BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER SYSTEMS
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

EMOTION RECOGNITION FROM ANALYSIS OF A PERSON'S SPEECH USING DEEP LEARNING

ROZPOZNÁNÍ EMOCÍ Z ANALÝZY ŘEČI ČLOVĚKA POMOCÍ HLUBOKÉHO UČENÍ

MASTER'S THESIS
DIPLOMOVÁ PRÁCE

AUTHOR **Bc. ŠIMON GALBA**
AUTOR PRÁCE

SUPERVISOR **doc. AAMIR SAEED MALIK, Ph.D.**
VEDOUCÍ PRÁCE

BRNO 2024

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Master's Thesis Assignment

153400

Institut:	Department of Computer Systems (DCSY)
Student:	**Galba Šimon, Bc.**
Programme:	Information Technology and Artificial Intelligence
Specialization:	Machine Learning
Title:	**Emotion Recognition from Analysis of a Person's Speech using Deep Learning**
Category:	Biocomputing
Academic year:	2023/24

Assignment:

1. Study and learn about the various emotions and moods and how they affect the various features of the speech of a person.
2. Get acquainted with audio and speech processing methods as well as machine learning techniques and their application to the recognition of emotions and moods.
3. Find out the challenges for emotion and mood interpretation from a person's speech as well as the limitations of the existing methods.
4. Design an algorithm for interpretation of emotion and mood from the audio of a person's speech using deep learning.
5. Implement the designed algorithm.
6. Create a set of benchmark tasks to evaluate the quality of emotion and mood recognition from a person's speech (audio) as well as the corresponding computational performance and memory usage.
7. Conduct critical analysis and discuss the achieved results and their contribution.

Literature:

- According to supervisor's advice.

Requirements for the semestral defence:

- Items 1 to 4 of the assignment.

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor:	**Malik Aamir Saeed, doc., Ph.D.**
Head of Department:	Sekanina Lukáš, prof. Ing., Ph.D.
Beginning of work:	1.11.2023
Submission deadline:	17.5.2024
Approval date:	30.10.2023

Abstract

This thesis deals with the analysis and implementation of a neural network for the purpose of recognizing emotions from human speech using deep learning. The thesis also focuses on tuning this network to achieve greater sensitivity to a specific emotion and explores the time and indirectly the financial requirements of this tuning. The inspiration for creating this work is the increasing integration of artificial intelligence in the fields of biology, healthcare, as well as psychology, and one of the goals is also to study the complexity of creating specific models of neural networks for purposes in these sciences, which should contribute to better accessibility of artificial intelligence models. The work is based on the implementation of the "AST: Audio Spectrogram Transformer" model, which is publicly available under the BSD 3-Clause License and utilizes methods that have been used so far for classification and recognition of images by converting an audio track into a spectrogram. The resulting values of weighted accuracy are as follows: 93.5% for the EMODB dataset, 92.8% for EMOVO, and 92.9% for the RAVDESS dataset.

Abstrakt

Táto práca sa zaoberá analýzou a implementáciou neurónovej siete za účelom rozpoznávania emócií z reči človeka pomocou hlbokého učenia. Práca sa taktiež zaoberá ladením tejto siete za účelom dosiahnutia väčšej citlivosti voči konkrétnej emócii a skúma časové a nepriamo aj finančné nároky tohto ladenia. Inšpiráciou na vytvorenie tejto práce je stúpajúca integrácia umelej inteligencie v oblasti biológie, zdravotníctva ako aj psychológie a jedným z cieľov je aj skúmanie náročnosti vytvárať konkrétne modely neurónových sietí na účely v týchto vedách, čo by malo prispieť k lepšej dostupnosti modelov umenelej inteligencie. Práca stavia na základe implementácie modelu "AST: Audio Spectrogram Transformer" ktorá je verejne dostupná pod licenciou BSD 3-Clause License a využíva metódy ktoré boli doposiaľ využívané na klasifikáciu a rozpoznávanie obrazov vďaka premene zvukovej stopy na spektrogram. Výsledné hodnoty váženej presnosti sú následovné: 93.5% pre EMODB dataset, 92.8% pre EMOVO a 92,9% pre dataset RAVDESS.

Keywords

deep learning, Audio Spectrogram Transformer, speech emotion recognition, speech signal processing, emotion classification

Klíčová slova

hluboké učení, Audio Spectrogram Transformer, rozpoznávání emocí z řeči, zpracování řečového signálu, klasifikace emocí

Reference

GALBA, Šimon. *Emotion Recognition from Analysis of a Person's Speech using Deep Learning.* Brno, 2024. Master's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor doc. Aamir Saeed Malik, Ph.D.

Rozšířený abstrakt

Táto diplomová práca sa zameriava na rozpoznávanie emócií z ľudskej reči prostredníctvom hlbokého učenia s využitím neurónovej siete Audio Spectrogram Transformer (AST). Hlavným cieľom práce je implementácia a optimalizácia AST modelu na analýzu emocionálneho obsahu v reči, kde vstupnými dátami sú spektrogramy. Spektrogramy, získané transformáciou zvukových signálov, poskytujú vizuálnu reprezentáciu frekvenčných komponentov, ktoré AST model efektívne spracováva na rozpoznanie špecifických emócií.

Práca detailne popisuje proces prípravy a predspracovania dát, vrátane konverzie audio signálov na spektrogramy, čo umožňuje modelu AST naučiť sa rozpoznávať vzory spojené s rôznymi emocionálnymi stavmi. Tento prístup vyžaduje nielen technické znalosti o spracovaní zvuku, ale tiež pochopenie, ako rôzne emocionálne stavy ovplyvňujú akustické vlastnosti reči.

Ďalším kľúčovým aspektom práce je využitie cross-corpus prístupu pre tréning modelu, ktorý zahŕňa dátové sady z rôznych lingvistických a kultúrnych prostredí, ako sú EMODB, EMOVO a RAVDESS. Tento prístup umožňuje modelu získať schopnosť generalizovať emocionálne rozpoznávanie naprieč rôznymi korpusmi, čím sa zvyšuje jeho robustnosť a adaptabilita.

V neskoršej fáze práca preskúma možnosti jemného ladenia, ktoré je známe pod zaužívaným anglickým názvom "fine-tuning", modelu na špecifickú dátovú sadu s nižšími výpočtovými nárokmi. Fine-tuning sa zameriava na optimalizáciu výkonu modelu pri zachovaní nízkej výpočtovej náročnosti, čo je kľúčové pre aplikácie v reálnom čase. Tento proces zahŕňa úpravu parametrov vrstiev modelu AST, ktoré sú zodpovedné za konečnú klasifikáciu emocionálnych stavov, s cieľom dosiahnuť vyššiu presnosť pri rozpoznávaní cieľových emócií.

