone achieved an averaged accuracy of 77.6%.

A hybrid approach was explored by Sun et al. [15], who tested several CNN models trained with different types of features, mainly global and local representations from image and video data from the VGAF database. Global representations refer to features extracted by processing the whole file, while local representations are features extracted from only a portion/section of the file. In addition, they trained an LSTM model using only audio features for comparison. As audio features, they used the INTERSPEECH 2010 Challenge feature set [119] that contains 1582 features. The audio model alone achieved an average accuracy of 56% on their validation dataset over the three classes (Positive, Negative, Neutral). By fusing the outputs of all models using grid search, they achieved an accuracy of 71.93% on the same validation dataset.

Another hybrid approach was conducted by Wang et al. [11]. Using a group emotion image database, they analyzed three types of features for group emotion recognition: facial expressions, body postures, and global image features. They employed multiple CNNs trained on the images, and the scores of all models were averaged to estimate the group emotion in the image as either Positive, Negative, or Neutral, achieving an overall accuracy of 67.48% on the test set. Similarly, Quach et al. [9] and Favaretto et al. [120] studied group emotion recognition from video data, focusing on crowds. These studies did not use audio signals as a source of features. Instead, both approaches utilized temporal information between video frames, as well as global and local representations from still images, corresponding to the entire video frame and the individual faces, respectively.

Wang et al. [121] proposed a context-consistent cross-graph NN (ConGNN) for "in-the-wild" group emotion recognition from images. Extracting facial information, local features, and global scene features (whole image) to form multi-cue emotion feature sets. They tested their ConGNN model on the GroupEmoW dataset, achieving accuracies of 72.09% for the Negative class, 80.69% for the Neutral, and 88.37% for the Positive class. They suggest that Neutral samples are difficult to distinguish from Negative samples.

In summary, the state of the art in GER highlights the techniques that have been applied for the recognition of group emotion. Hybrid approaches are the most commonly explored, using facial expressions as individual-context features and the whole image to extract the group-context features. In most cases where audio data was used, the audio signal was processed as a raw audio file without employing any speech detection or speech separation methods. Multi-modal data approaches have demonstrated significant potential and, in many instances, achieved the highest accuracies. Nonetheless, the use of acoustic features alone has also proven satisfactory. SVMs are the most commonly used non-DL method for SER at the individual level and have been tested against MLPs and LSTM NN, outperforming them in certain cases. DNNs are the preferred choice among researchers in GER, with satisfactory results, particularly in the image modality. However, the scores reported for "in-the-wild" video data typically do not exceed 70%.

2.7.3. Analysis of the influence of acoustic features for emotion recognition

Some studies have explored the contribution of hand-crafted acoustic features to the recognition of emotion from speech at the individual level. However, similar studies for group-level SER have not been found to date.

The work of [78] aimed to identify the most relevant acoustic features for the classification of each emotion in SER (Anger, Happiness, Sadness, Boredom, Anxiety, and Neutral) at the individual level. They conducted an incremental analysis of different acoustic feature groups (pitch, energy, duration, articulation, voice quality, and zero-crossing rate) by concatenating them to determine which group of features contributed the most to emotion recognition. Their results indicate that, on average, pitch features, when supported by voice quality features, achieved a 74.1% average recognition rate on the six classes. Pitch features alone reached a 62.1% average recognition rate, categorizing them as the most relevant feature group. A Bayesian classifier was used for the emotion recognition.

Schuller et al. [4] employed Linear Discriminant Analysis to rank various suprasegmental acoustic features based on their quantitative contribution to emotion recognition.

The study highlights that spectral features are significantly dependent on phonemes and, consequently, on the phonetic content of an utterance. This dependency contradicts the objective of achieving independence from the spoken content in acoustic analysis. Ideally, acoustic analysis should focus on features that remain consistent regardless of the specific words or phonemes being spoken, enabling a more generalized understanding of speech patterns. To address this issue, the researchers limited their study to spectral characteristics between 250 Hz and 650 Hz. Their ranking revealed that combining all pitch features resulted in 69.81% averaged accuracy on the seven emotions (Anger, Disgust, Fear, Joy, Neutral, Sadness, and Surprise), whereas utilizing all energy-related features achieved 36.58% averaged accuracy.

From both studies, it can be inferred that from all acoustic features, pitch information plays the most important role in SER. However, it is important to note that both studies made use of acted speech databases with recording of individual speakers, which means no real-life conditions and low speaker variability.

2.7.4. Speech separation in emotion recognition

This section describes two approaches that investigated the effect of speech separation on SER. Although both approaches focused on SER at the individual level, their methods of separating emotional speech are relevant to this study.

In [102], the authors extracted the speech of a target speaker from a mixture of speech sources and used the separated signal for SER training. They mixed audio files from individual speakers of different emotion categories from one dataset with random utterances from a different speech dataset (with no emotion label) considered as noise. The number of speakers in the mixture was two, and the speaker whose speech was labeled with emotion was targeted. They employed a pre-trained TSE model for the speech separation, and for SER they used a ShiftCNN model [122]. Two SER emotion classifiers were trained with clean and noisy speech separately. The effect of the TSE method on SER was analyzed using weighted (WA) and unweighted accuracy (UA) as metrics. They compared the performance of the SER classifiers with and without the TSE as a preprocessing step for an input signal, and found that both metrics were worse when the TSE model was used for denoising the signals. As a solution, they proposed a new SER model using ShiftNN but trained with denoised signals. This model outperformed by almost 10% the UA scores obtained by the models that were trained on noisy signals on the four classes recognized (Happy-Excitement, Angry, Sad, and Neutral). They also found that the performance of the TSE models was higher when the speakers in the noisy signal were of different genders, highlighting the gap among TSE-SER-based systems dealing with same- and different-gender mixtures.

The study by [123] addresses the degradation of accuracy in SER using single-channel overlapped speeches compared to the use of clean speech. First, an end-to-end DL speech separation model was trained with mixtures of speeches that were randomly selected from a speech database. The different mixtures of speech from individual speakers labeled as Happy, Neutral, or Angry were created. These mixtures contained only two speakers and combinations of emotions such as Happy-Neutral, Happy-Angry, and Neutral-Angry. The mixtures were fed to the pre-trained model to be separated. Then, the separated audio outputs of the model were fed to a self-supervised pre-trained model (Wav2vec-2.0 [124]) to perform emotion recognition and speech recognition. This pre-trained model uses a supervised method to train a large amount of data to learn the representation of speech. The accuracy rate of the SER model on non-separated audio files was very low (33%). It is not clear if only one or both emotions were recognized (recalling that each mixture was a combination of two emotions). Nevertheless, the reported accuracy rate was 52% on the emotion recognition after the separation process. The study concluded that speech separation experiments can be more integrated with speech emotion recognition and encouraged the creation of end-to-end speech separation-emotion recognition models.

Both studies highlighted how overlapped speech degrades the accuracy of SER systems and demonstrated that using speech separation to extract emotional speech can help overcome this limitation. This technique can also be advantageous for group-level SER and is therefore explored in the current study.

Interestingly (or surprisingly), another two studies were found that analyzed the opposite: the effect of emotional speech on speech separation algorithms, instead of analyzing the influence of speech separation on emotion recognition. Since most speech separation models are trained on neutral speech, they hypothesized that the performance of the models would be affected when dealing with strong emotional speech.

In [103], using custom datasets with mixtures of emotional speech, they tested a state-of-the-art speech separation model. They concluded that emotions do result in performance degradation when the mixture contains strong emotional expressions. In [105], it was observed that BSS algorithms are relatively robust to emotional speech, while TSE, which requires identifying and extracting the speech of a target speaker, is much more sensitive to emotions.

The findings of these two studies suggest that the performance of speech separation pre-trained models for separating emotional speech may not be as optimal as for neutral speech. However, the impact is less pronounced in BSS algorithms compared to TSE algorithms.

3 Methods

This work focuses on analyzing the influence of different groups of acoustic features of the individual speakers on the recognition of the group emotion.

The approach begins with the preparation of data, which includes extracting the audio signal from the video files of the database and identifying the number of speakers in each audio signal, and keeping the audio files containing speech mixtures of two speakers to build a new database for GER on groups of two speakers. Next, speech separation is performed to isolate each of the individual speakers within the speech mixtures. Then, acoustic features are extracted from the separated speech (individual features) and the speech mixtures (mixture features), and are combined strategically to develop different feature sets for training an SVM and an FCNN model for GER. Each model is first trained using only mixture features as input, and their performance is assessed on two subsets, one speaker-dependent and another speaker-independent, and the results obtained on each subset are used as baselines. Then, six experiments are performed, in each an SVM and an FCNN are trained on different sets composed of individual and mixture features. Figure 3.1 illustrates the steps followed for the recognition of the group emotion in each experiment and baseline. The performance of the models trained on individual and mixture features is hypothesized to overcome the baseline. Finally, a significance test is performed to validate the results of the experiments. In the following sections, every step followed in the methodology is described in detail.

All calculations, the training of the models, and all experiments are conducted under the following hardware/software setup:

- **Processor (CPU)**: Intel Core i7-14700F 2.10 GHz
- **Memory (RAM)**: 32.0 GB 5600 MHz
- **Graphics Card (GPU)**: NVIDIA GeForce RTX 4070 SUPER 12 GB
- **Operating System**: Windows 11 Home 24H2
- **Python Version**: 3.11.0
- **Compiler**: MSC v.1916 64 bit (AMD64)
- **CPU cores**: 28

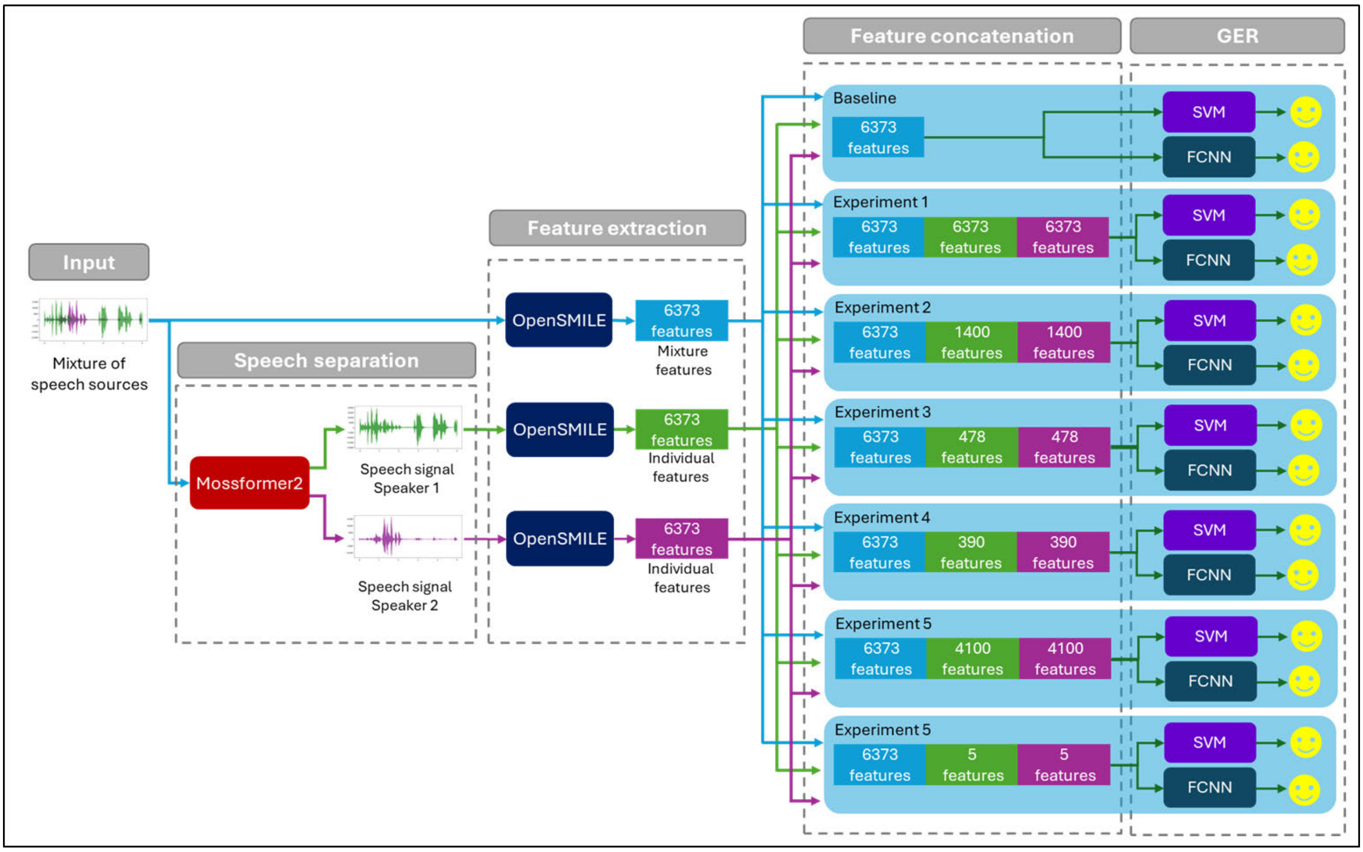


Figure 3.1: Proposed approach for GER. The audio signal containing a mixture of two speeches is processed by a pre-trained model for speech separation to obtain the speech of each speaker. Then, the individual features are extracted from the separated speech signals and the mixture features from the mixture signal. The features extracted are concatenated and used to develop seven different feature sets. In each experiment, an SVM and an FCNN are trained for GER using the corresponding feature set.

3.1. Data preparation

The Video Group Affect database (VGAF) database [50] used in the EmotiW 2020 edition [66] has been selected for the present work due to its widespread use in studies related to GER [117, 8, 14, 7]. The database consists of videos with a Creative Commons license collected from YouTube. The videos reflect groups of people interacting in real life scenarios (”in-the-wild” conditions), and these were found by searching for keywords such as “interview”, “festival”, “party”, “silent protest”, “violence”, “argument”, “birthday”, “wedding”, “meeting”, and “fighting”. Figure 3.2 shows three frames from videos of the VGAF database. Each video is cropped into 5-second segments, and each of the segments is manually labeled by annotators according to the group emotion classified as *Positive*, *Neutral*, or *Negative*, corresponding to the valence axis. The number of video clips (segments) produced by one video varies depending on the duration of the original video. For instance, the longest video produced 128 clips, while some videos produced only one clip.

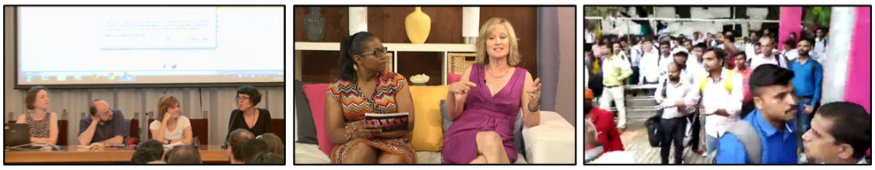


Figure 3.2: Three frames from video clips of the VGAF database [50]. The video clips are given a group emotion label by annotators in the valence emotion dimension. From left to right: Positive, Neutral, and Negative.

Each of the videoclips in the database is named with an identifier of the source video, followed by an identifier for the cropped segment. For example, for a file named “34_5.mp4” the first digits (“34”) correspond to the video it is cropped from, and the last digit (“5”) identifies the segment number. In this case, “34_5” indicates that the file is the fifth segment obtained from “video number 34”.

A total of 3427 labeled video clips are provided by the authors of the database after following the procedure for requesting access. The video files were originally divided into the *Training* subset (2661 video clips) and the *Validation* subset (766 video clips). The video files of the *Test* subset were released only for participants of the 2020 EmotiW challenge.

The database is designed for audio-visual group emotion recognition. However, since the focus of this work is on SER, the visual information is not required. Hence, the next logical step is to extract the audio signals from all video clips. The audio signal from the videos is extracted as WAV file using FFmpeg, a widely used free software tool for multimedia file manipulation [125].

After exploring the database, it is noticed that many audio files do not contain any speech. This is not surprising, due to the videos being collected from “in-the-wild” conditions. For example, some videos were recorded during a boxing match, so the audio signal contains only the crowd's euphoric cheers and whistles. Other videos were recorded during music festivals, hence the audio signal consists solely of background music. Other videos simply depict people remaining silent, making the audio signal not suitable for an SER task. Thus, it is necessary to filter the videos to ensure the analysis of speech signals.

Recalling the purpose of this project, the focus is on analyzing the speech features of individual speakers to determine group emotion. Therefore, it is essential to have a database with audio files that contain the speech of more than one speaker. For detecting the number of speakers on each audio file, an end-to-end speaker diarization pre-trained model is used. Reviewing manually the 3000 audio files one by one to assign a label for the number of speakers would have been a tedious task and would require annotators to do the job. Therefore, it is more efficient to use an existing pre-trained speaker diarization model for this purpose.

As reviewed in section 2.6, end-to-end speaker diarization models take as input the raw audio file and generate as output information about the speakers contained in the audio signal. To find a suitable speaker diarization pre-trained model, first, the *Awesome Speaker Diarization* repository [126] is reviewed. It contains a curated list of relevant speaker diarization studies, libraries, datasets, and other resources from 26 contributors. Although it is not a direct list of pre-trained models, it condenses the information of most open-source frameworks for speaker diarization.

From all frameworks reviewed, only two end-to-end pre-trained models for speaker diarization could be successfully implemented, mainly due to the limitations of the software environment used in this work. The two models are described below.

`pyannote.audio`: This tool can handle multiple speakers and consists of a pipeline of pre-trained models for feature extraction, VAD, overlapped speech detection, speaker change detection, speaker embedding, and clustering [127]. It is based on the PyTorch[1] framework. Version 3.1 is the latest one at the moment of redaction and is hosted on Hugging Face [2]. It has been employed in multiple studies involving speaker diarization [105, 128].

`simple_diarizer`: Consists of a pipeline that contains pre-trained models for voice activity detection, speaker embedding extraction, clustering, and, optionally, speech-to-text translation. It handles multiple speakers [129].

Another search for pre-trained models for speaker diarization is done on Hugging Face. Nevertheless, only one pre-trained is found that could run successfully on the computer system:

[1] PyTorch is a framework for building and deploying AI models https://pytorch.org/

[2] Hugging Face is a popular platform for sharing and easily downloading machine learning modelshttps://huggingface.co/

`RevAI`: This model was built upon the `pyannote.audio` version 3.0 framework and it was fine-tuned on private transcriptions. [130].

The three pre-trained models have to be tested to find the most suitable for identifying the number of speakers in each audio file of the database. Before feeding the audio files to the pretrained models, it is required to resample them from 44 kHz to 8 kHz. Resampling is performed using Python's library Librosa [85].

For testing the models, a sample of 295 audio files is manually labeled with the number of speakers to serve as ground truth. The distribution of labels (number of speakers) in the sample is shown in Table 3.1. Then, each audio file from the sample is fed into the pre-trained models to predict the number of speakers. These predictions are then compared against the ground truth.

Table 3.1: Distribution of the 295 total audio files by number of speakers

Number of speakers	Total of audiofiles
0	10
1	25
2	213
3	40
4	5
5	2

To evaluate the performance of the models, a confusion matrix (CM) is generated for each of them, comparing the predicted labels vs the ground truth. These CMs can be observed in Figure 3.3.

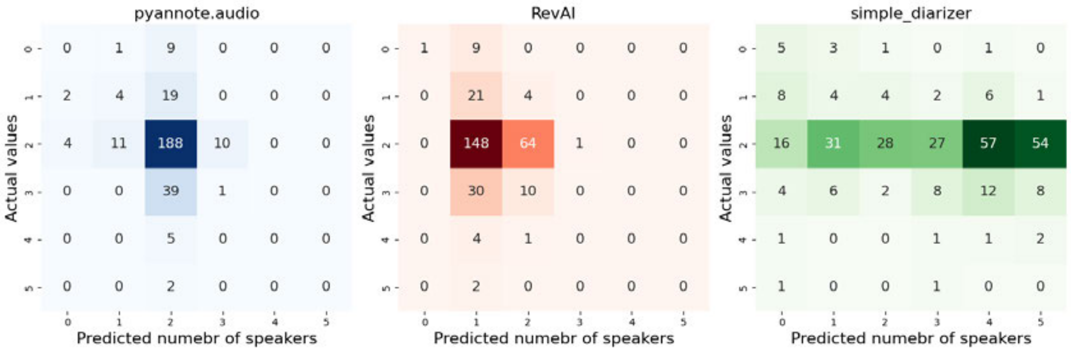


Figure 3.3: Confusion matrices for evaluating the predictions of the speaker diarization models on a sample of 295 audio files. From left to right: `pyannote.audio`, `RevAI`, and `simple_diarizer` confusion matrices of their predictions vs actual labels.

By analysing the CMs in Figure 3.3, it is evident that from the three pre-trained models, `pyannote.audio`, from now on `pyannote`, classified more audio files correctly.

Therefore, this tool is chosen for identifying the number of speakers for the whole database. The rest of the audio files that are not part of the sample are fed into the pre-trained model and are given a label for the number of speakers.

Once the number of speakers is found for each audio file, only those files containing the speech of two speakers are retained. Thus, the GER is delimited to groups of two people. Although larger groups of speakers can be studied, it is decided to stick to a uniform number of speakers in the mixture. Two speakers are considered reasonable due to the duration of the audio files (5 seconds). Capturing meaningful interactions of more than two speakers would require longer audio signals.

Initially, a total of 1208 files are classified as two speakers, which are used to build the new database with speech mixtures of two speakers. Before discarding the remaining files that are not classified as two speakers by `pyannote`, another approach is taken, aiming to retain as many audio files as possible from the original database. This is done considering that the pre-trained model can misclassify some of the samples. Thus, the goal is to find false negatives (audio files that have two speakers but were initially classified with a different number of speakers) to increase the number of videos in the new database with two speakers.

The approach consists of processing the rest of the audio files with two pre-trained models for speech enhancement, to further process the "enhanced" audio files by `pyannote` again to identify the number of speakers. It is hypothesized that the "enhanced" audio files improve the speaker diarization process. After implementing the approach, a total of 128 new files were identified as containing two speakers. Each of these files is manually reviewed (by carefully listening to the non-enhanced audio files) to verify if the new label for the number of speakers is correct. From the 128 audio files, only 61 were correctly classified as two speakers.

Finally, the new database consists of 1482 audio files with speech mixtures of two speakers. The distribution of labels in the new database is 575 Negative, 420 Neutral, and 487 Positive, and is visualized in Figure 3.4.

The speech-enhancement models could indeed enhance the speech for some of the audio files. Nonetheless, during the manual revision of the files, it was noticed that important segments of speech were lost in some audio files after the enhancement. Losing speech is not desirable. Hence, the audio files in the new database are processed without enhancing or denoising methods applied.

The next step is to reorganize the data into training, validation, and test subsets, ensuring that the distribution of labels is maintained across all subsets. Two separate test subsets are built: a speaker-independent subset and a speaker-dependent subset. The term *independent* implies that the speakers in the test subset are not the same as in the audio files of the training and validation subsets. Having these two test subsets allows evaluating the model's performance on known and unknown speakers.

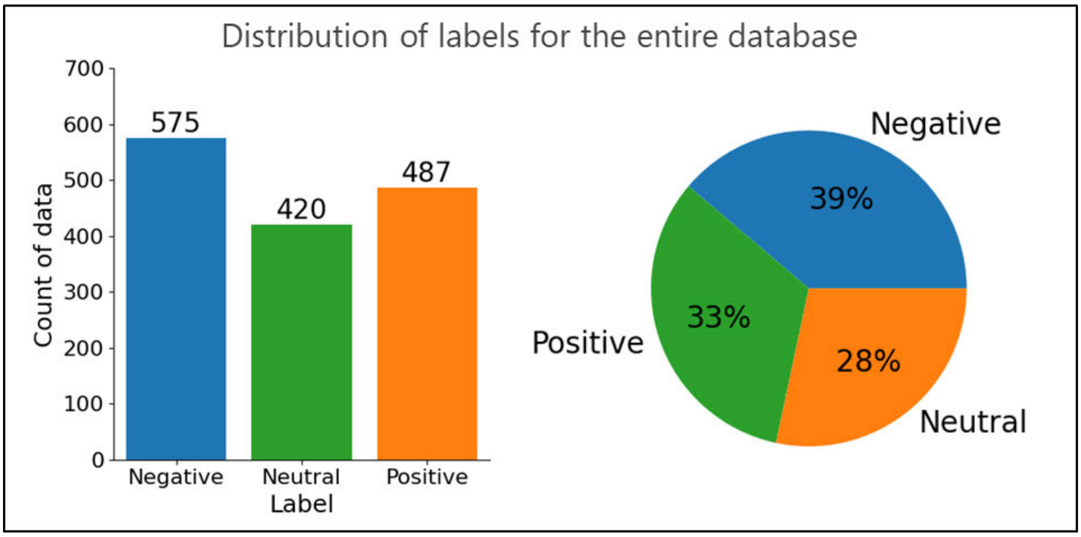


Figure 3.4: Distribution of labels in the new database: 39% Negative, 28% Neutral and 33% Positive

In the process of building the subsets, the audio files are grouped by their video identifier (video ID), which is contained in the file name. This makes it easy to identify how many files are segments cropped from the same video. A plot is generated to visualize the number of segments and labels for each video ID. In Figure 3.5, the top 30 videos with the highest number of segments are plotted.

It is assumed that all segments cropped from the same video have the same speakers. Thus, to build the speaker-independent test subset, nine video IDs are manually selected and removed from the rest of the database. The nine video IDs ("328", "64", "168", "81", "178", "43", "209", "296" and "319") are strategically chosen to build a subset with the same distribution of labels as the whole database: 39% Negative, 28% Neutral and 33% Positive. As observed in Figure 3.6, the speaker-independent test subset has a total of 100 segments (audio files).

The next step is to build the training, validation, and speaker-dependent test subsets using the remaining data. For each video ID, the approach involves assigning 70% of the segments to the training subset, 15% to the validation subset, and the remaining 15% to the speaker-dependent test subset. For example, if 20 segments share the same video ID, 14 of them are assigned to the training subset, 3 segments to the validation subset, and the remaining 3 segments to the speaker-dependent test subset. In cases where only three segments share the same video ID, one segment is assigned to each subset. When only two or one segments share the same video ID, these segments are directly assigned to the training subset.

Thus, after following the approach, the new data subsets are built with the following distributions of data: 921 audio files in the training subset, 247 in the validation

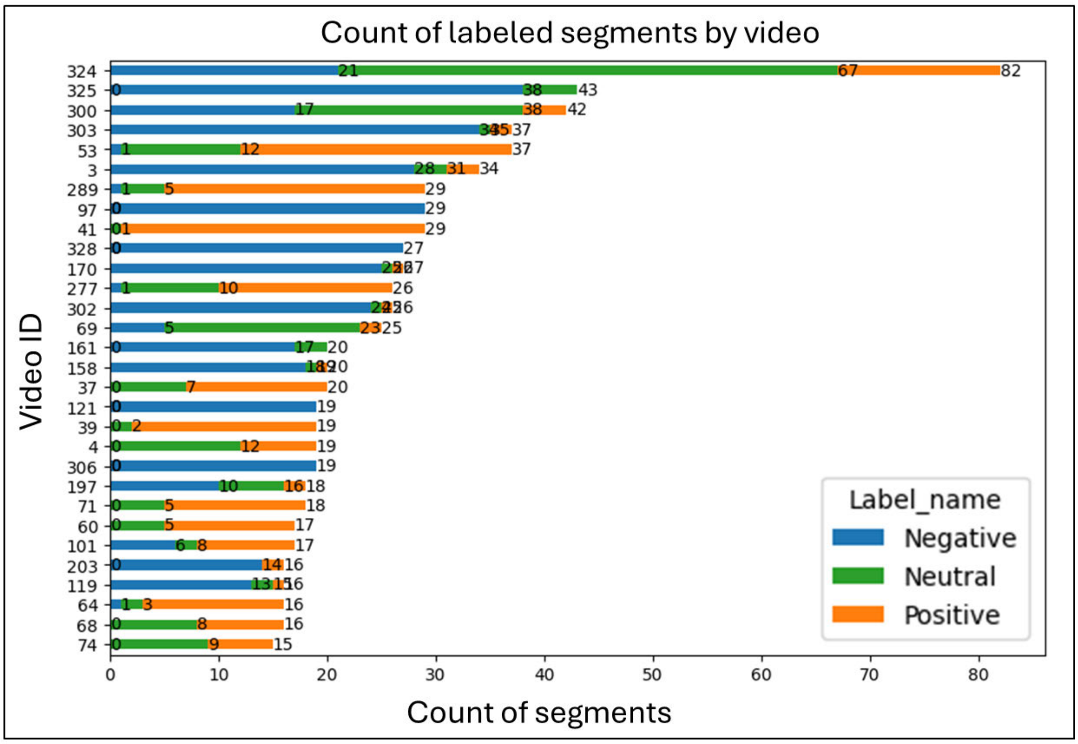
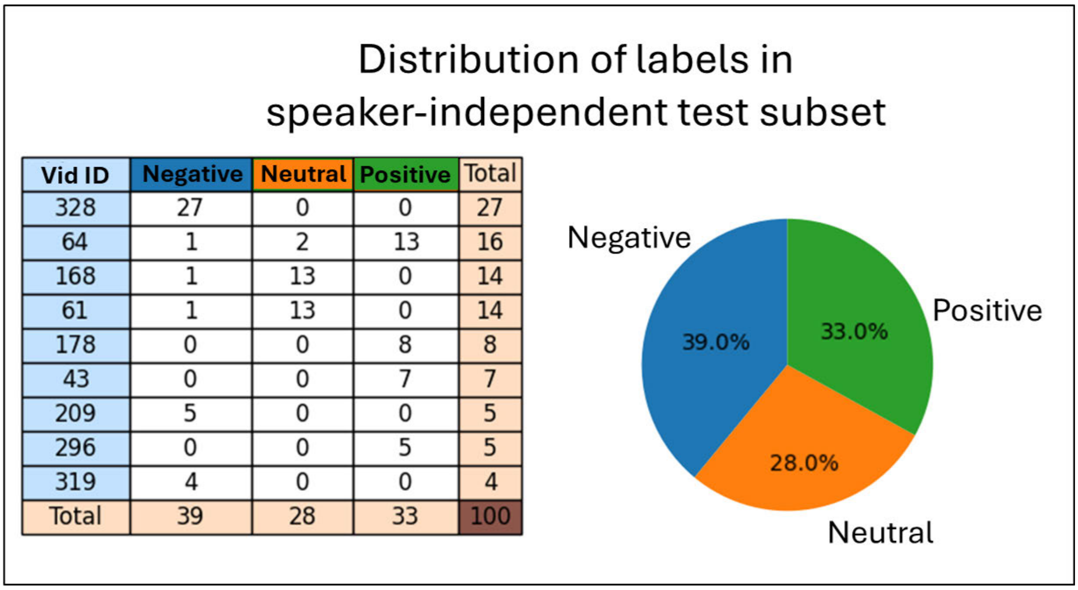


Figure 3.5: Distribution of labels and count of segments for each video ID (showing only the plot for the 30 videos with the largest number of segments). The video ID “324” has the highest number of segments (82), 21 of them are labeled as Negative, 46 as Neutral, and 15 as Positive.

subset, 214 in the speaker-dependent test subset, and 100 in the speaker-independent test subset. The total of data for each subset can be visualized in Figure 3.7.

The relative distribution of group emotion labels for each subset is close to the distribution for the whole database (39% Negative, 28% Neutral, and 33% Positive), with minor differences. The distribution of labels across classes for each subset is visualized in Figure 3.8.



Vid ID	Negative	Neutral	Positive	Total
328	27	0	0	27
64	1	2	13	16
168	1	13	0	14
61	1	13	0	14
178	0	0	8	8
43	0	0	7	7
209	5	0	0	5
296	0	0	5	5
319	4	0	0	4
Total	39	28	33	100

Figure 3.6: Distribution of labels for the speaker-independent test subset. A total of 100 audio segments are selected, corresponding to nine different video IDs for building the subset. A total of 39 labels are Negative, 28 are Neutral, and 33 are Positive.

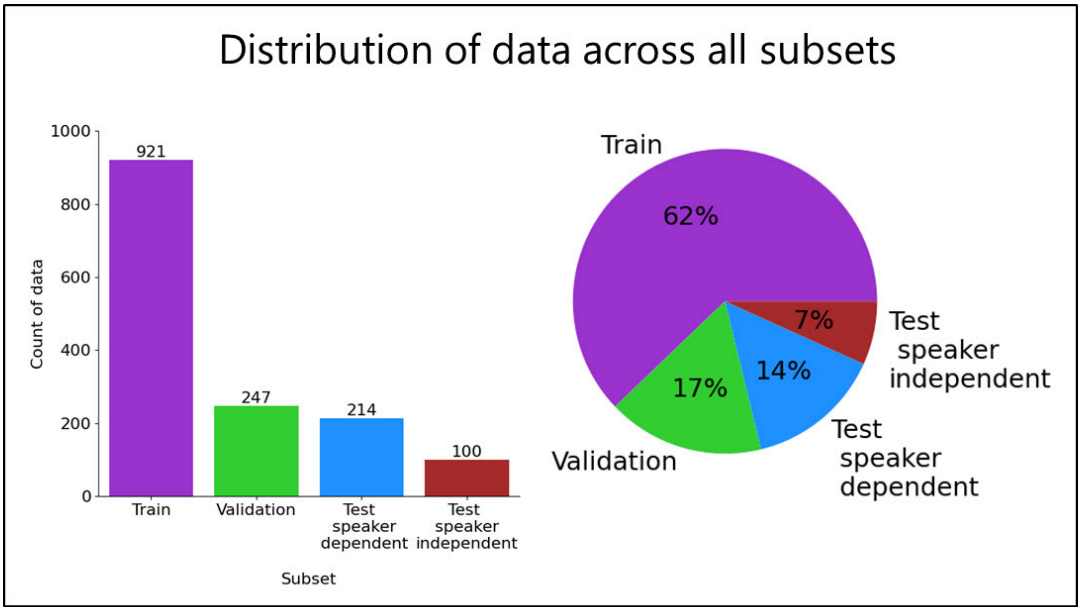


Figure 3.7: Total of data in each subset as number and as relative percentage of the whole database.