Výsledky práce ukazujú, že upravený model AST dosahuje vysokú presnosť rozpoznávania emócií a demonstruje jeho praktickú aplikovateľnosť vo viacerých oblastiach, vrátane klinickej psychológie, bezpečnostných systémov a interaktívnych systémov založených na rozpoznávaní reči. Rozšírená analýza a evaluácia modelu na rôznych dátových sadách potvrdzujú jeho efektivitu a poukazujú na potenciálne vylepšenia pre budúce výskumy.

Táto práca prispieva k hlbšiemu porozumeniu možností hlbokého učenia v oblasti rozpoznávania emócií z reči a predstavuje dôležitý krok k lepšej integrácii umelej inteligencie do aplikácií súvisiacich s interakciou človeka a počítača.

Emotion Recognition from Analysis of a Person's Speech using Deep Learning

Declaration

I hereby declare that this Master's thesis was prepared as an original work by the author under the supervision of doc. Aamir Saeed Malik, Ph.D. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

.....
Šimon Galba
May 16, 2024

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Doc. Aamir Saeed Malik, Ph.D., for his invaluable guidance, patience, and expert advice throughout the duration of this research. His insights and expertise have been fundamental to the completion of this thesis, and his encouragement was crucial in overcoming the challenges encountered along the way.

Special thanks goes to doc. Ing. Jiří Jaroš Ph.D. for a fast and simple access to the supercomputer used to calculate the models in this thesis.

I am also immensely thankful to my family, whose unwavering support and understanding have been my pillars of strength throughout my studies. Their endless encouragement and belief in my abilities have been a constant source of motivation and have significantly lightened the burden during the most demanding periods of my academic journey.

A special word of appreciation goes to my girlfriend, who has created a harmonious and supportive environment that was essential for my focus and productivity. Her patience, love, and understanding provided the calm amidst the storm of rigors of research, making it possible for me to pursue my academic goals without reservation.

Contents

1 Introduction **3**

2 Emotion Models in Computer Software **4**
2.1 Historical Overview of Emotion Theories 4
2.2 Emotion Models in Computer Software 5
2.3 Signal Processing Theoretics: Audio Signals 8

3 Emotion recognition from speech **12**
3.1 SER Datasets 12
3.2 Traditional Machine Learning Approaches for SER 13
3.3 Deep Learning Approaches for SER 17
3.4 Summary 27

4 Proposed Methodology **30**
4.1 Pre-trained model 31
4.2 Model parameters 32
4.3 Evaluation Metrics 34
4.4 Strategies to Improve Evaluation Metrics for an Audio Spectrogram Transformer (AST) Model 36
4.5 Fine-Tuning for Speech Emotion Recognition 37

5 Implementation **39**
5.1 Top Level Overview 39
5.2 Model training 40
5.3 Fine Tuning 47
5.4 Computational Complexity 48

6 Results and Discussion **52**
6.1 Optimizing Recall for Disease Recognition Tasks 52
6.2 Cross-Corpus Training and Fine-Tuning on Datasets 54
6.3 Known issues and complications 59
6.4 Future work 59

7 Conclusion **60**

Bibliography **61**

A SD card content **65**

List of Figures

2.1 The Circumplex Model of Emotion [43] 6
2.2 Plutchik's wheel of emotions [29] 7
2.3 Lövheim Cube of Emotion [24] 8
2.4 Different waveforms for different emotion affectation for the same sentence [44] 9
2.5 Spectral Envelope [32] 9
2.6 Hamming Window 10
2.7 Mel Filter Banks 11
2.8 Jitter and Shimmer 11

3.1 The architecture of TIM-Net 19
3.2 Architecture proposed by the authors of AST [13] 21

4.1 Architecture of a Neural Network based on the AST Model 32

5.1 The overall flow of the program 39
5.2 Orignal waveform 43
5.3 Random noise ugmentation of the signal 43
5.4 Random speed change augmentation 44
5.5 Random Impulse Response augmentation (RIR) 44
5.6 Time masking spectogram augmentation 45
5.7 Frequency masking spectogram augmentation 45
5.8 WanDB training run charts 47
5.9 Comparison of one epoch time training on CPU vs GPU(CUDA activated) 51

6.1 Confusion Matrix of the Base Model 53
6.2 Confusion Matrix of the Fine-Tuned Model, with emphasis on Anger detection 53
6.3 Confusion matrix of the cross-corpus model tested on the RAVDESS dataset 55
6.4 Confusion matrix of the model after fine-tuning on the RAVDESS dataset . 56
6.5 Confusion matrix of the model tested on the EMODB dataset 57
6.6 Confusion matrix of the fine-tuned model tested on the EMODB dataset . . 57
6.7 Initial confusion matrix of the model tested on the EMOVO dataset. 58
6.8 Confusion matrix of the model after additional tuning on the EMOVO dataset. 59

Chapter 1

Introduction

Emotion recognition and simulation have become pivotal in the interface between humans and computers, marking a significant evolution in both cognitive science and artificial intelligence. Humans experience and express emotions with a complex interplay of physiological, cognitive, and social factors [38]. These emotional expressions are often subtle and nuanced, influenced by personal experiences and cultural contexts. In contrast, computers must rely on explicit models and algorithms to „understand“ or simulate emotions. They do this by processing observable data such as facial expressions, voice modulations, and body language, which are then interpreted through predefined frameworks like the Circumplex Model of Emotion 2.2.1 or Plutchik's Emotion Wheel 2.2.2.

In this thesis, we will focus on classifying human emotions from their speech with the help of deep learning neural networks.

Chapter 2

Emotion Models in Computer Software

Unlike humans, who can intuitively grasp and react to emotional subtleties, computers require extensive data and sophisticated algorithms to approximate this understanding. This disparity arises because human emotional processing involves not only basic sensory input but also a deep, often unconscious synthesis of past experiences, cultural norms, and personal expectations. Computers, however, operate within the confines of their programming and algorithms, which can only mimic this process to a limited extent. For instance, while a human might detect sarcasm or a subtle shift in mood from a slight change in tone, a computer needs clear, distinct patterns that fit within its programmed understanding.

This chapter explores how emotion models are conceptualised and implemented in software to bridge this gap between human emotional complexity and computer processing capabilities. By integrating these models into systems, developers aim to enhance the machine's ability to interpret human emotions accurately and interact in a more human-like, empathetic manner. Such advancements not only improve the user experience but also open new avenues in how we understand and interact with technology, making interactions more natural and intuitive.

2.1 Historical Overview of Emotion Theories

The study of emotions spans multiple disciplines including psychology, neuroscience, and philosophy. The understanding of emotions has evolved significantly from ancient to modern times, impacting how emotions are modelled in computational systems today.

Ancient and Philosophical Perspectives

The philosophical inquiry into emotions dates back to the works of Aristotle and Plato, who pondered the role of emotions in human rationality and ethics. Aristotle's "Rhetoric" discusses emotions as persuasive tools, while Plato considered them part of the psyche that could disturb rational thinking [33, 28].