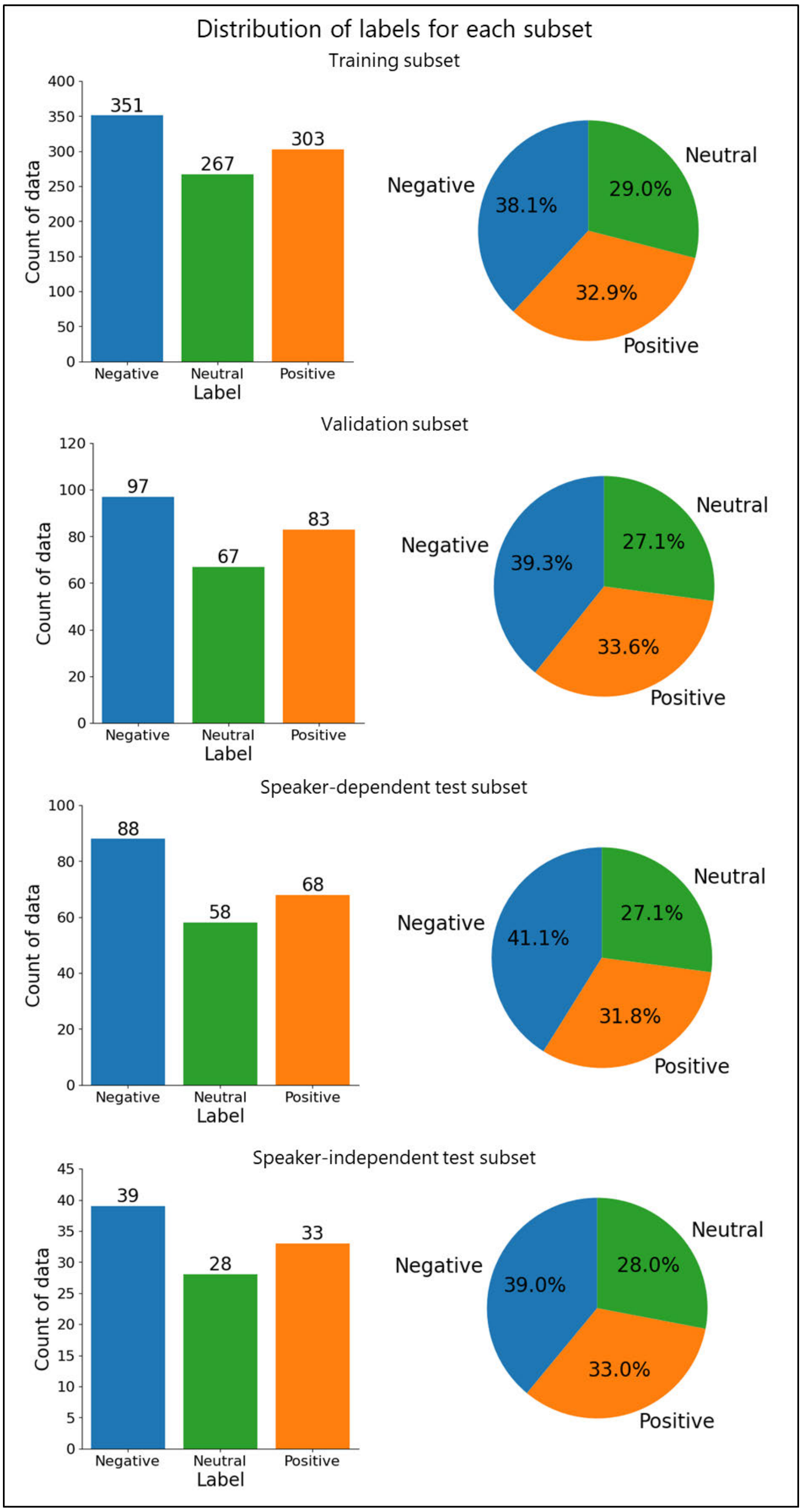


Figure 3.8: Distribution of group emotion labels for each subset.

3.1.0.1. Speech separation

Once the audio files are assigned to the different subsets, the next step is to separate the speech sources from the mixture. Each audio file is expected to have a mixture of two speech sources; thus, two separated speech signals containing the speech of the individual speakers are expected as output from the speech separation process.

For separating the speech mixtures, an end-to-end speech separation system is used. The MossFormer2 [131] is selected for the speech separation task. Not only due to being easy to implement, but also due to its state-of-the-art performance for mono-channel speech separation. The Mossformer2 is a pre-trained model for speech separation based on a hybrid transformer and a recurrent module. It is ranked among the top five speech separation models on benchmark datasets such as the WSJO-2mix [132], Libri2Mix [133], WHAM! [134] and WHAMR! [135].

The audio files sampled at 8 kHz are used as input for the model. As output, the model generates two WAV files, containing the separated speech of the individual speakers. These two files are named similarly to the audio file with the mixture signal, but with the suffix "_s1" and "_s2" to identify them as the separated speech of the "speaker 1" and "speaker 2", respectively. For instance, for the file "34_5.wav" containing the non-separated speech, two files are generated: "34_5_s1.wav", which contains the separated speech of the first speaker, and "34_5_s2.wav", which contains the separated speech of the second speaker. In summary, for each label in the database, there are now three WAV files: one containing the speech mixture and two containing the separated speeches.

It is noticed that the separated signals are rescaled by the speech separation model, while rescaling does not affect the frequency content of the signal, it can affect features that depend on the amplitude of the spectral components and the energy of the signal. This is taken into consideration for building the feature sets.

3.2. Extraction of features

OpenSMILE is an open-source, modular, and flexible feature extractor designed for signal processing and machine learning applications, targeting researchers and system developers [83]. This tool allows users to select different feature sets to be extracted as both LLDs and/or functionals. The tool has configurations for extracting frequently used acoustic feature sets for SER, including all the baseline feature sets of the INTERSPEECH ComParE challenges.

As mentioned in Section 2.3, the ComParE 2013 feature set is one of the most comprehensive sets available, comprising 6373 features. It has been used as the standard acoustic feature set in the ComParE challenge series and served as the baseline for the EmotiW 2020 challenge, where the Video Group Affect database (VGAF) database, used in this study, was introduced. Given its extensive coverage and proven effectiveness, this feature set was selected for the present study, with OpenSMILE chosen as the feature-extraction tool.

To ensure clarity in the following descriptions, the terms "audio file" or "audio segment" will be exclusively used to refer to the original audio files containing the mixture signals. The term "WAV files" might be used to refer to both the files containing the mixture signals and the files containing the separated speech.

All WAV files —mixture signals and separated speech— are processed by the tool, these files have a sampling frequency of 8 kHz. A total of 6373 parameters are calculated for each WAV file. The features extracted from the WAV files containing the mixture signals are referred to as *mixture features*, and the features extracted from the WAV files containing the separated speech are referred to as *individual features*. Thus, a total of 6373 mixture features and 12746 individual features are computed for each audio segment in the database. These two types of features are combined to produce different feature sets that are used for training models on recognizing group emotion. The development of these feature sets and the implementation of the models are described further in Section 3.3.

3.3. Preparation of feature sets

As reviewed earlier, the group emotion depends on factors at the group and individual levels. Therefore, the feature sets used for training the models need to contain mixture features and individual features. In this work, different combinations of mixture and individual features as inputs for training models to perform GER are explored. This has the main objective to identify and analyze the contribution of features from individual speakers to the recognition of the group emotion.

To achieve that, a feature-level fusion method is employed: *concatenation*. According to the literature review by Huang et al. [18], this method is commonly employed for fusion of features from different data modalities, and features extracted at different levels (group and individual level).

The individual features are first grouped into categories according to their acoustic properties (cepstral, spectral, prosodic, sound quality, or temporal). Meanwhile, the mixture features remain ungrouped. As a baseline, two models are trained on mixture features exclusively. Then, in each experiment, the mixture features are concatenated with a different category of individual features and used for training the models. This approach allows for a clear observation of the effect of different groups of features from individual speakers on the recognition of group emotion. Following this, the feature sets are outlined below in alignment with the objectives of each experiment.

- **Baseline feature set:**

  The baseline aims to reflect the ability of the models to recognize group emotion from mixture features only. Thus, the baseline feature set consists exclusively of mixture-based parameters. All mixture features are used, resulting in 6373 features for each sample.

- **Experiment 1: All individual features**

  This experiment aims to reflect the ability of the models to recognize group emotion from mixture features supported by *all* features extracted from the individual speakers. This feature set consists of all of the mixture features (6373) concatenated with all of the individual features (12746 features, 6373 from each speaker), totaling 19119 features per sample.

- **Experiment 2: Cepstral-individual features**

  This experiment aims to reflect the ability of the models to recognize group emotion from mixture features supported by *cepstral* features extracted from the individual speakers. This feature set consists of all of the mixture features (6373) concatenated with cepstral-individual features (2800 features, 1400 from each speaker), totaling 9173 features per sample.

- **Experiment 3: Prosodic-individual features**

  This experiment aims to reflect the ability of the models to recognize group emotion from mixture features supported by *prosodic* features extracted from the individual speakers. This feature set consists of all of the mixture features (6373) concatenated with prosodic-individual features (956 features, 478 from each speaker), totaling 7329 features per sample.

- **Experiment 4: Sound quality-individual features**

  This experiment aims to reflect the ability of the models to recognize group emotion from mixture features supported by *sound quality* features extracted from the individual speakers. This feature set consists of all of the mixture features (6373) concatenated with sound quality-individual features (780 features, 390 from each speaker), totaling 7153 features per sample.

- **Experiment 5: Spectral-individual features**

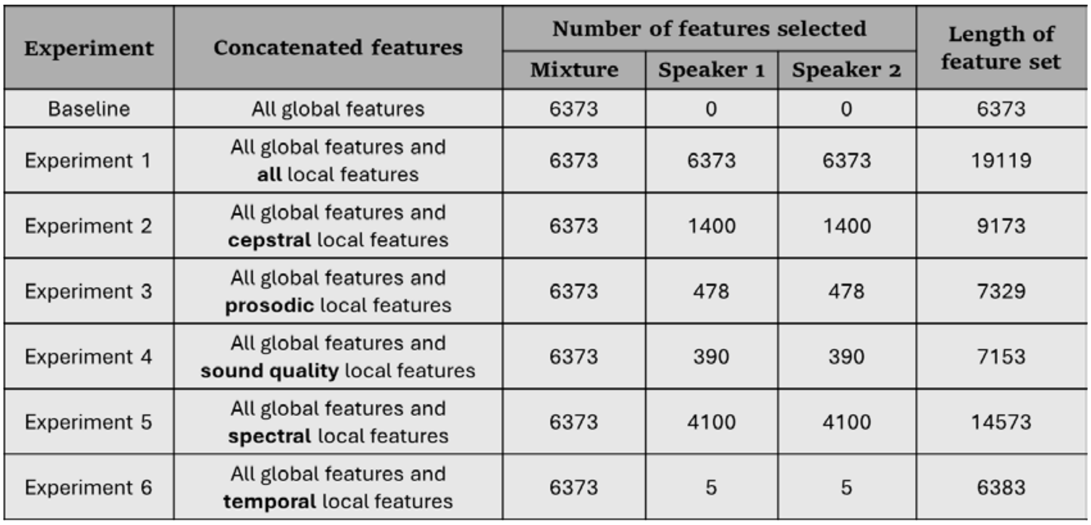
  This experiment aims to reflect the ability of the models to recognize group emotion from mixture features supported by *spectral* features extracted from the individual speakers. This feature set consists of all of the mixture features (6373) concatenated with spectral-individual features (8200 features, 4100 from each speaker), totaling 14573 features per sample.

- **Experiment 6: Temporal-individual features**

  This experiment aims to reflect the ability of the models to recognize group emotion from mixture features supported by *temporal* features extracted from the individual speakers. This feature set consists of all of the mixture features (6373) concatenated with temporal-individual features (10 features, 5 from each speaker), totaling 6383 features per sample.

The four data subsets (training, validation, speaker-dependent, and speaker-independent test subsets) are adapted according to each experiment, ensuring that the samples have the corresponding feature sets. Each feature in the training subset is normalized with a z-score before being fed to the models. The mean and standard deviation of the features from the training subset are also used to normalize the features in the validation and both test subsets. This normalization step is conducted at this stage, as it aligns with methodologies employed in multiple studies [5, 14, 59, 136]. Table 3.2 summarizes the feature sets used for the baseline and each experiment.

Table 3.2: Feature sets for each experiment: The baseline feature set consists solely of mixture features (features derived from the mixture signal). In each experiment, the feature sets comprise a specific group of individual features (features of the individual speakers) concatenated with all mixture features, resulting in feature sets of different lengths.



Experiment	Concatenated features	Number of features selected			Length of feature set
		Mixture	Speaker 1	Speaker 2	
Baseline	All global features	6373	0	0	6373
Experiment 1	All global features and **all** local features	6373	6373	6373	19119
Experiment 2	All global features and **cepstral** local features	6373	1400	1400	9173
Experiment 3	All global features and **prosodic** local features	6373	478	478	7329
Experiment 4	All global features and **sound quality** local features	6373	390	390	7153
Experiment 5	All global features and **spectral** local features	6373	4100	4100	14573
Experiment 6	All global features and **temporal** local features	6373	5	5	6383

Considering that the separated speech signals are rescaled (having a different amplitude than the mixture signal) and that amplitude-dependent features are affected by this, concatenating the features ensures that any potential variations present in the individual features are uniformly propagated across all feature sets. This approach, together with normalizing the features before being fed to the models, minimizes the impact of the rescaling and maintains consistency in the model's input data.

3.4. Group emotion recognition

In this study, the two models selected for GER are SVMs and FCNNs. In each experiment, the "same models" are trained using different normalized feature sets as input. Although the term "same models" is used, it refers to the same type of model rather than identical models, since each is trained on different training datasets. Therefore, while the models share the same conceptual framework, their training data will differ, resulting in variations in their learned parameters.

As reviewed earlier, SVMs are the most widely used ML model for SER [77, 107, 48]. SVMs are effective for high-dimensional data and provide robust performance with limited datasets. They are particularly suitable for multi-class classification tasks, making them ideal for this application.

Additionally, FCNNs are chosen due to their ability to capture complex patterns and relationships within the data. They are highly scalable and versatile, allowing for effective emotion recognition from audio signals. The use of DNNs is well-supported by state-of-the-art methods in SER and GER; furthermore, their performance has already been reported for GER on the VGAF database [50, 117, 48].

In the next sections, Section 3.4.1 and Section 3.4.2, the implementation of the models is described.

3.4.1. Support vector machines

The SVM models implemented use the `SVC` class from `sklearn`[3] library. The following default hyperparameters are used for the SVM models:

- **Kernel:** Radial Basis Function (RBF)
- **C:** 1.0 (Regularization parameter)
- **Gamma:** "scale" (Kernel coefficient)

The SVMs can easily be adapted for different lengths of feature sets using the same hyperparameters. Therefore, the same values defined above are used in the models of all experiments and the baseline, independently of the length or shape of the feature sets.

[3] sklearn, also referred to as scikit-learn, is an open source machine learning Python library that provides tools for model development and evaluation `https://scikit-learn.org/stable/`

A total of seven SVMs are trained, one for each experiment, including the baseline. Each SVM is fitted to the training data and is then used for predicting (recognizing) emotions on both the speaker-dependent and speaker-independent test subsets, producing a total of 14 sets of predicted labels.

3.4.2. Fully connected neural networks

The FCNNs implemented use the PyTorch framework (`torch` version 2.5.1), and are trained using CUDA[4] version 12.4 for GPU acceleration.

The architecture of the FCNNs is reimplemented from an FCNN used in [50] for GER. Only the number of units at the input layer N is adapted to fit the length of the different feature sets that are used as inputs in each experiment.

1. **Input layer:** N number of input units.
2. **First hidden layer:** 128 neurons with ReLU activation function.
3. **Second hidden layer:** 256 neurons with ReLU activation function.
4. **Third hidden layer:** 512 neurons with ReLU activation function.
5. **Fourth hidden layer:** 1024 neurons with ReLU activation function.
6. **Fifth hidden layer:** 2048 neurons with ReLU activation function.
7. **Output layer:** 3 neurons corresponding to each of the labels of the dataset (Negative, Positive, Neutral) with a ReLu activation function.

Similar to the reference FCNN model, Cross-Entropy loss is selected as the loss function. Unlike SVMs, which can easily be adapted for different input lengths, the architecture and hyperparameters of NNs need to be customized to the training data, such as the number of units in the input layer and the learning rate. The number of inputs must be established in the architecture before training the model. Therefore, for each experiment, the value N is the length of the feature set used in that experiment.

Using the same learning rate for training different models is not recommended, since different feature sets may require different learning rates to achieve optimal performance. Therefore, for training the models, a *learning rate scheduler*, `ReduceLROnPlateau`, is used for adjusting the learning rate when the validation loss is not reduced after a certain number of epochs. Typically, a patience parameter is given, which will tell the algorithm to "wait" for a certain number of epochs of no improvement before reducing the learning rate.

[4] CUDA (Compute Unified Device Architecture) is a parallel computing platform and application programming interface (API) model developed by Nvidia that allows software to use certain types of graphics processing units (GPUs) `https://developer.nvidia.com/`

Additionally, an *early stopping* class is used to interrupt the training loop when there is no significant improvement in the validation loss between epochs. This prevents the model from overfitting[5]. Additionally, a separate function keeps track of the parameters of the best model between epochs (i.e., the parameters of the model with the lowest validation loss), and when the training is interrupted, the model is updated with these parameters.

To ensure the reproducibility of the models, the parameters are initialized before the training. The biases are initialized to zero. The weights are initialized using a Xavier uniform initialization method, with a torch generator. The torch generator is set to a fixed seed value to maintain consistency across runs. The seed = 0 was used for the models in the Baseline and Experiment 1; the rest of the experiments have a seed = 3. For the training, mini-batch gradient descent is chosen as the optimizer. Thus, the model's parameters are updated more frequently, improving both computational efficiency and the convergence rate of the training process. The batch size is set to 32, with shuffle activated for the training data. Again, a random number generator with a manual seed with value 0 is used to ensure reproducibility of the shuffling process. Using a batch size of 32 means that the model processes 32 samples to calculate the gradient and adjust the weights before moving on to the next set of 32 samples. Although this is considered a small batch size, it has the advantages of updating the weights more frequently between epochs and producing noisy gradients, which can help in escaping local minima.

To summarize, the following list presents functions, methods, and hyperparameters that are used to train the FCNN models across all experiments and the baseline. All of these hyperparameters remained consistent for training the FCNN in all experiments.

1. **Batch size:** 32.
2. **Loss function:** Cross-Entropy Loss.
3. **Optimizer:** Mini-batch gradient descent.
4. **Learning rate:** Initial learning rate set to 0.01.
5. **Learning rate scheduler:** `ReduceLROnPlateau` with a reducing factor of 0.1, patience 2, using "validation loss" as metric, and decreasing mode ("min").
6. **Weights initializer:** Xavier uniform.
7. **Early stopping:** Custom early stopping function with patience set to 30 and delta set to 0.001.
8. **Epochs:** If the early stopping function does not interrupt the training, a maximum of 1000 epochs is set up.