Evolutionary Theories

Charles Darwin's work in the 19th century marked a pivotal turn toward understanding emotions from an evolutionary perspective. In his seminal book „The Expression of the

Emotions in Man and Animals“ [9], Darwin proposed that emotions served adaptive evolutionary functions, which could be understood through patterns of expression that were consistent across cultures [9]. This work laid the foundation for later scientific studies into the biological bases of emotion.

Early Psychological Theories

In the late 19th and early 20th centuries, William James and Carl Lange independently proposed what is now known as the James-Lange Theory of Emotions. This theory suggests that physiological arousal precedes the experience of emotion in which people feel sad because they cry, and not the other way around [17]. Although later debated and refined, this theory was crucial in shifting the focus to the physiological underpinnings of emotional experiences.

The Development of Modern Emotion Psychology

Throughout the 20th century, further theories emerged that expanded upon these foundations. The Cannon-Bard theory challenged the James-Lange theory by proposing that emotions and physiological responses occur simultaneously rather than sequentially [8]. Later, Schachter and Singer's Two-Factor Theory introduced the idea that both physiological arousal and cognitive interpretation are necessary for the experience of emotion, adding complexity to understanding how emotions are processed [35].

All of these theories have contributed to the rich tapestry from which modern emotion models in computer software have been developed. They provide the necessary historical context to appreciate the complexity and depth of human emotions that we attempt to model today.

2.2 Emotion Models in Computer Software

2.2.1 The Circumplex Model of Emotion

Developed by James A. Russell in the early 1980s, the Circumplex Model of Emotion 2.1 is a seminal framework in affective psychology that classifies emotions in a two-dimensional space of arousal and valence [34]. Arousal indicates the level of energy associated with an emotion, whereas valence reflects the degree of pleasantness. This model has been particularly influential in the development of emotion recognition software, which uses these two dimensions to analyze facial expressions, voice tone, and physiological responses to categorize the emotional state of users [3]. The simplicity of this model makes it highly effective for real-time emotion assessment in interactive applications such as virtual assistants and customer service chatbots.

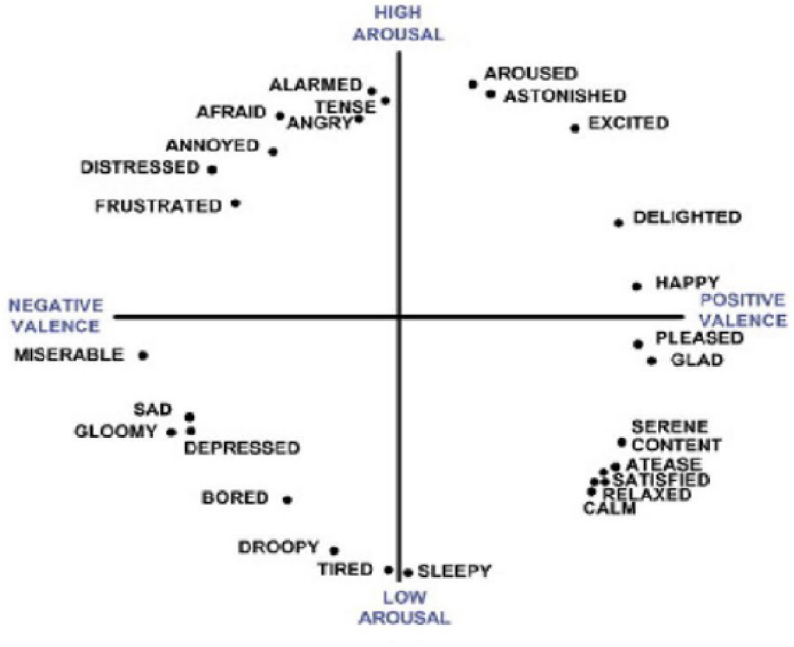


Figure 2.1: The Circumplex Model of Emotion [43]

The combination of these dimensions allows for the placement of specific emotions within the circle. For example:

- High Arousal, Positive Valence: Excitement, ecstasy
- High Arousal, Negative Valence: Fear, anger
- Low Arousal, Positive Valence: Contentment, satisfaction
- Low Arousal, Negative Valence: Boredom, sadness

2.2.2 Plutchik's Emotion Wheel

Robert Plutchik proposed his Emotion Wheel 2.2 in 1980 as a way to illustrate the relationships among different emotions, conceptualizing them as eight primary bipolar emotions: joy versus sadness, anger versus fear, trust versus disgust, and surprise versus anticipation [29]. This model extends to include various degrees of intensity of each emotion, and combinations of the primary emotions can form complex feelings. In computer software, Plutchik's model is utilised to enhance emotional analysis algorithms. It enables more complex emotion recognition capabilities that are critical in areas such as behavioural prediction, personalised content delivery, and therapeutic settings where understanding nuanced emotional responses is key [25].

Here are the eight primary emotions in Plutchik's model, along with their opposites:

Joy <-> Sadness

Trust <-> Disgust

Fear <-> Anger

Surprise <-> Anticipation

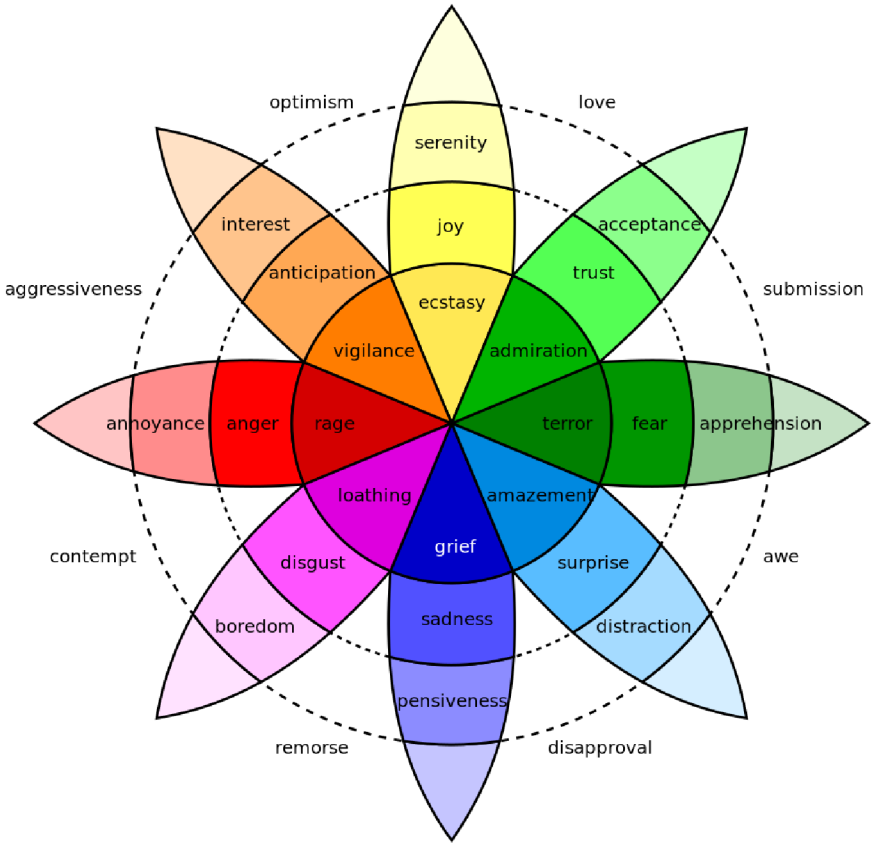


Figure 2.2: Plutchik's wheel of emotions [29]