[5] Overfitting occurs when a model learns the training data too well, including its noise and outliers, resulting in poor generalization to new, unseen data.

During the training, it was noticed that the early stopping function interrupted the training in all experiments. In general, the interruptions occurred between epochs 30 and 40. Since patience is set to 30, this means that 10 or fewer epochs were needed for training the models. The learning rate scheduler adjusted the learning rates between epochs. However, in every experiment, the model that achieved the best performance (lowest validation loss) was reported to use a learning rate of 0.01.

A total of seven FCNNs are trained, one for each experiment, including the baseline. Each model is trained on the training data and uses the validation data for optimizing the parameters during training. The trained models are used for predicting (recognizing) emotions on both the speaker-dependent and speaker-independent test subsets. This produced a total of 14 sets of predicted labels. In summary, seven SVMs and seven FCNNs are trained, each of them tested on both test subsets, producing a total of 28 predicted labels. Using the predictions of the models and the true labels, the performance of the models can be assessed.

3.5. Evaluation of the models

Given the imbalance in the database, it is crucial to use metrics that can effectively evaluate the performance of the model under these conditions. Therefore, the chosen metrics to be used for assessing the model's performance are CM, UAR, and macro F1-score. These metrics were introduced in Section 2.1 as common evaluation metrics for multi-class classification models. These are particularly suited for imbalanced datasets as they provide a clearer understanding of the model's ability to correctly classify minority class instances. Additionally, the F1-scores of the Positive, Negative, and Neutral classes are calculated for each model, to provide a better view of the performance of the models classwise. For each of the 28 predicted sets of labels, the three metrics are calculated and visualized in the results (Section 4).

3.6. Significance test

For each model, the UAR and F1-scores reported in each experiment are compared against the baseline with a significant test, specifically a *one-tailed z-test.* This test is employed to determine if one process (p_2) is better than another process (p_1) at a given significance level.

The test is made on the assumption that two experiments, consisting of N independent trials, were conducted. In the first experiment, $p_1 \times N$ trials resulted in success. In the second experiment, $p_2 \times N$ trials resulted in success. The objective is to determine if p_2 is significantly better than p_1 in a statistical sense, based on N samples.

For the hypothesis test, these two hypotheses need to be formulated.

- Null Hypothesis (H_0): The scores from both processes are identical.
- Alternative Hypothesis (H_1): The score from process p_2 is better than the score from process p_1.

In this work, p_1 represents the score between 0 and 1 obtained by the baseline, while p_2 represents the score between 0 and 1 obtained by the experiment to be compared against the baseline. The value of N is determined by the sample size of the test subset to be evaluated.

To calculate if p_2 is indeed better than p_1 at a certain significance level, the following procedure needs to be followed:

1. Calculate the difference between the scores:

$$\text{diff} = |p_1 - p_2| \tag{3.1}$$

2. Calculate the standard deviation of the difference:

$$\sigma_{\text{diff}} = \sqrt{\frac{p_1 \cdot (1 - p_1) + p_2 \cdot (1 - p_2)}{N}} \tag{3.2}$$

3. Compute the z-score:

$$z = \frac{\text{diff}}{\sigma_{\text{diff}}} \tag{3.3}$$

4. Compare the Z-score to critical values for different significance levels:

   Table 3.3 shows common significance levels and their corresponding critical values, which are derived from the standard normal distribution.

Table 3.3: Significance Levels and Critical Values

Significance Level (α)	**Critical Value (z_α)**
0.05	1.644853627
0.01	2.326347874
0.005	2.575829304
0.002	2.878161739
0.001	3.090232306

   If $z > z_\alpha$, the difference is significant at that level.

For each experiment, the UAR and macro F1-scores obtained for each model are subjected to a significance test against their respective baselines. This has the purpose of determining whether the scores achieved in the experiments are significantly superior to the baseline scores at different significance levels.

4 Results and discussion

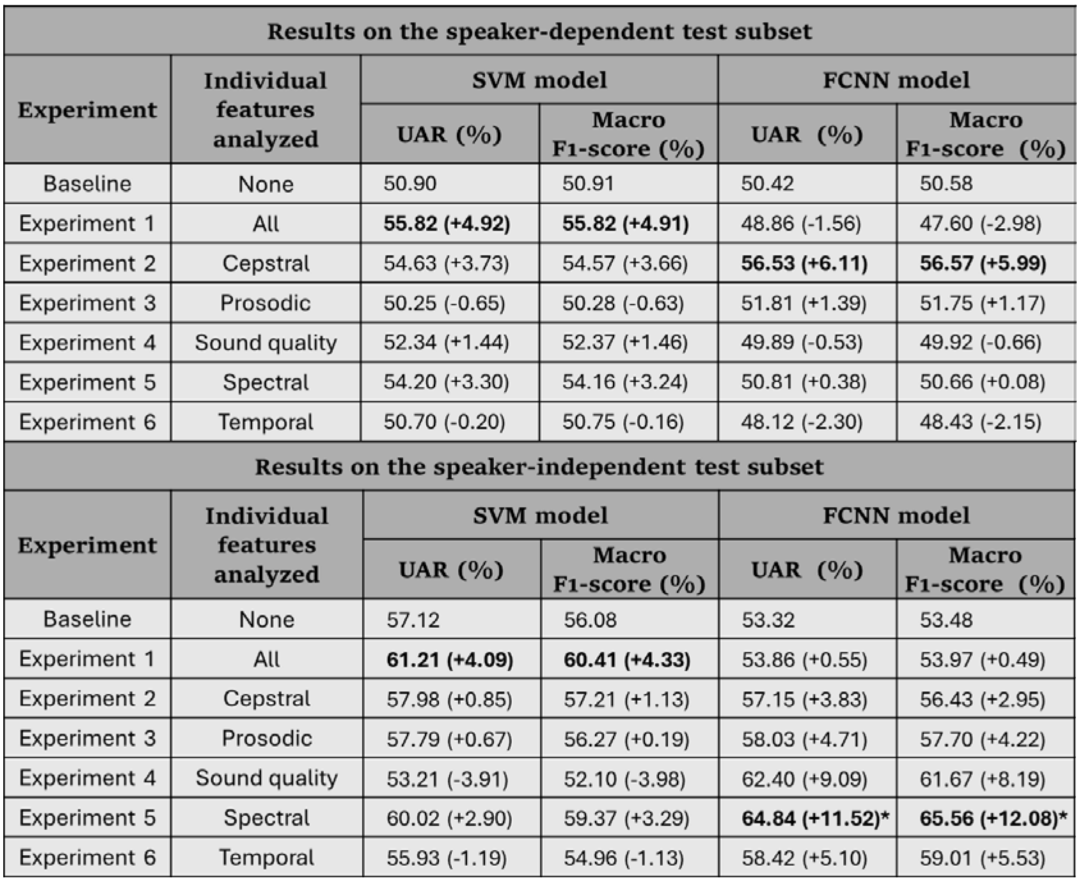
Multiple experiments were conducted using SVMs and FCNNs for GER from speech. For the SVMs, one baseline was calculated on the speaker-independent test subset, and a second one on the speaker-dependent test subset. For brevity, from now on the speaker-dependent and speaker-independet test subsets, are referred to as *spk-dep* and *spk-ind*, respectively. Similarly, for the FCNN, two baselines were calculated, one for each test subset. Thus, a total of four baselines were obtained. The models used for the baselines were trained exclusively on mixture features (features extracted from speech mixtures). Whereas, the models in the experiments were trained on combinations of both mixture features and individual features (features derived from the individual speakers).

The performance of each model is compared against its corresponding baseline. As evaluation metrics, the macro-averaged F1-scores, UAR scores, and CMs are used. Additionally, the F1-scores for each class are used for the analysis. The macro F1-score is the average of the F1-scores for each class, defined as the harmonic mean of precision and recall. Precision measures how many samples classified as a class i are indeed samples of class i, and recall is a measure of how many of the actual samples of class i are correctly classified. The values for macro F1-score range between 0 and 100%, with an F1-score = 100% indicating perfect precision and recall across all classes. UAR is the average recall for each class; the value ranges between 0 and 100%, with a UAR = 100% indicating perfect recall across all classes. The CM is a table that shows the actual versus predicted classifications, where a good CM has high values along the diagonal (true positives) and low values off-diagonal (false positives and false negatives).

Table 4.1 shows the UAR and macro F1-scores obtained on both the spk-dep and spk-ind test subsets, respectively. Figure 4.1 presents the macro F1-scores obtained in each experiment for both models on the two test subsets.

As observed in Table 4.1 and Figure 4.1, counterintuitively, in most of the experiments, including the baseline, the scores obtained on the spk-ind test subset (unknown speakers) were higher than those obtained on the spk-dep test subset (known speakers). This could be attributed to several factors. Starting from the distribution of labels on both subsets, the proportion of Negative samples in the spk-dep subset (41.1%) is 2.1% higher compared to the proportion of Negative samples existing in the spk-ind subset (39%), as presented earlier in Section 3 on Figure 3.8. Although the difference in proportion is not that big, this could still produce an impact, inflating the UAR and

Table 4.1: UAR and macro F1-scores obtained for each model on both the speaker-dependent and speaker-independent test subsets. UAR values range from 0 to 100%. UAR = 100% indicates perfect recall on the three classes. Macro F1-scores range from 0 to 100%, with 100% indicating perfect precision and recall across all three classes. Higher scores indicate a good performance of the model in recognizing the three classes, lower scores indicate poorer performance of the model. The values in parentheses indicate the difference from the baseline results. The results highlighted in bold correspond to the SVM and FCNN models, which achieved the highest scores in the speaker-dependent test subset. An asterisk (*) denotes that the score is statistically significant $p < 0.05$.



Results on the speaker-dependent test subset					
Experiment	**Individual features analyzed**	**SVM model**		**FCNN model**	
		UAR (%)	**Macro F1-score (%)**	**UAR (%)**	**Macro F1-score (%)**
Baseline	None	50.90	50.91	50.42	50.58
Experiment 1	All	**55.82 (+4.92)**	**55.82 (+4.91)**	48.86 (-1.56)	47.60 (-2.98)
Experiment 2	Cepstral	54.63 (+3.73)	54.57 (+3.66)	**56.53 (+6.11)**	**56.57 (+5.99)**
Experiment 3	Prosodic	50.25 (-0.65)	50.28 (-0.63)	51.81 (+1.39)	51.75 (+1.17)
Experiment 4	Sound quality	52.34 (+1.44)	52.37 (+1.46)	49.89 (-0.53)	49.92 (-0.66)
Experiment 5	Spectral	54.20 (+3.30)	54.16 (+3.24)	50.81 (+0.38)	50.66 (+0.08)
Experiment 6	Temporal	50.70 (-0.20)	50.75 (-0.16)	48.12 (-2.30)	48.43 (-2.15)

Results on the speaker-independent test subset					
Experiment	**Individual features analyzed**	**SVM model**		**FCNN model**	
		UAR (%)	**Macro F1-score (%)**	**UAR (%)**	**Macro F1-score (%)**
Baseline	None	57.12	56.08	53.32	53.48
Experiment 1	All	**61.21 (+4.09)**	**60.41 (+4.33)**	53.86 (+0.55)	53.97 (+0.49)
Experiment 2	Cepstral	57.98 (+0.85)	57.21 (+1.13)	57.15 (+3.83)	56.43 (+2.95)
Experiment 3	Prosodic	57.79 (+0.67)	56.27 (+0.19)	58.03 (+4.71)	57.70 (+4.22)
Experiment 4	Sound quality	53.21 (-3.91)	52.10 (-3.98)	62.40 (+9.09)	61.67 (+8.19)
Experiment 5	Spectral	60.02 (+2.90)	59.37 (+3.29)	**64.84 (+11.52)***	**65.56 (+12.08)***
Experiment 6	Temporal	55.93 (-1.19)	54.96 (-1.13)	58.42 (+5.10)	59.01 (+5.53)

macro F1-scores when the classes are not equally distributed across both subsets. Additionally, the number of samples in the spk-ind subset is half the number of samples in the spk-dep subset. The spk-ind subset, which has fewer samples, is more prone to variance; therefore, a significance test is needed to validate if the differences achieved in scores are statistically significant at different significance levels.

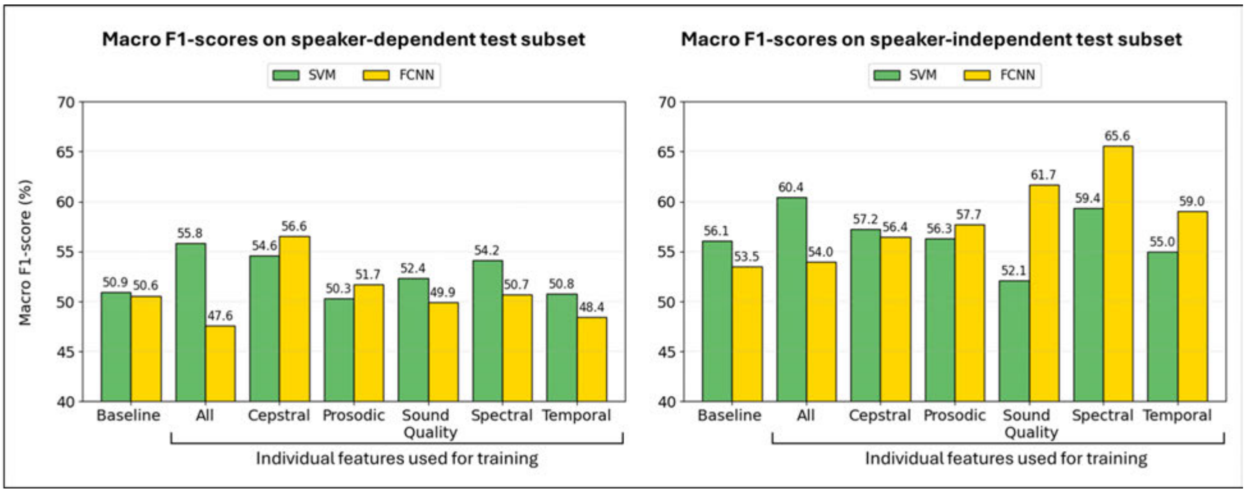


Figure 4.1: Macro F1-scores obtained for both models on both test subsets. Macro F1-scores range from 0 to 100%, with 100% indicating perfect precision and recall across all three classes. Higher scores indicate a good performance of the model in recognizing the three classes, lower scores indicate poorer performance of the model.

The following paragraphs provide a detailed discussion of the results, with each experiment addressed individually.

In Experiment 1 (model trained on all mixture features and all groups of individual features), the SVM model reached its peak performance on both test subsets, surpassing the macro F1-score of their respective baselines by 4.91% for the spk-dep subset, and 4.33% for the spk-ind subset, as observed in Table 4.1 and Figure 4.1. Contrastingly, for the same experiment but with the FCNN model, the outcomes were unsatisfactory. The model's performance declined on the spk-dep, as indicated by the macro-F1-score = 47.60%, which is 2.98% lower than the baseline, and the scores obtained on the spk-ind showed little to no improvement (both < 0.5%).

The models in Experiment 2 were trained on a fusion of mixture features and cepstral-individual features. Cepstral features consist of functionals applied to the MFCCs of the audio signals. The MFCCs alone have been used in SER and proven efficient [137, 138, 118]. Therefore, this experiment was expected to achieve some of the highest scores. However, this was only the case for the FCNN model, where it achieved the highest scores (UAR = 56.53% and macro F1-score = 56.57%) on the spk-dep test subset, as indicated in Table 4.1. On the same table, it is observed that the performance of the FCNN on the spk-ind test subset did not benefit much from the cepstral-individual features, improving the baseline macro F1-score by only 2.95%. Concerning the SVM model for the same Experiment 2, the results obtained surpassed the macro F1-scores of the baseline on both test subsets, by 3.66% on the spk-dep, and with less impact on the spk-ind subset by only 1.13%.

In Experiment 3, prosodic features from the individual speakers were used in combination with mixture features for training the model. As introduced in Section 2.7 (State of the art), two studies on SER at the individual level concluded that pitch features demonstrated the highest contribution to SER. Given that pitch is grouped as a prosodic feature, it led to the expectation of getting higher scores for this experiment. However, only the FCNN model on the spk-ind test subset obtained substantially higher scores than the baseline, with a UAR score of 58.03% against the baseline UAR of 50.42%, as observed in Table 4.1. On the spk-dep subset, the improvement in UAR and macro F1-score was barely above 1% for the same model. On the other hand, the SVM did worse than the baseline on the spk-dep test subset, and achieved less than 1% improvement of the scores on the spk-ind test subset.

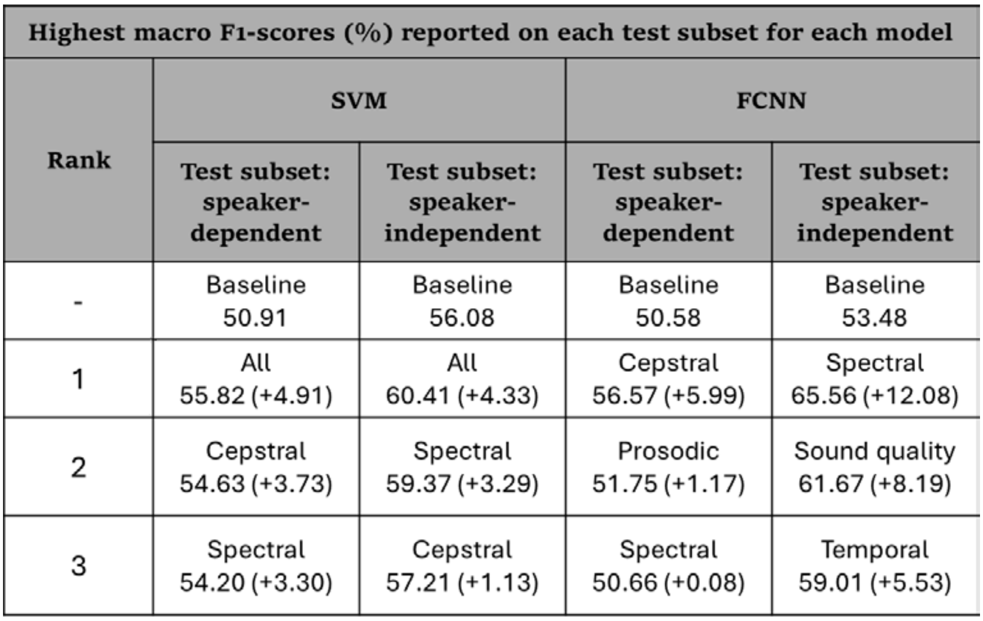
The results are contrasting for Experiment 4, where the models were trained using sound quality-individual features concatenated with mixture features. As observed in Table 4.1, the SVM model achieved a UAR only 1.5% higher than the baseline for the spk-dep subset, while on the spk-ind subset, the scores were worse than the baseline. Meanwhile, the FCNN model surpassed the UAR and macro F1-score of the baseline on the spk-ind test subset by 9 and 8%, respectively. Interestingly, the same model reported lower scores than the baseline on the subset with known speakers (spk-dep), which might suggest that sound-quality features, which contain information about the clarity and noisiness of the speech signal, are more beneficial for recognizing the group emotion on unknown speakers.

In Experiment 5, spectral-individual features are used in combination with mixture features for training the models. According to the results observed in Table 4.1, the SVM model demonstrated enhancements over the baseline scores on both test subsets; on the spk-dep subset, the UAR score was 3.30% above the baseline; and on the spk-ind subset, the UAR score was 2.90% above the baseline. The performance of the FCNN model differed for each test subset. On the spk-dep subset, the improvement was negligible, with a UAR score just 0.38% higher than the baseline. However, on the spk-ind subset, the model showed significant enhancement as visualized in Figure 4.1, achieving a UAR of 64.84% and a macro F1-score of 65.56%, which represented increases of 11.52% and 12.08% over the baseline, respectively. These scores were the highest observed in all conducted experiments and were identified as statistically significant at a 0.05 significance level, as indicated by the results of the hypothesis test (the scores are signalized with an asterisk in Table 4.1). The improvements in the scores obtained by the two models on both test subsets indicate the potential of spectral-individual features in improving GER.

Finally, in Experiment 6, mixture features were concatenated with temporal-individual features for model training. In this experiment, the performance of the SVM was worse than the baseline on both test subsets. On the other hand, for the FCNN model, improvements were achieved but only on the spk-ind subset, where the UAR score reported is 5.53% above the baseline. This suggests that temporal information from the individual speakers is only beneficial for GER on unknown speakers.

In summary, despite training both models on the same features in each experiment, their results differ substantially between the two test subsets. Table 4.2 shows the top three groups of individual features that yielded the highest scores for each model on each test subset and their baseline. For the SVM, on both test subsets, the highest scores were obtained when the SVM was trained with mixture features concatenated with the following individual features: all features together, spectral-individual features, and cepstral-individual features. For the FCNN, spectral, sound quality and temporal individual features improved the performance of the model on the spk-ind test subset; while cepstral, prosodic and spectral individual features did it on the spk-dep test subset.

Table 4.2: Top 3 individual features that yielded the highest scores for the models when concatenated with mixture features for training the models, the top three is presented by model and by test subset. The macro F1-scores (%) are also displayed together with the difference they achieved against the baseline in parentheses.

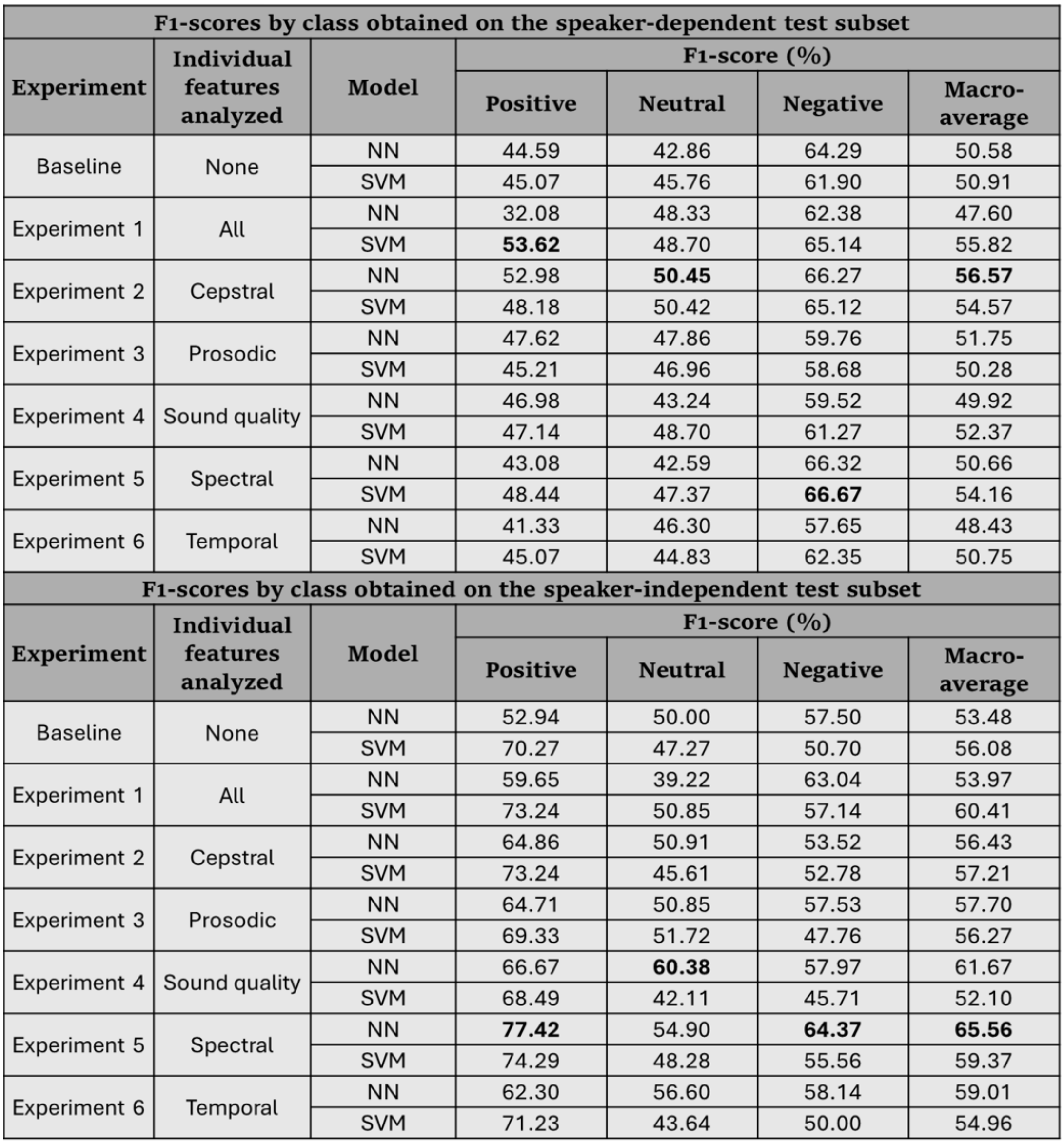


Highest macro F1-scores (%) reported on each test subset for each model				
Rank	**SVM**		**FCNN**	
	Test subset: speaker-dependent	**Test subset: speaker-independent**	**Test subset: speaker-dependent**	**Test subset: speaker-independent**
-	Baseline 50.91	Baseline 56.08	Baseline 50.58	Baseline 53.48
1	All 55.82 (+4.91)	All 60.41 (+4.33)	Cepstral 56.57 (+5.99)	Spectral 65.56 (+12.08)
2	Cepstral 54.63 (+3.73)	Spectral 59.37 (+3.29)	Prosodic 51.75 (+1.17)	Sound quality 61.67 (+8.19)
3	Spectral 54.20 (+3.30)	Cepstral 57.21 (+1.13)	Spectral 50.66 (+0.08)	Temporal 59.01 (+5.53)

To continue, a general view at the F1-scores achieved for each class across every experiment is visualized in Table 4.3. Additionally, in Table 4.4 the CMs of the experiments with the top-three results for each model on each test subset are visualized, i.e., those that appear in Table 4.2.

From the F1-scores observed in Table 4.3 is evident that, for the spk-dep subset, the Negative samples were better recognized with a maximum F1-score of 66.78% for that class achieved in Experiment 5 by the SVM model. Meanwhile, on the spk-ind subset, the Positive class achieved the highest F1-scores in most cases, with a maximum F1-score of 77.42% for that class, which was achieved by the NN model in Experiment 5.

Table 4.3: F1-scores for each class and macro averaged in each experiment on both test subsets for each model. F1-scores range from 0 to 100%, with 100% indicating perfect precission and recall for that class. Higher values indicate a good performance of the model in recognizing that class, while lower values indicate poorer performance. In bold letters are highlighted the highest scores for each class.



F1-scores by class obtained on the speaker-dependent test subset						
Experiment	**Individual features analyzed**	**Model**	**F1-score (%)**			
			Positive	**Neutral**	**Negative**	**Macro-average**
Baseline	None	NN	44.59	42.86	64.29	50.58
		SVM	45.07	45.76	61.90	50.91
Experiment 1	All	NN	32.08	48.33	62.38	47.60
		SVM	**53.62**	48.70	65.14	55.82
Experiment 2	Cepstral	NN	52.98	**50.45**	66.27	**56.57**
		SVM	48.18	50.42	65.12	54.57
Experiment 3	Prosodic	NN	47.62	47.86	59.76	51.75
		SVM	45.21	46.96	58.68	50.28
Experiment 4	Sound quality	NN	46.98	43.24	59.52	49.92
		SVM	47.14	48.70	61.27	52.37
Experiment 5	Spectral	NN	43.08	42.59	66.32	50.66
		SVM	48.44	47.37	**66.67**	54.16
Experiment 6	Temporal	NN	41.33	46.30	57.65	48.43
		SVM	45.07	44.83	62.35	50.75

F1-scores by class obtained on the speaker-independent test subset						
Experiment	**Individual features analyzed**	**Model**	**F1-score (%)**			
			Positive	**Neutral**	**Negative**	**Macro-average**
Baseline	None	NN	52.94	50.00	57.50	53.48
		SVM	70.27	47.27	50.70	56.08
Experiment 1	All	NN	59.65	39.22	63.04	53.97
		SVM	73.24	50.85	57.14	60.41
Experiment 2	Cepstral	NN	64.86	50.91	53.52	56.43
		SVM	73.24	45.61	52.78	57.21
Experiment 3	Prosodic	NN	64.71	50.85	57.53	57.70
		SVM	69.33	51.72	47.76	56.27
Experiment 4	Sound quality	NN	66.67	**60.38**	57.97	61.67
		SVM	68.49	42.11	45.71	52.10
Experiment 5	Spectral	NN	**77.42**	54.90	**64.37**	**65.56**
		SVM	74.29	48.28	55.56	59.37
Experiment 6	Temporal	NN	62.30	56.60	58.14	59.01
		SVM	71.23	43.64	50.00	54.96

The Neutral class had the lowest F1-scores on both test subsets in most experiments. The CMs in Table 4.4 corroborate the findings, the Neutral class has less correct classified samples than the other classes for all experiments, the Negative class has higher correctly classified samples by the models tested on the spk-dep subset, while the Positive class has the highest number of correctly classified samples by most of the models tested on the spk-ind test subset. According to [8], the Neutral class is the hardest to classify among the three; they also employed the VGAF database for GER and achieved the lowest scores for the Neutral class.

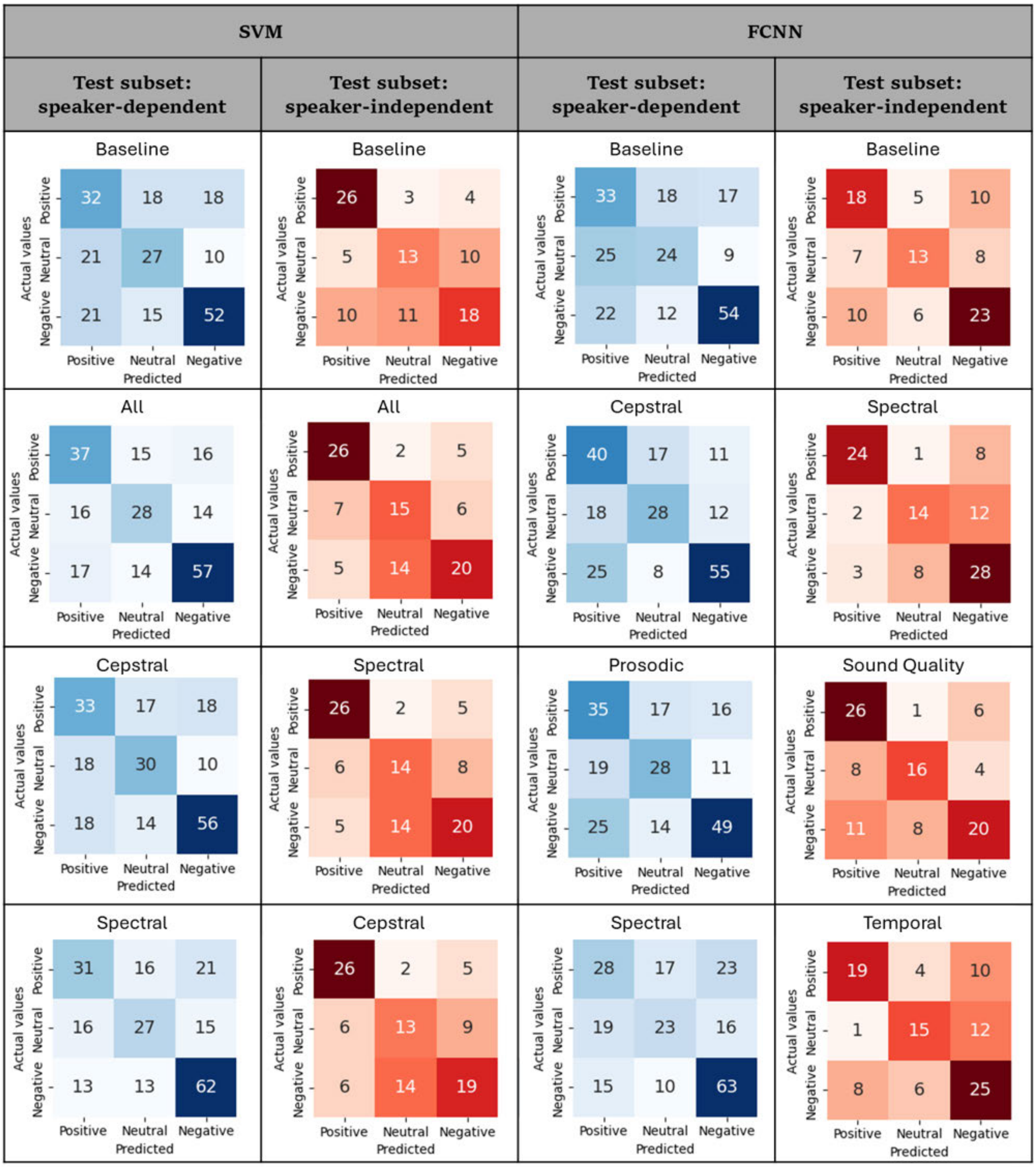
The recognition of classes made by the SVM models, as observed in the CMs contained in Table 4.4, is consistent between the baseline and the predictions of each model, which might indicate that the support vectors are mainly obtained from mixture features. This could explain why the scores did not get much higher after concatenating the different groups of individual features with the mixture features for training the model. On the other hand, the recognition rates for each class for the FCNN models vary substantially with each group of individual features on each test subset. Few improvements were achieved by the FCNN model on the spk-dep subset as observed in Figure 4.1, where only the FCNN trained with cepstral individual features, surpassed the macro F1-score of the baseline by more than 5%. Meanwhile, the biggest impact was observed on the spk-ind subset. From the F1-scores reported in Table 4.3 and the CMs for the FCNN observed in Table 4.4, it is noticeable that the individual features yield improvements in the recognition of Positive and Negative samples, but not for the Neutral samples, except for the Experiment 4 where the Neutral class achieved F1-score = 60%, which is 10% than the baseline for the Neutral class using the FCNN model.

Given the fact that there was no access to the test subset of the VGAF database and that only groups of two speakers were targeted in this work, it is not possible to make a direct comparison with the results obtained here. As summarized by Augusma et al. [7], the accuracies reported by studies where the VGAF was employed for multi-modal GER (using audio and video data) are in the range between 60 and 77%. Some of the models trained in this work achieved UAR and macro F1-scores higher than 60% by using acoustic features only, which suggests that the audio data plays a big role in GER.

Although the performance of the models could have been improved by fine-tuning methods, this was not done since the focus of this work is to investigate the effect of individual feature groups on the models under identical settings, rather than enhancing the models by tuning the hyperparameters. In general, the variations observed in the results across all experiments, as well as the low scores obtained in some of the experiments, could be attributed to several factors. One reason might the feature-fusion method. Concatenating features allows for richer feature sets and increases data diversity; however, if the signals are highly correlated, for example, when a speaker dominates most of the conversation, concatenation might introduce redundancy, negatively impacting model performance. Another source of variation could arise from the pre-trained models used for detecting the number of speakers and separating the speech signals. Despite being trained on extensive data, pre-trained models might still produce errors, particularly with real-life data. This was evident when testing pyannote with manually labeled speaker samples, observed in Figure 3.3 in Section 3, where not all samples were correctly classified. Nevertheless, the performance remained acceptable and better than the other two tools tested. If an error occurs in detecting the number of speakers, it may propagate to the speech separation process, as the model used here was trained to separate mixtures of two speeches. Additionally, as discussed in Section 2.7, studies have indicated that emotional-speech mixtures affect speech separation systems, reducing their performance. Although the impact on BSS systems was described as "relatively small" [105], it could still degrade the performance of speech separation systems, specially with real-life speech data.