2.2.3 Lövheim Cube of Emotion

The Lövheim Cube of Emotion 2.3 presents a three-dimensional model based on the levels of the neurotransmitters serotonin, dopamine and noradrenaline, positing that different combinations of these levels of neurotransmitters lead to different emotions [24]. This model is especially relevant in the development of affective computing systems that need to simulate human emotions with high accuracy. For example, in therapeutic software used in mental health treatment, the Lövheim Cube can guide the simulation of patient emotions under various scenarios, thus aiding in more effective treatment planning and support.

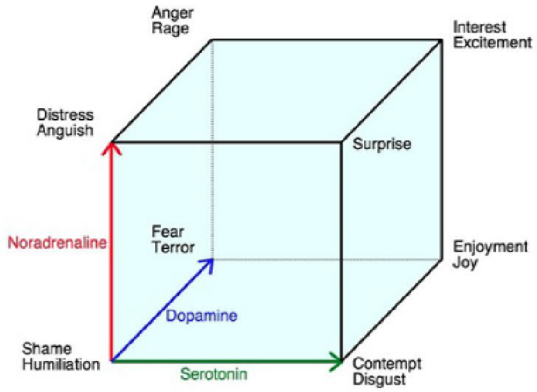


Figure 2.3: Lövheim Cube of Emotion [24]

2.2.4 Conclusion

The integration of emotion models into computer software has transformed the landscape of human-computer interaction. By employing sophisticated models such as those discussed in this chapter, developers can create software that not only understands human emotions, but also responds to them in a manner that mimics human empathy and understanding, thereby enhancing user experience and broadening the applicability of technology in emotionally sensitive applications.

2.3 Signal Processing Theoretics: Audio Signals

This section delves into the fundamental aspects of audio signal processing that are crucial to understanding how emotional cues can be extracted from speech and vocal expressions. Each subsection focuses on a key concept or technique.

„A signal is a representation of a quantity that varies over time or space and is used to convey information. In computer science, signals are often processed digitally and can be represented as a sequence of samples. Examples include audio signals, images, and network traffic. The primary focus of signal processing is to analyze, modify, and interpret these signals for various applications such as communication, audio processing, and image analysis.“ [27] In the Figure 2.4, you can see samples of how audio signals can differentiate based on the emotion expressed in the same sample of speech.

Figure 2.4: Different waveforms for different emotion affectation for the same sentence [44]

2.3.1 Audio Signal Power Spectrum and Its Spectral Envelope

The power spectrum of an audio signal represents the distribution of power in the frequency components that make up that signal. It provides insights into the harmonic content and the energy of the signal at various frequencies. The spectral envelope, on the other hand, is a smooth curve that represents the peaks of the power spectrum, effectively capturing the resonant frequencies of the vocal tract that are critical for characterizing speech sounds [32]. An example of such envelope is shown in Figure 2.5.

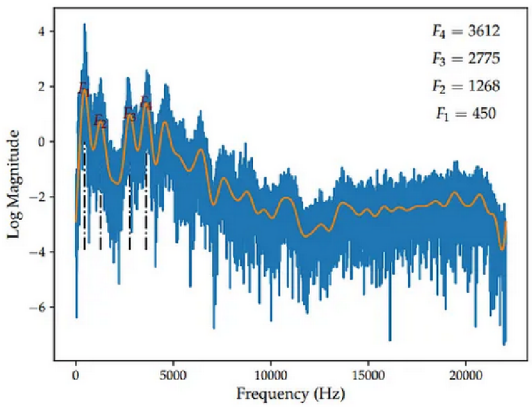


Figure 2.5: Spectral Envelope [32]

2.3.2 Waveform Examples

A waveform is a visual representation of the shape of a sound signal in the time domain. Analyzing waveforms allows for the observation of characteristics such as amplitude and frequency over time, providing a fundamental understanding of sound properties. This analysis is essential for distinguishing between different types of sound expressions and is particularly useful in speech analysis [12].

2.3.3 Windowing Examples

Windowing is a technique used in signal processing where the signal is multiplied by a window function. This method reduces artifacts in the Fourier transform of the signal, particularly discontinuities at the edges of a sampled time window. Common windows include Hamming, Hanning, and Blackman windows shown in Figure 2.6, each with specific properties that make them suitable for different types of signal analysis tasks [15].

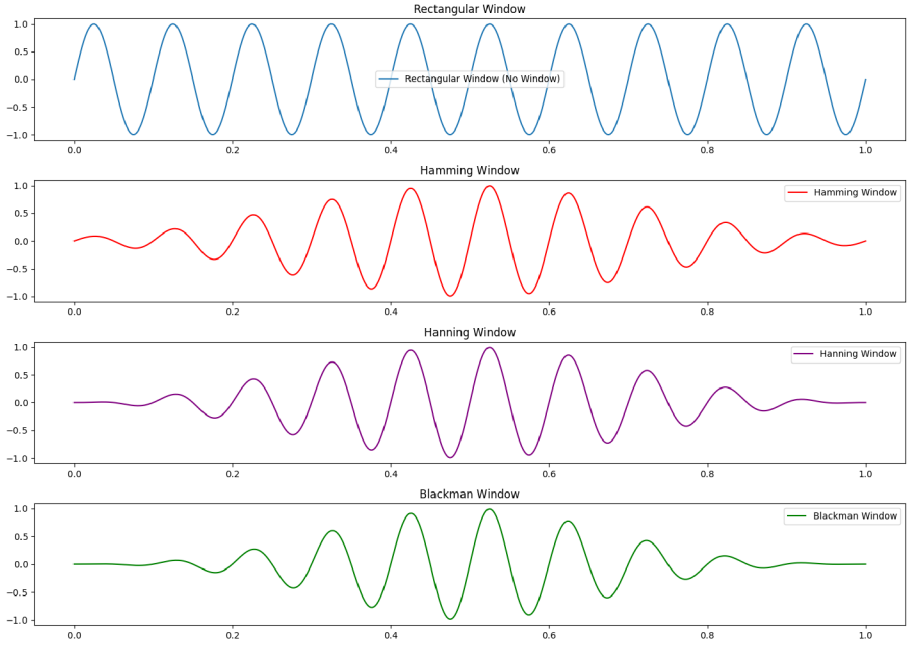


Figure 2.6: Hamming Window

2.3.4 Mel Filter Bank Example

The Mel Filter Bank is used to mimic the human ear's response to different frequencies, capturing the essential characteristics of sound in terms of human perception. It consists of a set of triangular filters, as seen in Figure 2.7, each tuned to a specific frequency band centered on the Mel scale. This technique is extensively used in voice recognition and speech processing applications to extract features that are robust and relevant for identifying emotional content in speech [10].

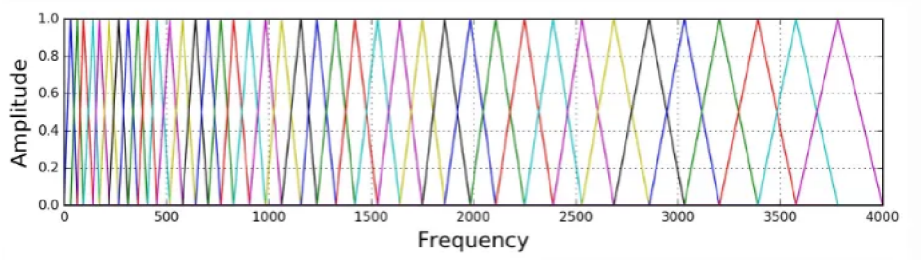


Figure 2.7: Mel Filter Banks

2.3.5 Jitter and Shimmer Example

Jitter and shimmer, as seen in Figure 2.8, are measures used to assess the stability and quality of the human voice. Jitter refers to the frequency variation from one cycle to the next, while shimmer refers to the amplitude variation. These parameters are important indicators of voice disorders and are also useful for emotional state analysis, as emotional states can influence voice stability and quality [18].

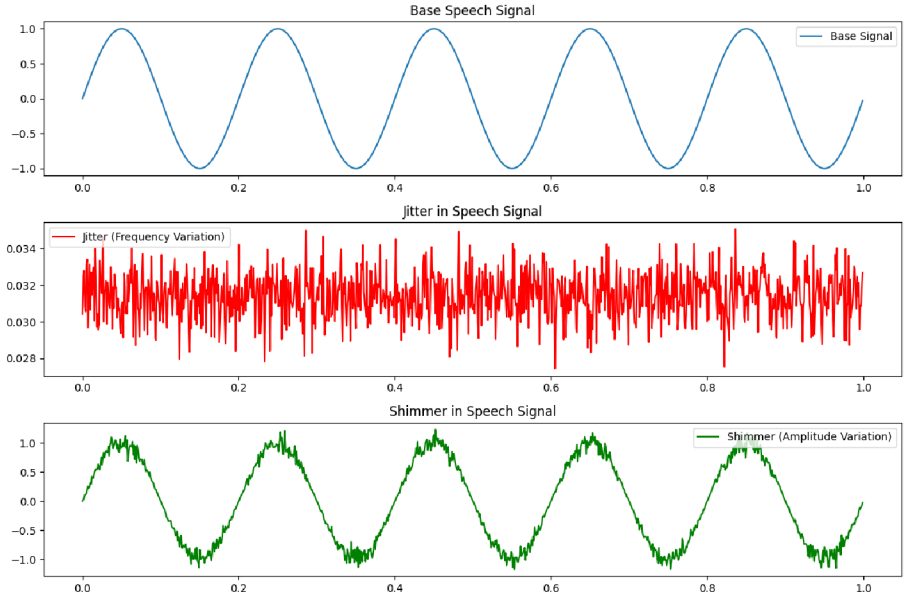


Figure 2.8: Jitter and Shimmer

This thesis aims to propose the Speech Emotion Recognition (SER) system that recognizes emotions from a speech in a way that approaches the abilities of humans, and to do so there is a need to study how to categorize emotions like them as the fundamental brick for the following work. Two types of emotional models are interesting from the SER point of view: **discrete (categorical)** model and **dimensional (continuous)** model [2].

Chapter 3

Emotion recognition from speech

3.1 SER Datasets

In recent advancements, the field of emotion recognition has become a focal point in machine learning research, prompting the development of numerous datasets. These datasets vary greatly in size, quality, and type, each tailored to meet specific research needs in the field of speech emotion recognition (SER).

3.1.1 Types of SER Datasets

Datasets in SER are classified based on the nature of the emotion elicitation and recording:

- **Acted:** Actors are instructed to express particular emotions, which are then captured in recordings. This type typically allows for controlled study of specific emotional states but may lack naturalism.
- **Naturalistic:** These datasets consist of recordings from real-life interactions or monologues where emotions occur naturally, providing a more genuine insight into human emotional expression.
- **Bilingual:** Unique datasets where the same phrases are recorded by the same speaker in multiple languages, enriching the dataset with linguistic diversity that is beneficial for multi-lingual emotion recognition systems.
- **Cross-corpus:** Perhaps the most valuable for robust algorithm training, these datasets combine various types of data collections, enhancing the model's ability to generalize across different languages, modalities, and emotional expressions.

3.1.2 Highlighted SER Datasets

This thesis utilizes five key datasets, whose overview can be seen in table 3.1, chosen for their relevance in benchmarking the performance of contemporary models as well as their availability through existing university resources. Detailed exploration of these datasets facilitates a deeper understanding of model accuracies and advancements in SER technology. The datasets include:

- **MSP-Podcast:** This extensive dataset includes over 104,267 speaking turns, accumulating to about 166 hours and 9 minutes of naturalistic emotional speech collected

from various podcasts. It is particularly valuable for studying spontaneous emotional expressions in speech [23].

- **EMO-DB:** A well-established dataset, the Berlin Database of Emotional Speech (EMO-DB), supports the analysis and development of algorithms for emotion recognition. It includes a variety of emotional states expressed in German through scripted statements, making it a staple in many SER studies [5].
- **IEMOCAP:** Focused on dyadic interactions, the Interactive Emotional Dyadic Motion Capture database offers a rich source of audio-visual data capturing naturalistic emotional expressions in controlled scenarios. This dataset is widely used for training and testing SER systems [6].
- **RAVDESS:** The Ryerson Audio-Visual Database of Emotional Speech and Song contains meticulously recorded audio and visual data of professional actors expressing a range of emotions through speech and song. Its detailed annotation system includes information on modality, vocal channel, emotion, emotional intensity, statement, repetition, and actor identity, providing a structured framework for comprehensive emotion analysis [22].

Table 3.1: Brief description of databases for SER.

Name	*Type*	*Language*	*Emotions*	*Citations count*
Emo-DB	Acted	German	Neutral, anger, sadness, fear, boredom, happiness, disgust.	1237
IEMOCAP	Elicited	English	Anger, happiness, sadness, frustration, neutral.	1421
RAVDESS	Acted	English	Surprise, anger, fear, disgust, sadness, neutral, calm, happiness.	558
MSP-Podcast	Natural	English	Surprise, anger, fear, disgust, sadness, neutral, calm, happiness, concerned, depressed, excited.	853

3.2 Traditional Machine Learning Approaches for SER

3.2.1 Features

Designing an effective speech emotion recognition (SER) system involves the meticulous identification and extraction of key emotion-related speech features. Human capabilities in interpreting both linguistic and paralinguistic cues from speech highlight the complexity of this task. The selection of appropriate speech features is crucial for enhancing the performance of SER classifiers.