Finally, it cannot be overlooked that the audio samples were derived from an "in-the-wild" database, where sample variability is high due to real-life audio data recorded under different conditions (different languages, recording devices, speakers, background noises, etc.). These data introduce numerous variation factors, making it more challenging for the models to generalize. Since only the results obtained by the FCNN on the spk-ind test subset were identified as statistically significant at a significance level of 0.05. This indicates that, despite higher accuracy values obtained in some experiments, the differences in the scores could still be attributed to random variations rather than genuine enhancements in model performance. Therefore, a bigger sample size for testing is needed to validate the improvements.

Table 4.4: Confusion matrices of the predictions of the baseline models and the predictions of the models with the three highest scores for each test subset. Each confusion matrix is identified with the category of individual features employed for training the model.



5 Future work

Several key areas for future work and improvement in GER systems have been identified. One critical need is the development of group emotion databases based on speech data. Current methodologies, including this study, often rely on speech signals extracted from videos, which are labeled according to the perceived emotions of annotators. This labeling process may be biased by visual information. Therefore, creating group emotion databases with speech data, labeled solely based on audio content, could enhance the accuracy and effectiveness of GER systems.

There is a general need for more group emotion databases to serve as benchmarks and facilitate the comparison of GER systems. Additional benchmark databases would enable researchers to test their models on diverse data, which is essential for drawing robust conclusions about model performance. Moreover, there is a need for better-labeled group emotion databases that provide detailed information about individuals in the group, such as the number of speakers in audio mixtures, individual emotions, language, and gender. This detailed information could be used to enhance the performance and adaptability of GER systems for different numbers of group members.

Future research on GER should prioritize groups with more than two speakers. A larger number of group members translates to more overlapping voices that require separation and processing, and concatenating features can lead to long feature sets that increase the complexity of the systems. Consequently, alternative feature-fusion methods and feature reduction techniques should be explored.

While the use of spectral-individual features in training the FCNN model yielded a statistically significant enhancement in GER, other feature categories deserve further exploration. Future studies could investigate training GER models with the same category of individual and mixture features, or by integrating different categories of acoustic features to assess if their combined effect yields a higher impact on GER. Additionally, future research could focus on determining which parameter functionals within the spectral features are the most impactful for GER. Future research could reimplement similar settings to those used in this work, utilizing alternative feature sets commonly used in SER and exploring feature categories beyond the ComParE 2013 feature set.

Finally, although SVMs have been widely used and effective in individual-level SER, they are less explored in GER, where DNNs are more common. This study demonstrated that SVMs can also yield improvements and consistent performance across test subsets. Therefore, future research on group-level SER should employ SVMs, experimenting with different kernels and fine-tuning methods. Similarly, other machine learning models and neural network architectures should be tested and optimized.

6 Conclusion

This work explored how different features extracted from individual speakers contribute to improving the performance of an SVM and an FCNN model for speech-based GER for known and unknown speakers. The contribution of the individual features was analyzed by category: cepstral, spectral, prosodic, temporal, and sound quality features.

A baseline was established by training the models separately on 6373 acoustic features extracted at the group context, referred to as mixture features due to being obtained from speech mixtures. The features consisted of functionals applied to LLDs of the five categories of acoustic features. The speech mixtures were processed by a pre-trained model for speech separation to isolate the speech of each speaker in the mixture. Since the speech mixtures consisted of two speakers, two separate speech signals were obtained as output of the pre-trained model. For each separated speech, the same 6373 features were extracted as in the baseline; these features were referred to as individual features, since they were derived from the speech of the individual speakers. Then, the individual features were grouped into the five different categories abovementioned and were used for developing feature sets together with the mixture features.

Six different experiments were conducted where the models were trained separately using different feature sets composed of mixture features and individual features. The first experiment intended to explore the impact on the performance of the models when trained on all groups of individual features concatenated with all mixture features. In experiments 2 to 6, the same two models were trained, but only using a specific category of individual features concatenated with all of the mixture features, aiming to explore the influence of each category of individual features on the performance of the GER models. As a baseline, the models were trained exclusively using mixture features and tested on both test subsets, spk-dep and spk-ind.

The use of all individual features concatenated with mixture features improved the performance of the SVM model on both test subsets, surpassing the UAR baselines by 4.92% and 4.09%, respectively. Spectral- and cepstral-individual features also improved the performance of the SVM on both test subsets, albeit to a lesser extent. Although none of the improvements achieved with the SVM models was considered statistically significant, the consistent results of the SVM on both test subsets indicate potential for further study of these two types of features and their impact on SVM for GER.

For the FCNN models, the groups of individual features that demonstrated improvements varied across each test subset. For the test subset with known speakers (spk-dep), cepstral-individual features concatenated with all mixture features had the highest impact (UAR = 56.53%), resulting in a 6.11% higher UAR than the baseline. These were followed by prosodic and spectral-individual features, which showed minimal improvements in UAR, with increases of 1.39% and 0.38%, respectively. On the test subset with unknown speakers (spk-ind), every feature group achieved improvement on the performance of the FCNN model. Spectral-individual features had the most substantial impact (11.52% UAR above the baseline), followed by sound quality and temporal-individual features, which surpassed the UAR of the baseline by 9.09% and 5.1%, respectively. These results indicate the potential of including these categories of features from the individual speakers as input for training FCNNs for real-life applications of GER.

Emotion recognition tasks are inherently complex, particularly when performed at a group level using real-life data, which exhibit greater diversity and complexity compared to data recorded under controlled environments. While higher scores have been achieved with multi-modal approaches that integrate both audio and visual data, it is noteworthy what can be accomplished through the analysis of audio data alone for GER. As demonstrated here, macro F1-scores up to 65% are achievable by only analyzing speech. This evidence supports the relevance of studying speech-based GER and the potential improvement of GER systems that can be achieved by incorporating individual speaker features as inputs during model training.

Bibliography

[1] Serdar Yildirim, Murtaza Bulut, Chul Lee, Abe Kazemzadeh, Sungbok Lee, Shrikanth Narayanan, and Carlos Busso. *An acoustic study of emotions expressed in speech.* October 2004. doi: 10.21437/Interspeech.2004-242.

[2] Qing Zhu, Qirong Mao, Jialin Zhang, Xiaohua Huang, and Wenming Zheng. Towards A Robust Group-level Emotion Recognition via Uncertainty-Aware Learning, October 2023. URL `http://arxiv.org/abs/2310.04306`. arXiv:2310.04306 [cs].

[3] Moataz El Ayadi, Mohamed S. Kamel, and Fakhri Karray. Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, 44(3):572–587, March 2011. ISSN 0031-3203. doi: 10.1016/j.patcog.2010.09.020. URL `https://www.sciencedirect.com/science/article/pii/S0031320310004619`.

[4] B. Schuller, G. Rigoll, and M. Lang. Speech emotion recognition combining acoustic features and linguistic information in a hybrid support vector machine-belief network architecture. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–577, May 2004. doi: 10.1109/ICASSP.2004.1326051. URL `https://ieeexplore.ieee.org/document/1326051/`. ISSN: 1520-6149.

[5] Umut Avci. Speech Emotion Recognition Using Spectrogram Patterns as Features. In Alexey Karpov and Rodmonga Potapova, editors, *Speech and Computer*, pages 57–67, Cham, 2020. Springer International Publishing. ISBN 978-3-030-60276-5. doi: 10.1007/978-3-030-60276-5_6.

[6] Kyuhong Lee and Taeyong Kim. Group emotion recognition based on psychological principles using a fuzzy system. *The Visual Computer*, 40(5):3503–3514, May 2024. ISSN 1432-2315. doi: 10.1007/s00371-023-03048-w. URL `https://doi.org/10.1007/s00371-023-03048-w`.

[7] Anderson Augusma, Dominique Vaufreydaz, and Frédérique Letué. Multimodal Group Emotion Recognition In-the-wild Using Privacy-Compliant Features. In *INTERNATIONAL CONFERENCE ON MULTIMODAL INTERACTION*, pages 750–754, October 2023. doi: 10.1145/3577190.3616546. URL `http://arxiv.org/abs/2312.05265`. arXiv:2312.05265 [cs].

[8] Anastasia Petrova, Dominique Vaufreydaz, and Philippe Dessus. Group-Level Emotion Recognition Using a Unimodal Privacy-Safe Non-Individual Approach. In *Proceedings of the 2020 International Conference on Multimodal Interaction*, ICMI '20, pages 813–820, New York, NY, USA, October 2020. Association for Computing Machinery. ISBN 978-1-4503-7581-8. doi: 10.1145/3382507.3417969. URL `https://dl.acm.org/doi/10.1145/3382507.3417969`.

[9] Kha Gia Quach, Ngan Le, Chi Nhan Duong, Ibsa Jalata, Kaushik Roy, and Khoa Luu. Non-volume preserving-based fusion to group-level emotion recognition on crowd videos. *Pattern Recognition*, 128:108646, August 2022. ISSN 00313203. doi: 10.1016/j.patcog.2022.108646. URL `https://linkinghub.elsevier.com/retrieve/pii/S0031320322001273`.

[10] Xingzhi Wang, Dong Zhang, Hong-Zhou Tan, and Dah-Jye Lee. A Self-Fusion Network Based on Contrastive Learning for Group Emotion Recognition. *IEEE Transactions on Computational Social Systems*, 10(2):458–469, April 2023. ISSN 2329-924X. doi: 10.1109/TCSS.2022.3202249. URL `https://ieeexplore.ieee.org/document/9887975`. Conference Name: IEEE Transactions on Computational Social Systems.

[11] Kai Wang, Xiaoxing Zeng, Jianfei Yang, Debin Meng, Kaipeng Zhang, Xiaojiang Peng, and Yu Qiao. Cascade Attention Networks For Group Emotion Recognition with Face, Body and Image Cues. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, ICMI '18, pages 640–645, New York, NY, USA, October 2018. Association for Computing Machinery. ISBN 978-1-4503-5692-3. doi: 10.1145/3242969.3264991. URL `https://doi.org/10.1145/3242969.3264991`.

[12] Yu Wang. ConGNN: Context-consistent cross-graph neural network for group emotion recognition in the wild. 2022.

[13] Xin Guo, Luisa F. Polania, Bin Zhu, Charles Boncelet, and Kenneth E. Barner. Graph Neural Networks for Image Understanding Based on Multiple Cues: Group Emotion Recognition and Event Recognition as Use Cases, February 2020. URL `http://arxiv.org/abs/1909.12911`. arXiv:1909.12911 [cs].

[14] Sandra Ottl, Shahin Amiriparian, Maurice Gerczuk, Vincent Karas, and Björn Schuller. Group-level Speech Emotion Recognition Utilising Deep Spectrum Features. In *Proceedings of the 2020 International Conference on Multimodal Interaction*, ICMI '20, pages 821–826, New York, NY, USA, October 2020. Association for Computing Machinery. ISBN 978-1-4503-7581-8. doi: 10.1145/3382507.3417964. URL `https://doi.org/10.1145/3382507.3417964`.

[15] Mo Sun, Jian Li, Hui Feng, Wei Gou, Haifeng Shen, Jian Tang, Yi Yang, and Jieping Ye. Multi-modal Fusion Using Spatio-temporal and Static Features for Group Emotion Recognition. In *Proceedings of the 2020 International Conference on Multimodal Interaction*, ICMI '20, pages 835–840, New York, NY, USA,

October 2020. Association for Computing Machinery. ISBN 978-1-4503-7581-8. doi: 10.1145/3382507.3417971. URL https://dl.acm.org/doi/10.1145/3382507.3417971.

[16] Sunan Li, Hailun Lian, Cheng Lu, Yan Zhao, Chuangao Tang, Yuan Zong, and Wenming Zheng. Audio-Visual Group-based Emotion Recognition using Local and Global Feature Aggregation based Multi-Task Learning. In *Proceedings of the 25th International Conference on Multimodal Interaction*, ICMI '23, pages 741–745, New York, NY, USA, October 2023. Association for Computing Machinery. ISBN 979-8-4007-0055-2. doi: 10.1145/3577190.3616544. URL https://doi.org/10.1145/3577190.3616544.

[17] Tatsuya Hayamizu, Sano Mutsuo, Kenzaburo Miyawaki, Hiroaki Mori, Satoshi Nishiguchi, and Nobuyuki Yamashita. Group emotion estimation using Bayesian network based on facial expression and prosodic information. In *2012 IEEE International Conference on Control System, Computing and Engineering*, pages 177–182, November 2012. doi: 10.1109/ICCSCE.2012.6487137. URL https://ieeexplore.ieee.org/document/6487137/?arnumber=6487137.

[18] Xiaohua Huang, Jinke Xu, Wenming Zheng, Qirong Mao, and Abhinav Dhall. A Survey of Deep Learning for Group-level Emotion Recognition, August 2024. URL http://arxiv.org/abs/2408.15276. arXiv:2408.15276 [cs].

[19] Emmeke A. Veltmeijer, Charlotte Gerritsen, and Koen V. Hindriks. Automatic Emotion Recognition for Groups: A Review. *IEEE Transactions on Affective Computing*, 14(1):89–107, January 2023. ISSN 1949-3045. doi: 10.1109/TAFFC.2021.3065726. URL https://ieeexplore.ieee.org/document/9376944. Conference Name: IEEE Transactions on Affective Computing.

[20] Simon J.D. Prince. *Understanding Deep Learning*. The MIT Press, November 2024. URL http://udlbook.com.

[21] Qi Wang, Yue Ma, Kun Zhao, and Yingjie Tian. A Comprehensive Survey of Loss Functions in Machine Learning. *Annals of Data Science*, 9(2):187–212, April 2022. ISSN 2198-5812. doi: 10.1007/s40745-020-00253-5. URL https://doi.org/10.1007/s40745-020-00253-5.

[22] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, September 1995. ISSN 1573-0565. doi: 10.1007/BF00994018. URL https://doi.org/10.1007/BF00994018.

[23] Christopher M. Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer, New York, 2006. ISBN 978-0-387-31073-2.

[24] Support Vector Machines: All you need to know! URL https://www.youtube.com/watch?v=ny1iZ5A8ilA.

[25] Anil Ananthaswamy. *Why machines learn: the elegant math behind modern AI*. Dutton, New York, 2024. ISBN 978-0-593-18574-2.

[26] Prajakta P. Dahake, Kailash Shaw, and P. Malathi. Speaker dependent speech emotion recognition using MFCC and Support Vector Machine. In *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, pages 1080–1084, September 2016. doi: 10.1109/ICACDOT.2016.7877753. URL `https://ieeexplore.ieee.org/document/7877753/`.

[27] Scikit learn developers. Support Vector Machines. URL `https://scikit-learn.org/stable/modules/svm.html`.

[28] Josiah Adesola. SVM Kernels Explained: How to Tackle Nonlinear Data in Machine Learning, June 2025. URL `https://www.freecodecamp.org/news/svm-kernels-how-to-tackle-nonlinear-data-in-machine-learning/`.

[29] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4): 115–133, December 1943. ISSN 1522-9602. doi: 10.1007/BF02478259. URL `https://doi.org/10.1007/BF02478259`.

[30] A. Géron. *Neural networks and deep learning*. O'Reilly, 2018. URL `https://books.google.de/books?id=5pm6tQEACAAJ`.

[31] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958. ISSN 1939-1471, 0033-295X. doi: 10.1037/h0042519. URL `https://doi.apa.org/doi/10.1037/h0042519`.

[32] Sally Antoin Jerjees, Hala Jassim Mohammed, Hayder Saadi Radeaf, Basheera M. Mahmmod, and Sadiq H. Abdulhussain. Deep Learning-Based Speech Enhancement Algorithm Using Charlier Transform. In *2023 15th International Conference on Developments in eSystems Engineering (DeSE)*, pages 100–105, January 2023. doi: 10.1109/DeSE58274.2023.10099854. URL `https://ieeexplore.ieee.org/document/10099854/`.

[33] Aharon Satt, Shai Rozenberg, and Ron Hoory. Efficient Emotion Recognition from Speech Using Deep Learning on Spectrograms. In *Interspeech 2017*, pages 1089–1093. ISCA, August 2017. doi: 10.21437/Interspeech.2017-200. URL `https://www.isca-archive.org/interspeech_2017/satt17_interspeech.html`.

[34] Shiqing Zhang, Yijiao Yang, Chen Chen, Xingnan Zhang, Qingming Leng, and Xiaoming Zhao. Deep learning-based multimodal emotion recognition from audio, visual, and text modalities: A systematic review of recent advancements and future prospects. *Expert Systems with Applications*, 237:121692, March 2024. ISSN 0957-4174. doi: 10.1016/j.eswa.2023.121692. URL `https://www.sciencedirect.com/science/article/pii/S0957417423021942`.

[35] Tae Jin Park, Naoyuki Kanda, Dimitrios Dimitriadis, Kyu J. Han, Shinji Watanabe, and Shrikanth Narayanan. A Review of Speaker Diarization: Recent Advances with Deep Learning, November 2021. URL http://arxiv.org/abs/2101.09624. arXiv:2101.09624 [eess].

[36] Classification: Accuracy, recall, precision, and related metrics | Machine Learning. URL https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall.

[37] Krystian Hartmann. Unlocking the language: Key features of emotions. *Acta Psychologica*, 251:104628, November 2024. ISSN 0001-6918. doi: 10.1016/j.actpsy.2024.104628. URL https://www.sciencedirect.com/science/article/pii/S0001691824005067.

[38] Rainer Reisenzein. What is a definition of emotion? And are emotions mental-behavioral processes? *Social Science Information*, 46(3):424–428, September 2007. ISSN 0539-0184. doi: 10.1177/05390184070460030110. URL https://doi.org/10.1177/05390184070460030110. Publisher: SAGE Publications Ltd.