Speech Feature Categories

Numerous types of features have been studied extensively in SER research:

- **Local and Global Features:** Local features represent short-term properties of speech, whereas global features capture long-term aspects.
- **Continuous Speech Features:** These features are derived from flowing speech, providing a dynamic perspective of emotional expression.
- **Qualitative Features:** Subjective qualities such as tone and stress fall under this category.
- **Spectral Features:** These include fundamental frequency, formants, and other frequency-related characteristics that are vital for distinguishing emotional states in speech.
- **Teager Energy Operator (TEO) Features:** TEO-based features help in analyzing the energy operators of speech signals, which are effective in identifying speech modulations.
- **Excitation Source Features:** These features, including pitch and voice quality, are derived from the source of vocal excitation.
- **Vocal Tract Features:** Represent the configuration and dynamics of the speaker's vocal tract during speech.

Speech signals are inherently nonstationary, thus they are segmented into small frames to render them stationary for analysis, focusing primarily on excitation source features, vocal tract characteristics, prosodic features, and various combinations of these features [19].

Classifier Design and Effectiveness

Speech Emotion Recognition (SER) is employed to classify the underlying emotions in any given utterance. The classification of SER can be approached through two distinct methods: traditional classifiers and deep learning classifiers 3.3. While numerous classifiers have been applied in SER systems, determining the most effective one poses a challenge, leading to ongoing pragmatic research in the field.

SER systems commonly leverage various traditional classification algorithms. The learning algorithm predicts a new class input by utilizing labeled data that recognizes respective classes and samples through the approximation of the mapping function. Following the training process, the remaining data is employed to test the classifier's performance. Examples of traditional classifiers include Gaussian Mixture Model, Hidden Markov Model, Artificial Neural Network, and Support Vector Machines. Other traditional classification techniques, such as k-Nearest Neighbor, Decision Trees, Naïve Bayes Classifiers, and k-means, are also frequently preferred. Additionally, an ensemble technique is employed for emotion recognition, combining different classifiers to achieve more robust and acceptable results.

The effectiveness of a SER system significantly depends on the choice of classifiers. Various machine learning classifiers have been implemented and evaluated for their performance in SER [21]:

- **Single Classifiers:** These involve using one classifier type to predict emotional states.
- **Multiple Classifiers:** This approach uses several classifiers, each trained on different aspects of speech features to improve recognition accuracy.
- **Hybrid Classifiers:** Combining features or methods can result in hybrid classifiers that leverage the strengths of various approaches.
- **Ensemble Classifiers:** These classifiers use a group of models to better generalize over different datasets, enhancing robustness and accuracy.

The design of speech databases, crucial for assessing classifier effectiveness, varies based on environmental conditions and language specifics. It's essential that the features chosen for classifier design are robust enough to perform effectively across different speech emotion contexts. Classifiers are typically trained and tested within the same database to ensure consistency in performance evaluation [21].

3.2.2 Summary

In the domain of machine learning for emotion recognition, traditional methodologies typically rely on manual feature extraction and established classifiers. Essential speech features such as pitch, energy, and formants are extracted and then utilized to feed classifiers like Support Vector Machines (SVMs) and Gaussian Mixture Models (GMMs). An overview of such models is probided in the following table 3.2. These traditional models are somewhat effective, yet they may not capture the complex and high-dimensional patterns in data as efficiently as more modern deep learning approaches, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs).

In this thesis, we will depart from these traditional models in favor of exploring deep learning techniques. The subsequent section will detail these deep learning algorithms and their application to emotion recognition, highlighting their advantages over traditional methods in capturing nuanced emotional expressions in speech.

Table 3.2: This table provides a quick overview over existing traditional models used for speech emotion recognition

No.	Methods	Methodology overview	Results	References
1	GMMs	All features were used to model GMMs on the frame level. Emotions were classified in three categories. Method has been tested on two different datasets	85% accuracy	Neiberg et al. (2006)
2	SVM	Emotions were classified into 7 different categories. Majority of features that were used are in time domain. Methodology was testing using one dataset	81% recognition rate	Lalitha et al. (2014)
3	GMM Supervector-based SVM vs GMM	GMM supervectors for calculated for each utterance, which were further used as input for SVM. Utterances were classified to 5 emotions	GMM Supervector-based SVM significantly outperforms standard GMM system	Hu et al. (2007)
4	HMM and SVM	Utterances were classified in 5 categories. Feature selection was performed using SFS. Both HMM and SVM were used for classification separately to compare	Recognition rate of 99.5% through HMM, 88.9% through SVM	Lin and Wei (2005)
5	SVM	Different combination of features was used to develop different SVM models. Best one was chosen based on accuracy rate	Accuracy rate of 91.3% for Chinese database, 95.1% for Berlin databases	Pan et al. (2012)
6	Hybrid SVM-Belief Network Architecture	Utterances were classifieds into 7 emotions. Hybrid system was built using SVM and Belief Network. Results were integrated in a soft decision fusion using MLP	Error rate of 8.0%	Schuller et al. (2004)
7	SVM, LDA, QDA, HMM	Important features were selected, multiple methods were used	Accuracy rate of 70.1% (4 emotions), 96.3% (2 emotions) using GSVM	Kwon et al. (2003)
8	HMM (bimodal, integrating audio and video)	Hybrid method, both video and audio sources were used to classify emotions into 4 categories	Approximate accuracy of 70% through video source, 30% through audio, and 72% through bimodal	Silva and Ng(2000)
9	Nearest-mean criterion, model each class with Gaussian distribution and classify test samples	Features that give highest recognition rates are selected. Both video and audio sources were used to classify emotions into 6 categories	Best accuracy of 77.8% through audio source, 97.2% through audio and video source	Chen et al. (1998)
10	k-nearest neighbor, neural network, ensemble of neural network	Emotions were classified into 5 categories. Recognition rate of each emotion was calculated. Accuracy rate of each emotion was determined to find out which emotions are being categorized more accurately	Accuracy of 55% through K-nearest neighbors, 65% through neural network, 70% through ensemble of neural network. Accuracy of classifying fear was worst, while anger and sadness was best	Petrushin (2000)

3.3 Deep Learning Approaches for SER

Deep Neural Networks (DNNs) are a class of machine learning algorithms inspired by the biological neural networks that constitute human brains. These sophisticated models consist of multiple layers of interconnected nodes, and they are highly adept at identifying complex patterns and relationships within large datasets. DNNs are particularly impactful in the field of affective computing for tasks such as emotion recognition from speech.

Central to the effectiveness of DNNs is their ability to autonomously derive hierarchical feature sets from raw data inputs. In the realm of speech emotion recognition, DNNs are exceptionally skilled at detecting subtle acoustic nuances linked with various emotional states. This capability is enhanced by using Recurrent Neural Networks (RNNs) equipped with Long Short-Term Memory (LSTM) units, which are crucial for processing the temporal aspects and dependencies of spoken language.

The operational mechanism of DNNs starts either with extracting critical features from speech, such as Mel-frequency cepstral coefficients (MFCCs), which capture the audio's frequency components, or providing the raw speech data as input. These features are fed into the neural network, initiating a training phase on annotated datasets. Throughout this phase, the DNN adjusts its internal weights and biases to align the feature inputs with corresponding emotional labels, a process refined through continual optimization iterations.

What sets DNNs apart in the field of emotion recognition is their nuanced capability to perceive fine variations in speech, such as changes in tone, pacing, and intonation, all of which are indicative of underlying emotions. This sensitivity allows the models to generalize effectively across different emotional expressions and ensures robust performance in practical applications.

As interest in SER expands within the research community—fueled by advancements in technology and greater accessibility to computational resources—deep learning methodologies are increasingly being integrated into this area. The subsequent sections will outline cutting-edge approaches in this domain, forming the foundation upon which the model will be developed and potential shortcomings of existing frameworks address. An overview of deep learning approaches below is provided by Table 3.3.

3.3.1 Convolutional Recurrent Neural Network (CRNN)

The Convolutional Recurrent Neural Network (CRNN) merges the spatial feature detection capabilities of Convolutional Neural Networks (CNNs) with the sequence modeling strengths of Recurrent Neural Networks (RNNs). This combination allows CRNNs to effectively process data that has both spatial and temporal dimensions, making them highly effective in tasks such as emotion recognition from speech signals [20].

Architecture Overview

In a CRNN, the initial layers are convolutional, designed to extract spatial features from the input data. These features are then processed by recurrent layers, which capture temporal dependencies using mechanisms such as Long Short-Term Memory (LSTM) or Gated Recurrent Units (GRUs). This architecture makes CRNNs suitable for applications where both the content of the signal and its temporal characteristics are important.

Applications and Performance

CRNNs are utilized for enhancing one-dimensional signals, such as in audio processing, where they can apply filters like Mel and Gammatone to improve signal clarity by removing noise. The integration of convolutional and recurrent layers enables CRNNs to achieve high accuracy and low loss rates during both training and testing phases, demonstrating their robustness and efficiency in handling complex tasks [20].

3.3.2 CNN Bidirectional LSTM (CNN-BiLSTM)

Wang et al. [39] introduce a novel transformer-based framework named DWFormer, designed specifically for the speech emotion recognition field. This framework is adept at identifying significant temporal regions at varying scales both within and between samples. Empirical evidence shows that DWFormer surpasses previous state-of-the-art methods in performance. Through an ablation study, the utility of the Dynamic Local Window Transformer (DLWT) and Dynamic Global Window Transforme (DGWT) modules within this framework is validated. Given its capability to pinpoint critical information, plans are underway to deploy DWFormer in the study of pathological speech recognition, aiming to aid researchers in analyzing the effects of diseases on speech articulation.

3.3.3 PCNSE

In full name the Parallel Convolutional Layers (PCN) integrated with Squeeze-and-Excitation Network. Zhao et al. [45] introduce an advanced deep neural network architecture that integrates Connectionist Temporal Classification (CTC) loss for targeted use in discrete speech emotion recognition. The efficacy of this innovative approach is validated through rigorous testing on two key emotion corpora: the Interactive Emotional Dyadic Motion Capture (IEMOCAP) and the FAU-Aibo Emotion corpus (FAU-AEC). The experimental outcomes highlight the suitability of this method for discrete SER, where it achieves a weighted accuracy (WA) of 73.1% and an unweighted accuracy (UA) of 66.3% on the IEMOCAP dataset. Furthermore, it also records an unweighted accuracy of 41.1% on the FAU-AEC dataset.

3.3.4 TIM-Net

In full words the Temporal-aware bI-direction Multi-scale Network. Ye et al. [42] present a cutting-edge approach for temporal emotional modeling in their paper, introducing TIM-Net which architecture is shown in figure 3.1. This model is designed to learn multi-scale contextual affective representations across various time scales. TIM-Net excels at capturing long-range temporal dependencies using bi-directional temporal modeling and dynamically fuses multi-scale information to adeptly adjust to variations in temporal scale.

The findings from experimental evaluations underscore the importance of leveraging context information with dynamic temporal scales for the speech emotion recognition task. Additional insights from ablation studies, visualizations, and domain generalization analyses further substantiate the benefits of TIM-Net. Looking forward, Ye proposes to explore the disentanglement of emotional and speech content within this temporal modeling framework to enhance generalization across different SER corpora.

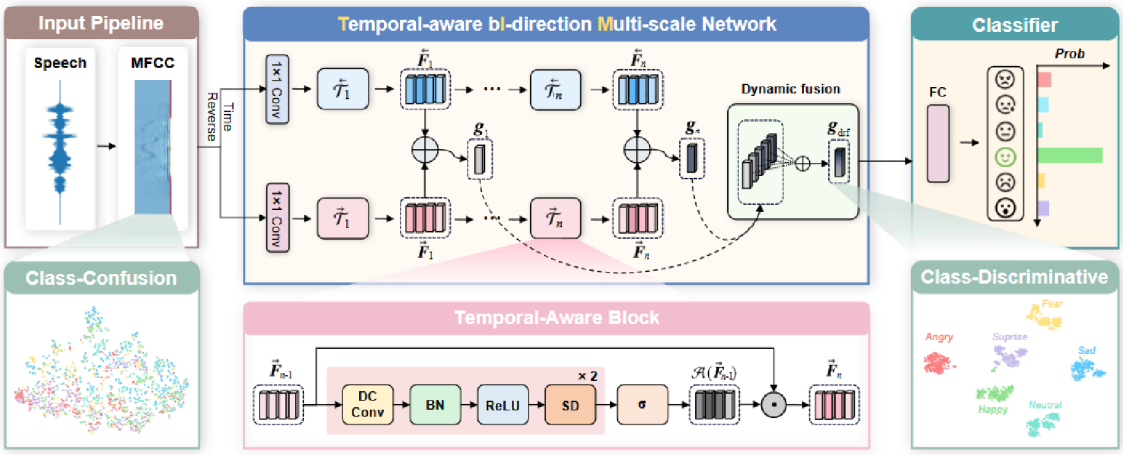


Figure 3.1: The architecture of TIM-Net is specifically designed for extracting affective features and includes two main components: a bi-directional module and a dynamic fusion module. It is important to note that the forward $\vec{\tau_j}$ and backward $\vec{\tau_j}$ components of the bi-directional module share the same structural design but differ in the inputs they process. [42]

3.3.5 DNN & ELM

Han et al. [14] introduce a novel approach in their study by employing a Deep Neural Network (DNN) to estimate emotional states from individual speech segments within an utterance. These segment-level estimations are then aggregated into an utterance-level feature vector. Subsequently, an Extreme Learning Machine (ELM) is utilized to perform the emotion recognition for the entire utterance. The experimental results from this study suggest that leveraging a DNN in conjunction with an ELM significantly enhances the performance of emotion recognition from speech signals. This method shows great promise in extracting and learning emotional information from low-level acoustic features through neural networks.