[39] Michael Inzlicht, Bruce D. Bartholow, and Jacob B. Hirsh. Emotional foundations of cognitive control. *Trends in Cognitive Sciences*, 19(3):126–132, March 2015. ISSN 1364-6613. doi: 10.1016/j.tics.2015.01.004. URL https://www.sciencedirect.com/science/article/pii/S1364661315000054.

[40] Mehmet Berkehan Akçay and Kaya Oğuz. Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers. *Speech Communication*, 116:56–76, January 2020. ISSN 0167-6393. doi: 10.1016/j.specom.2019.12.001. URL https://www.sciencedirect.com/science/article/pii/S0167639319302262.

[41] Wikimedia Commons. Plutchik's wheel of emotions, 2012. URL https://commons.wikimedia.org/wiki/File:Plutchik-wheel_de.svg. tex.urlseen: 18-03-25.

[42] Paul Ekman. An argument for basic emotions. *Cognition and Emotion*, 6(3-4):169–200, May 1992. ISSN 0269-9931, 1464-0600. doi: 10.1080/02699939208411068. URL https://www.tandfonline.com/doi/full/10.1080/02699939208411068.

[43] EMOTION: Theory, Research, and Experience. In Robert Plutchik and Henry Kellerman, editors, *Theories of Emotion*, page ii. Academic Press, January 1980. ISBN 978-0-12-558701-3. doi: 10.1016/B978-0-12-558701-3.50001-6. URL https://www.sciencedirect.com/science/article/pii/B9780125587013500016.

[44] Albert Mehrabian. Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in Temperament. *Current Psychol-*

ogy, 14(4):261–292, December 1996. ISSN 1936-4733. doi: 10.1007/BF02686918. URL https://doi.org/10.1007/BF02686918.

[45] James A Russell and Albert Mehrabian. Evidence for a three-factor theory of emotions. *Journal of Research in Personality*, 11(3):273–294, September 1977. ISSN 0092-6566. doi: 10.1016/0092-6566(77)90037-X. URL https://www.sciencedirect.com/science/article/pii/009265667790037X.

[46] James A. Russell. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178, 1980. ISSN 1939-1315. doi: 10.1037/h0077714. Place: US Publisher: American Psychological Association.

[47] Lisa Feldman Barrett and James A. Russell. Independence and bipolarity in the structure of current affect. *Journal of Personality and Social Psychology*, 74(4): 967–984, 1998. ISSN 1939-1315. doi: 10.1037/0022-3514.74.4.967. Place: US Publisher: American Psychological Association.

[48] Bagus Tris Atmaja, Akira Sasou, and Masato Akagi. Survey on bimodal speech emotion recognition from acoustic and linguistic information fusion. *Speech Communication*, 140:11–28, May 2022. ISSN 01676393. doi: 10.1016/j.specom.2022.03.002. URL https://linkinghub.elsevier.com/retrieve/pii/S0167639322000413.

[49] Magda B. Arnold. *Emotion and personality.* New York, Columbia University Press, 1960. URL http://archive.org/details/emotionpersonali01arno.

[50] Garima Sharma, Shreya Ghosh, and Abhinav Dhall. Automatic Group Level Affect and Cohesion Prediction in Videos. In *2019 8th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, pages 161–167, September 2019. doi: 10.1109/ACIIW.2019.8925231. URL https://ieeexplore.ieee.org/document/8925231.

[51] Shaundra B. Daily, Melva T. James, David Cherry, John J. Porter, Shelby S. Darnell, Joseph Isaac, and Tania Roy. Chapter 9 - Affective Computing: Historical Foundations, Current Applications, and Future Trends. In Myounghoon Jeon, editor, *Emotions and Affect in Human Factors and Human-Computer Interaction*, pages 213–231. Academic Press, San Diego, January 2017. ISBN 978-0-12-801851-4. doi: 10.1016/B978-0-12-801851-4.00009-4. URL https://www.sciencedirect.com/science/article/pii/B9780128018514000094.

[52] R W Picard. Affective Computing. *MIT Media laboratory perceptual computing section technical report no. 321*, 92:2139, 1995. URL https://vismod.media.mit.edu/pub/tech-reports/TR-321.pdf.

[53] Rosalind W. Picard. *Affective Computing.* The MIT Press, September 1997. ISBN 978-0-262-28158-4. doi: 10.7551/mitpress/1140.001.0001. URL https://direct.mit.edu/books/monograph/4296/Affective-Computing.

[54] Yeşim Ülgen Sönmez and Asaf Varol. In-depth investigation of speech emotion recognition studies from past to present –The importance of emotion recognition from speech signal for AI–. *Intelligent Systems with Applications*, 22:200351, June 2024. ISSN 26673053. doi: 10.1016/j.iswa.2024.200351. URL `https://linkinghub.elsevier.com/retrieve/pii/S2667305324000279`.

[55] Smith K. Khare, Victoria Blanes-Vidal, Esmaeil S. Nadimi, and U. Rajendra Acharya. Emotion recognition and artificial intelligence: A systematic review (2014–2023) and research recommendations. *Information Fusion*, 102:102019, February 2024. ISSN 15662535. doi: 10.1016/j.inffus.2023.102019. URL `https://linkinghub.elsevier.com/retrieve/pii/S1566253523003354`.

[56] Maria Egger, Matthias Ley, and Sten Hanke. Emotion Recognition from Physiological Signal Analysis: A Review. *Electronic Notes in Theoretical Computer Science*, 343:35–55, May 2019. ISSN 1571-0661. doi: 10.1016/j.entcs.2019.04.009. URL `https://www.sciencedirect.com/science/article/pii/S157106611930009X`.

[57] Lin Shu, Jinyan Xie, Mingyue Yang, Ziyi Li, Zhenqi Li, Dan Liao, Xiangmin Xu, and Xinyi Yang. A Review of Emotion Recognition Using Physiological Signals. *Sensors*, 18(7):2074, July 2018. ISSN 1424-8220. doi: 10.3390/s18072074. URL `https://www.mdpi.com/1424-8220/18/7/2074`. Number: 7 Publisher: Multidisciplinary Digital Publishing Institute.

[58] J. Wagner, Jonghwa Kim, and E. Andre. From Physiological Signals to Emotions: Implementing and Comparing Selected Methods for Feature Extraction and Classification. In *2005 IEEE International Conference on Multimedia and Expo*, pages 940–943, July 2005. doi: 10.1109/ICME.2005.1521579. URL `https://ieeexplore.ieee.org/document/1521579/`. ISSN: 1945-788X.

[59] Ayoub Ghriss, Bo Yang, Viktor Rozgic, Elizabeth Shriberg, and Chao Wang. Sentiment-Aware Automatic Speech Recognition Pre-Training for Enhanced Speech Emotion Recognition. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7347–7351, May 2022. doi: 10.1109/ICASSP43922.2022.9747637. URL `https://ieeexplore.ieee.org/document/9747637/?arnumber=9747637`. ISSN: 2379-190X.

[60] Smiley Blanton. The voice and the emotions. *Quarterly Journal of Speech*, 1(2):154–172, July 1915. ISSN 0033-5630. doi: 10.1080/00335631509360475. URL `https://doi.org/10.1080/00335631509360475`. Publisher: NCA Website _eprint: https://doi.org/10.1080/00335631509360475.

[61] John Decatur Williamson. Speech analyzer for analyzing pitch or frequency perturbations in individual speech pattern to determine the emotional state of the person, June 1978. URL `https://patents.google.com/patent/US4093821A/en`.

[62] Björn Schuller, Stefan Steidl, Anton Batliner, Alessandro Vinciarelli, Klaus Scherer, Fabien Ringeval, Mohamed Chetouani, Felix Weninger, Florian Eyben, Erik Marchi, Marcello Mortillaro, Hugues Salamin, Anna Polychroniou, Fabio Valente, and Samuel Kim. The INTERSPEECH 2013 computational paralinguistics challenge: social signals, conflict, emotion, autism. In *Interspeech 2013*, pages 148–152. ISCA, August 2013. doi: 10.21437/Interspeech.2013-56. URL `https://www.isca-archive.org/interspeech_2013/schuller13_interspeech.html`.

[63] Björn Schuller, Felix Weninger, Yue Zhang, Fabien Ringeval, Anton Batliner, Stefan Steidl, Florian Eyben, Erik Marchi, Alessandro Vinciarelli, Klaus Scherer, Mohamed Chetouani, and Marcello Mortillaro. Affective and behavioural computing: Lessons learnt from the First Computational Paralinguistics Challenge. *Computer Speech & Language*, 53:156–180, January 2019. ISSN 0885-2308. doi: 10.1016/j.csl.2018.02.004. URL `https://www.sciencedirect.com/science/article/pii/S0885230816303928`.

[64] Abhinav Dhall, Roland Goecke, Jyoti Joshi, Michael Wagner, and Tom Gedeon. *Emotion Recognition In The Wild Challenge (EmotiW) challenge and workshop summary*. December 2013. doi: 10.1145/2522848.2531749. Journal Abbreviation: ICMI 2013 - Proceedings of the 2013 ACM International Conference on Multimodal Interaction Pages: 372 Publication Title: ICMI 2013 - Proceedings of the 2013 ACM International Conference on Multimodal Interaction.

[65] Abhinav Dhall, Monisha Singh, Roland Goecke, Tom Gedeon, Donghuo Zeng, Yanan Wang, and Kazushi Ikeda. EmotiW 2023: Emotion Recognition in the Wild Challenge. In *Proceedings of the 25th International Conference on Multimodal Interaction*, ICMI '23, pages 746–749, New York, NY, USA, October 2023. Association for Computing Machinery. ISBN 979-8-4007-0055-2. doi: 10.1145/3577190.3616545. URL `https://doi.org/10.1145/3577190.3616545`.

[66] Abhinav Dhall, Garima Sharma, Roland Goecke, and Tom Gedeon. EmotiW 2020: Driver Gaze, Group Emotion, Student Engagement and Physiological Signal based Challenges. In *Proceedings of the 2020 International Conference on Multimodal Interaction*, ICMI '20, pages 784–789, New York, NY, USA, October 2020. Association for Computing Machinery. ISBN 978-1-4503-7581-8. doi: 10.1145/3382507.3417973. URL `https://dl.acm.org/doi/10.1145/3382507.3417973`.

[67] Björn Schuller, Stefan Steidl, Anton Batliner, Alessandro Vinciarelli, Klaus Scherer, Fabien Ringeval, Mohamed Chetouani, Felix Weninger, Florian Eyben, Erik Marchi, Marcello Mortillaro, Hugues Salamin, Anna Polychroniou, Fabio Valente, and Samuel Kim. The INTERSPEECH 2013 computational paralinguistics challenge: social signals, conflict, emotion, autism. In *Interspeech 2013*, pages 148–152. ISCA, August 2013. doi: 10.21437/

Interspeech.2013-56. URL https://www.isca-archive.org/interspeech_2013/schuller13_interspeech.html.

[68] Björn W. Schuller, Anton Batliner, Shahin Amiriparian, Christian Bergler, Maurice Gerczuk, Natalie Holz, Pauline Larrouy-Maestri, Sebastian P. Bayerl, Korbinian Riedhammer, Adria Mallol-Ragolta, Maria Pateraki, Harry Coppock, Ivan Kiskin, Marianne Sinka, and Stephen Roberts. The ACM Multimedia 2022 Computational Paralinguistics Challenge: Vocalisations, Stuttering, Activity, & Mosquitoes, May 2022. URL http://arxiv.org/abs/2205.06799. arXiv:2205.06799 [cs].

[69] Christos-Nikolaos Anagnostopoulos, Theodoros Iliou, and Ioannis Giannoukos. Features and classifiers for emotion recognition from speech: a survey from 2000 to 2011. *Artificial Intelligence Review*, 43(2):155–177, February 2015. ISSN 1573-7462. doi: 10.1007/s10462-012-9368-5. URL https://doi.org/10.1007/s10462-012-9368-5.

[70] Samaneh Madanian, Talen Chen, Olayinka Adeleye, John Michael Templeton, Christian Poellabauer, Dave Parry, and Sandra L. Schneider. Speech emotion recognition using machine learning — A systematic review. *Intelligent Systems with Applications*, 20:200266, November 2023. ISSN 26673053. doi: 10.1016/j.iswa.2023.200266. URL https://linkinghub.elsevier.com/retrieve/pii/S2667305323000911.

[71] Youddha Beer Singh and Shivani Goel. Survey on Human Emotion Recognition: Speech Database, Features and Classification. In *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pages 298–301, October 2018. doi: 10.1109/ICACCCN.2018.8748379. URL https://ieeexplore.ieee.org/document/8748379/?arnumber=8748379.

[72] Farah Chenchah and Zied Lachiri. Speech Emotion Recognition in Acted and Spontaneous Context. *Procedia Computer Science*, 39:139–145, 2014. ISSN 18770509. doi: 10.1016/j.procs.2014.11.020. URL https://linkinghub.elsevier.com/retrieve/pii/S1877050914014380.

[73] Wenwu Wang, editor. *Machine Audition: Principles, Algorithms and Systems.* IGI Global, 2011. ISBN 978-1-61520-919-4 978-1-61520-920-0. doi: 10.4018/978-1-61520-919-4. URL http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-61520-919-4.

[74] Reza Lotfian and Carlos Busso. Building Naturalistic Emotionally Balanced Speech Corpus by Retrieving Emotional Speech from Existing Podcast Recordings. *IEEE Transactions on Affective Computing*, 10(4):471–483, October 2019. ISSN 1949-3045. doi: 10.1109/TAFFC.2017.2736999. URL https://ieeexplore.ieee.org/document/8003425/?arnumber=8003425. Conference Name: IEEE Transactions on Affective Computing.

[75] Silke Steininger, Susen Rabold, Olga Dioubina, and Florian Schiel. Development of the User-State Conventions for the Multimodal Corpus in SmartKom.

[76] T. Vogt and E. Andre. Comparing Feature Sets for Acted and Spontaneous Speech in View of Automatic Emotion Recognition. In *2005 IEEE International Conference on Multimedia and Expo*, pages 474–477, Amsterdam, The Netherlands, 2005. IEEE. ISBN 978-0-7803-9331-8. doi: 10.1109/ICME.2005.1521463. URL http://ieeexplore.ieee.org/document/1521463/.

[77] Florian Eyben, Klaus R. Scherer, Björn W. Schuller, Johan Sundberg, Elisabeth André, Carlos Busso, Laurence Y. Devillers, Julien Epps, Petri Laukka, Shrikanth S. Narayanan, and Khiet P. Truong. The Geneva Minimalistic Acoustic Parameter Set (GeMAPS) for Voice Research and Affective Computing. *IEEE Transactions on Affective Computing*, 7(2):190–202, April 2016. ISSN 1949-3045. doi: 10.1109/TAFFC.2015.2457417. URL https://ieeexplore.ieee.org/document/7160715/?arnumber=7160715. Conference Name: IEEE Transactions on Affective Computing.

[78] Marko Lugger and Bin Yang. AN INCREMENTAL ANALYSIS OF DIFFERENT FEATURE GROUPS IN SPEAKER INDEPENDENT EMOTION RECOGNITION. 2007.

[79] Elliot Moore II, Mark A. Clements, John W. Peifer, and Lydia Weisser. Critical Analysis of the Impact of Glottal Features in the Classification of Clinical Depression in Speech. *IEEE Transactions on Biomedical Engineering*, 55(1): 96–107, January 2008. ISSN 1558-2531. doi: 10.1109/TBME.2007.900562. URL https://ieeexplore.ieee.org/document/4360055/.

[80] Wenming Zheng, Minghai Xin, Xiaolan Wang, and Bei Wang. A Novel Speech Emotion Recognition Method via Incomplete Sparse Least Square Regression. *IEEE Signal Processing Letters*, 21(5):569–572, May 2014. ISSN 1558-2361. doi: 10.1109/LSP.2014.2308954. URL https://ieeexplore.ieee.org/document/6750037/?arnumber=6750037. Conference Name: IEEE Signal Processing Letters.

[81] Jilt Sebastian and Piero Pierucci. Fusion Techniques for Utterance-Level Emotion Recognition Combining Speech and Transcripts. In *Interspeech 2019*, pages 51–55. ISCA, September 2019. doi: 10.21437/Interspeech.2019-3201. URL https://www.isca-archive.org/interspeech_2019/sebastian19_interspeech.html.

[82] Daniel Neiberg, Kjell Elenius, and Laskowski. *Emotion Recognition in Spontaneous Speech Using GMMs.* September 2006. doi: 10.21437/Interspeech.2006-277.

[83] Florian Eyben, Martin Wöllmer, and Björn Schuller. *openSMILE – The Munich Versatile and Fast Open-Source Audio Feature Extractor.* January 2010. doi:

10.1145/1873951.1874246. Journal Abbreviation: MM'10 - Proceedings of the ACM Multimedia 2010 International Conference Pages: 1462 Publication Title: MM'10 - Proceedings of the ACM Multimedia 2010 International Conference.

[84] Paul Boersma and David Weenink. Praat: doing phonetics by computer, 1992.

[85] Brian McFee, Colin Raffel, Dawen Liang, Daniel Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and Music Signal Analysis in Python. pages 18–24, Austin, Texas, 2015. doi: 10.25080/Majora-7b98e3ed-003. URL https://doi.curvenote.com/10.25080/Majora-7b98e3ed-003.

[86] Pavol Harar, Radim Burget, and Malay Kishore Dutta. *Speech emotion recognition with deep learning*. February 2017. doi: 10.1109/SPIN.2017.8049931. Pages: 140.

[87] Xin Guo, Bin Zhu, Luisa F. Polanía, Charles Boncelet, and Kenneth E. Barner. Group-Level Emotion Recognition Using Hybrid Deep Models Based on Faces, Scenes, Skeletons and Visual Attentions. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, ICMI '18, pages 635–639, New York, NY, USA, October 2018. Association for Computing Machinery. ISBN 978-1-4503-5692-3. doi: 10.1145/3242969.3264990. URL https://dl.acm.org/doi/10.1145/3242969.3264990.

[88] Sandeep Kumar Pandey, Hanumant Singh Shekhawat, and S.R.M. Prasanna. Multi-cultural speech emotion recognition using language and speaker cues. *Biomedical Signal Processing and Control*, 83:104679, May 2023. ISSN 17468094. doi: 10.1016/j.bspc.2023.104679. URL https://linkinghub.elsevier.com/retrieve/pii/S174680942300112X.

[89] Bin Zhu, Xinjie Lan, Xin Guo, Kenneth E. Barner, and Charles Boncelet. Multi-rate Attention Based GRU Model for Engagement Prediction. In *Proceedings of the 2020 International Conference on Multimodal Interaction*, ICMI '20, pages 841–848, New York, NY, USA, October 2020. Association for Computing Machinery. ISBN 978-1-4503-7581-8. doi: 10.1145/3382507.3417965. URL https://dl.acm.org/doi/10.1145/3382507.3417965.

[90] W. Wundt (1907): Outlines of Psychology - Contents. URL https://psychologie.lw.uni-leipzig.de/wundt/opera/wundt/OLiPsych/OLiPsyIn.htm.

[91] Gerben A. Van Kleef and Agneta H. Fischer. Emotional collectives: How groups shape emotions and emotions shape groups. *Cognition and Emotion*, 30(1):3–19, January 2016. ISSN 0269-9931, 1464-0600. doi: 10.1080/02699931.2015.1081349. URL http://www.tandfonline.com/doi/full/10.1080/02699931.2015.1081349.

[92] Dario Páez, Bernard Rimé, Nekane Basabe, Anna Wlodarczyk, and Larraitz Zumeta. Psychosocial effects of perceived emotional synchrony in collective gatherings. *Journal of Personality and Social Psychology*, 108(5):711–729, May 2015. ISSN 1939-1315. doi: 10.1037/pspi0000014.

[93] Amit Goldenberg. What Makes Groups Emotional? *Perspectives on Psychological Science*, 19(2):489–502, March 2024. ISSN 1745-6916. doi: 10.1177/17456916231179154. URL `https://doi.org/10.1177/17456916231179154`. Publisher: SAGE Publications Inc.

[94] Sigal G. Barsade and Donald E. Gibson. Group emotion: A view from top and bottom. In *Composition*, Research on managing groups and teams, Vol. 1., pages 81–102. Elsevier Science/JAI Press, US, 1998. ISBN 978-0-7623-0460-8.

[95] Abhinav Dhall, Roland Goecke, Shreya Ghosh, Jyoti Joshi, Jesse Hoey, and Tom Gedeon. From individual to group-level emotion recognition: EmotiW 5.0. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, ICMI '17, pages 524–528, New York, NY, USA, November 2017. Association for Computing Machinery. ISBN 978-1-4503-5543-8. doi: 10.1145/3136755.3143004. URL `https://dl.acm.org/doi/10.1145/3136755.3143004`.

[96] Abhinav Dhall, Amanjot Kaur, Roland Goecke, and Tom Gedeon. EmotiW 2018: Audio-Video, Student Engagement and Group-Level Affect Prediction, August 2018. URL `http://arxiv.org/abs/1808.07773`. arXiv:1808.07773 [cs].

[97] Abhinav Dhall, Jyoti Joshi, Ibrahim Radwan, and Roland Goecke. Finding Happiest Moments in a Social Context. In Kyoung Mu Lee, Yasuyuki Matsushita, James M. Rehg, and Zhanyi Hu, editors, *Computer Vision – ACCV 2012*, pages 613–626, Berlin, Heidelberg, 2013. Springer. ISBN 978-3-642-37444-9. doi: 10.1007/978-3-642-37444-9_48.

[98] Patrícia Bota, Joana Brito, Ana Fred, Pablo Cesar, and Hugo Silva. A real-world dataset of group emotion experiences based on physiological data. *Scientific Data*, 11(1):116, January 2024. ISSN 2052-4463. doi: 10.1038/s41597-023-02905-6. URL `https://www.nature.com/articles/s41597-023-02905-6`. Publisher: Nature Publishing Group.

[99] Yanmin Qian, Chenda Li, Wangyou Zhang, and Shaoxiong Lin. Contextual understanding with contextual embeddings for multi-talker speech separation and recognition in a cocktail party. *npj Acoustics*, 1(1):1–12, April 2025. ISSN 3005-141X. doi: 10.1038/s44384-025-00004-x. URL `https://www.nature.com/articles/s44384-025-00004-x`. Publisher: Nature Publishing Group.

[100] E. Colin Cherry. Some experiments on the recognition of speech, with one and with two ears. *Journal of the Acoustical Society of America*, 25:975–979, 1953. ISSN 0001-4966. doi: 10.1121/1.1907229. Place: US Publisher: Acoustical Society of American.

[101] Shibani Hamsa, Ismail Shahin, Youssef Iraqi, Ernesto Damiani, and Naoufel Werghi. Speaker Identification from emotional and noisy speech data using learned voice segregation and Speech VGG, October 2022. URL `http://arxiv.org/abs/2210.12701`. arXiv:2210.12701 [eess].

[102] Jinyi Mi, Xiaohan Shi, Ding Ma, Jiajun He, Takuya Fujimura, and Tomoki Toda. Two-stage Framework for Robust Speech Emotion Recognition Using Target Speaker Extraction in Human Speech Noise Conditions, September 2024. URL `http://arxiv.org/abs/2409.19585`. arXiv:2409.19585.

[103] Jia Qi Yip, Dianwen Ng, Bin Ma, and Chng Eng Siong. Analysis of Speech Separation Performance Degradation on Emotional Speech Mixtures. *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 2002–2007, October 2023. doi: 10.1109/APSIPAASC58517.2023.10317465. URL `https://ieeexplore.ieee.org/document/10317465/`. Conference Name: 2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) ISBN: 9798350300673 Place: Taipei, Taiwan Publisher: IEEE.

[104] Jharna Agrawal, Manish Gupta, and Hitendra Garg. A review on speech separation in cocktail party environment: challenges and approaches. *Multimedia Tools and Applications*, 82(20):31035–31067, August 2023. ISSN 1573-7721. doi: 10.1007/s11042-023-14649-x. URL `https://doi.org/10.1007/s11042-023-14649-x`.

[105] Ján Švec, Kateřina Žmolíková, Martin Kocour, Marc Delcroix, Tsubasa Ochiai, Ladislav Mošner, and Jan Honza Černocký. Analysis of Impact of Emotions on Target Speech Extraction and Speech Separation. In *2022 International Workshop on Acoustic Signal Enhancement (IWAENC)*, pages 1–5, September 2022. doi: 10.1109/IWAENC53105.2022.9914718. URL `https://ieeexplore.ieee.org/document/9914718/`.

[106] Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and John R. Hershey. Single-Channel Multi-Speaker Separation using Deep Clustering, July 2016. URL `http://arxiv.org/abs/1607.02173`. arXiv:1607.02173 [cs].

[107] Javier De Lope and Manuel Graña. An ongoing review of speech emotion recognition. *Neurocomputing*, 528:1–11, April 2023. ISSN 09252312. doi: 10.1016/j.neucom.2023.01.002. URL `https://linkinghub.elsevier.com/retrieve/pii/S0925231223000103`.

[108] Yuanbo Gao, Baobin Li, Ning Wang, and Tingshao Zhu. Speech Emotion Recognition Using Local and Global Features. In Yi Zeng, Yong He, Jeanette Hellgren Kotaleski, Maryann Martone, Bo Xu, Hanchuan Peng, and Qingming Luo, editors, *Brain Informatics*, pages 3–13, Cham, 2017. Springer International Publishing. ISBN 978-3-319-70772-3. doi: 10.1007/978-3-319-70772-3_1.

[109] Sinith M S, Aswathi Elampulakkadu, T. Deepa, C. Shameema, and Shiny Rajan. *Emotion recognition from audio signals using Support Vector Machine.* December 2015. doi: 10.1109/RAICS.2015.7488403. Pages: 144.

[110] Felix Burkhardt, A. Paeschke, M. Rolfes, Walter F. Sendlmeier, and Benjamin Weiss. A database of German emotional speech. In *Interspeech 2005*, pages 1517–1520. ISCA, September 2005. doi: 10.21437/Interspeech.2005-446. URL https://www.isca-archive.org/interspeech_2005/burkhardt05b_interspeech.html.

[111] Lijiang Chen, Xia Mao, Yuli Xue, and Lee Lung Cheng. Speech emotion recognition: Features and classification models. *Digital Signal Processing*, 22(6):1154–1160, December 2012. ISSN 1051-2004. doi: 10.1016/j.dsp.2012.05.007. URL https://www.sciencedirect.com/science/article/pii/S1051200412001133.

[112] Fekade Getahun and Mikiyas Kebede. Emotion Identification from Spontaneous Communication. In *2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pages 151–158, November 2016. doi: 10.1109/SITIS.2016.32. URL https://ieeexplore.ieee.org/document/7907459/.

[113] Promod Yenigalla, Abhay Kumar, Suraj Tripathi, Chirag Singh, Sibsambhu Kar, and Jithendra Vepa. Speech Emotion Recognition Using Spectrogram & Phoneme Embedding. In *Interspeech 2018*, pages 3688–3692. ISCA, September 2018. doi: 10.21437/Interspeech.2018-1811. URL https://www.isca-archive.org/interspeech_2018/yenigalla18_interspeech.html.

[114] Leimin Tian, Johanna D. Moore, and Catherine Lai. Recognizing emotions in dialogues with acoustic and lexical features. In *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 737–742, September 2015. doi: 10.1109/ACII.2015.7344651. URL https://ieeexplore.ieee.org/document/7344651/. ISSN: 2156-8111.

[115] Guanming Lu and Wenjing Zhang. Happiness Intensity Estimation for a Group of People in Images using Convolutional Neural Networks. In *2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE)*, pages 1707–1710, October 2019. doi: 10.1109/EITCE47263.2019.9094832. URL https://ieeexplore.ieee.org/document/9094832.

[116] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition, April 2015. URL http://arxiv.org/abs/1409.1556. arXiv:1409.1556 [cs].

[117] Yanan Wang, Jianming Wu, Panikos Heracleous, Shinya Wada, Rui Kimura, and Satoshi Kurihara. Implicit Knowledge Injectable Cross Attention Audiovisual Model for Group Emotion Recognition. In *Proceedings of the 2020 International*

Conference on Multimodal Interaction, ICMI '20, pages 827–834, New York, NY, USA, October 2020. Association for Computing Machinery. ISBN 978-1-4503-7581-8. doi: 10.1145/3382507.3417960. URL `https://dl.acm.org/doi/10.1145/3382507.3417960`.

[118] Department of Computer Science, Government M.A.M College, Cluster University of Jammu, Jammu, India., Dr. Archana Sharma*, Dr. Vibhakar Mansotra, and Department of Computer Science and IT, University of Jammu, Jammu, India. Multimodal Decision-level Group Sentiment Prediction of Students in Classrooms. *International Journal of Innovative Technology and Exploring Engineering*, 8(12):4902–4909, October 2019. ISSN 22783075. doi: 10.35940/ijitee.L3549.1081219. URL `https://www.ijitee.org/portfolio-item/L35491081219/`.

[119] Björn Schuller, Stefan Steidl, Anton Batliner, Felix Burkhardt, Laurence Devillers, Christian Müller, and Shrikanth S. Narayanan. The INTERSPEECH 2010 paralinguistic challenge. In *Interspeech 2010*, pages 2794–2797. ISCA, September 2010. doi: 10.21437/Interspeech.2010-739. URL `https://www.isca-archive.org/interspeech_2010/schuller10b_interspeech.html`.

[120] Rodolfo Migon Favaretto, Paulo Knob, Soraia Raupp Musse, Felipe Vilanova, and Ângelo Brandelli Costa. Detecting Personality and Emotion Traits in Crowds from Video Sequences. *Machine Vision and Applications*, 30(5):999–1012, July 2019. ISSN 0932-8092, 1432-1769. doi: 10.1007/s00138-018-0979-y. URL `http://arxiv.org/abs/2104.12927`. arXiv:2104.12927 [cs].

[121] Yu Wang, Shunping Zhou, Yuanyuan Liu, Kunpeng Wang, Fang Fang, and Haoyue Qian. ConGNN: Context-consistent cross-graph neural network for group emotion recognition in the wild. *Information Sciences*, 610:707–724, September 2022. ISSN 0020-0255. doi: 10.1016/j.ins.2022.08.003. URL `https://www.sciencedirect.com/science/article/pii/S0020025522008830`.

[122] Siyuan Shen, Feng Liu, and Aimin Zhou. Mingling or Misalignment? Temporal Shift for Speech Emotion Recognition with Pre-trained Representations, March 2023. URL `http://arxiv.org/abs/2302.13277`. arXiv:2302.13277 [cs].

[123] Rong Jin, Mijit Ablimit, and Askar Hamdulla. Speech Separation and Emotion Recognition for Multi-speaker Scenarios. In *2022 3rd International Conference on Pattern Recognition and Machine Learning (PRML)*, pages 280–284, July 2022. doi: 10.1109/PRML56267.2022.9882231. URL `https://ieeexplore.ieee.org/document/9882231`.

[124] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations, October 2020. URL `http://arxiv.org/abs/2006.11477`. arXiv:2006.11477 [cs].

[125] FFmpeg Developers. FFmpeg tool, 2016. URL `http://ffmpeg.org/`.

[126] Quan Wang. Awesome diarization, 2025. URL `https://wq2012.github.io/awesome-diarization/`.

[127] Alexis Plaquet and Hervé Bredin. Powerset multi-class cross entropy loss for neural speaker diarization. In *INTERSPEECH 2023*, pages 3222–3226, August 2023. doi: 10.21437/Interspeech.2023-205. URL `http://arxiv.org/abs/2310.13025`. arXiv:2310.13025 [cs].

[128] Lian Remme and Kevin Tang. Playing with Voices: Tabletop Role-Playing Game Recordings as a Diarization Challenge, February 2025. URL `http://arxiv.org/abs/2502.12714`. arXiv:2502.12714 [cs].

[129] Chau Luu. simple_diarizer: Simplified diarization pipeline using some pretrained models - audio file to diarized segments in a few lines of code, 2025. URL `https://github.com/cvqluu/simple_diarizer`.

[130] Nishchal Bhandari, Danny Chen, Miguel Ángel del Río Fernández, Natalie Delworth, Jennifer Drexler Fox, Migüel Jetté, Quinten McNamara, Corey Miller, Ondřej Novotný, Ján Profant, Nan Qin, Martin Ratajczak, and Jean-Philippe Robichaud. Reverb: Open-Source ASR and Diarization from Rev, February 2025. URL `http://arxiv.org/abs/2410.03930`. arXiv:2410.03930 [cs].

[131] Shengkui Zhao, Yukun Ma, Chongjia Ni, Chong Zhang, Hao Wang, Trung Hieu Nguyen, Kun Zhou, Jiaqi Yip, Dianwen Ng, and Bin Ma. MossFormer2: Combining Transformer and RNN-Free Recurrent Network for Enhanced Time-Domain Monaural Speech Separation, November 2024. URL `http://arxiv.org/abs/2312.11825`. arXiv:2312.11825 [cs] version: 2.

[132] John R. Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation, August 2015. URL `http://arxiv.org/abs/1508.04306`. arXiv:1508.04306 [cs] version: 1.

[133] Joris Cosentino, Manuel Pariente, Samuele Cornell, Antoine Deleforge, and Emmanuel Vincent. LibriMix: An Open-Source Dataset for Generalizable Speech Separation, May 2020. URL `http://arxiv.org/abs/2005.11262`. arXiv:2005.11262 [eess].

[134] Gordon Wichern, Joe Antognini, Michael Flynn, Licheng Richard Zhu, Emmett McQuinn, Dwight Crow, Ethan Manilow, and Jonathan Le Roux. WHAM!: Extending Speech Separation to Noisy Environments, July 2019. URL `http://arxiv.org/abs/1907.01160`. arXiv:1907.01160 [cs].

[135] Matthew Maciejewski, Gordon Wichern, Emmett McQuinn, and Jonathan Le Roux. WHAMR!: Noisy and Reverberant Single-Channel Speech Separation, February 2020. URL `http://arxiv.org/abs/1910.10279`. arXiv:1910.10279 [cs].

[136] Florian Eyben, Martin Wöllmer, Alex Graves, Björn Schuller, Ellen Douglas-Cowie, and Roddy Cowie. On-line emotion recognition in a 3-D activation-valence-time continuum using acoustic and linguistic cues. *Journal on Multimodal User Interfaces*, 3(1):7–19, March 2010. ISSN 1783-8738. doi: 10.1007/s12193-009-0032-6. URL `https://doi.org/10.1007/s12193-009-0032-6`.

[137] Mohammed A. Almulla. A multimodal emotion recognition system using deep convolution neural networks. *Journal of Engineering Research*, page S2307187724000890, March 2024. ISSN 23071877. doi: 10.1016/j.jer.2024.03.021. URL `https://linkinghub.elsevier.com/retrieve/pii/S2307187724000890`.

[138] S. Lalitha, D. Geyasruti, R. Narayanan, and Shravani M. Emotion Detection Using MFCC and Cepstrum Features. *Procedia Computer Science*, 70:29–35, 2015. ISSN 18770509. doi: 10.1016/j.procs.2015.10.020. URL `https://linkinghub.elsevier.com/retrieve/pii/S1877050915031841`.

Declaration of Authorship

I hereby declare that this Master's thesis has been written by myself independently without any help from others, and only the defined sources and study aids were used. Sections that reflect the thoughts or works of others are made known through the definition of sources.

Hamburg, 15.05.2025