3.3.6 Multimodal System by Busso et al.

This study conducted by Busso et al. [7] delves into the statistical analysis of pitch contours in speech. Initially, pitch features extracted from emotional speech samples are compared to those from neutral speech using symmetric Kullback-Leibler distance to establish differences. Subsequently, the emotional discriminative power of these pitch features is assessed through the comparison of nested logistic regression models.

The findings reveal that broader pitch contour statistics—such as mean, maximum, minimum, and range—hold greater emotional significance than those describing the shape of the pitch. Furthermore, it is determined that pitch statistics evaluated at the utterance level yield more accurate and robust results than those assessed over shorter speech segments, such as voiced sections.

Building upon these insights, a binary emotion detection system is developed to differentiate emotional from neutral speech. The system employs a novel two-step methodology: initially, reference models for pitch features are trained using neutral speech to establish a baseline. Input features are then compared against these models to gauge similarity (for

neutral speech) or disparity (for emotional speech). The effectiveness of this approach is validated across four acted emotional databases, encompassing various emotional categories, recording settings, speakers, and languages.

The results demonstrate that this system achieves a recognition accuracy of over 77% using only pitch features, a significant improvement over the baseline of 50%. Compared to traditional classification methods, this novel approach exhibits enhanced accuracy and robustness. [7]

3.3.7 Audio Spectrogram Transformer (AST)

The Audio Spectrogram Transformer (AST) represents a groundbreaking shift in audio classification methodologies. Unlike traditional models, AST is the first to utilize a purely attention-based mechanism, devoid of convolutional layers, tailored specifically for audio tasks. It accommodates variable-length inputs and has been rigorously evaluated across several audio classification benchmarks. Remarkably, AST achieves a mean average precision (mAP) of 0.485 on AudioSet, 95.6% accuracy on ESC-50, and 98.1% accuracy on Speech Commands V2 [13]. The model's architecture, inspired by the successes of Transformer technology in natural language processing, adapts this approach to handle audio data through sophisticated time-frequency representations known as spectrograms.

Architecture

AST's architecture, also shown in Figure 3.2 fundamentally transforms the approach to audio signal processing by employing the Transformer model, which relies on self-attention mechanisms rather than the traditional convolutional neural networks (CNNs). This shift allows AST to dynamically weigh the importance of different segments of the audio without the constraint of local receptive fields typically imposed by CNNs. The Transformer layers in AST analyze the entire audio spectrum holistically, enabling it to capture complex patterns and dependencies that are crucial for accurate audio classification.

Input Representation

The input to AST is a mel-spectrogram, a sophisticated transformation of raw audio that reflects human perception of sound more accurately than standard spectrograms. This transformation involves segmenting the audio signal into short frames, applying a Fast Fourier Transform (FFT) to each frame to obtain the frequency spectrum, and then warping the frequencies onto the mel scale. This scale emphasizes perceptual relevance rather than linear frequency distribution, making it particularly effective for tasks involving human auditory perception, such as speech and music classification.

Applications

AST's flexibility and robustness allow it to excel in a variety of audio classification tasks. It has been successfully applied in environmental sound classification, where it identifies and categorizes natural and urban sounds, music genre classification, distinguishing among different musical styles, and speech emotion recognition, detecting emotional states from speech patterns. The model's ability to handle long-range dependencies and its sensitivity to the temporal dynamics of audio make it exceptionally suited for these complex audio analysis tasks.

Advantages

One of the foremost advantages of the AST model is its convolution-free architecture, which enables it to focus attention variably across different parts of an audio signal. This capability allows AST to perform exceptionally well on tasks that require a nuanced understanding of audio content. The self-attention mechanism assesses relationships across all parts of the audio signal, fostering a comprehensive understanding that often surpasses traditional methods, particularly in discerning subtle audio features essential for high-level audio analysis.

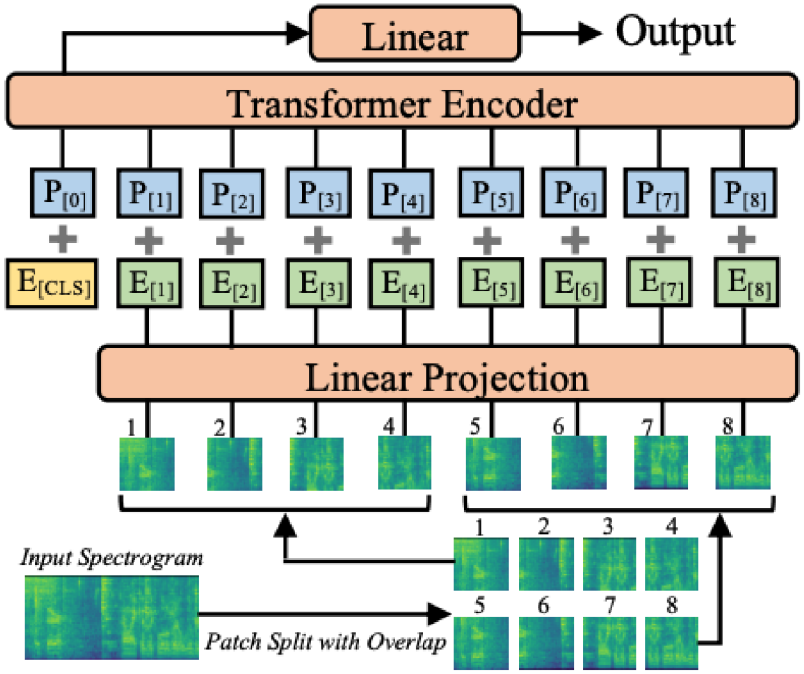


Figure 3.2: Architecture proposed by the authors of AST [13]

Table 3.3: The table provides a quick overview over existing deep learning models used for speech emotion recognition

Model	*Database(s)*	*Preprocessing*	*Results*
C-RNN	RAVDESS	-	70 % / -
CNN-BiLSTM	IEMOCAP	Framing with the frame length of 25 ms and hop length of 10 ms	92 % / 91.28 %
PCNSE	IEMOCAP, FAU-AEC	Framing with the frame length of 25 ms and hop length of 10 ms, Hamming window applied on each frame	66.3 % / 73.1 % for IEMOCAP, 41.1 % / - for FAU-AEC
TIM-Net	EmoDB, IEMOCAP, RAVDESS, SAVEE, CASIA, EMOVO	Framing with frame length of 50 ms and hop length of 12.5 ms, Hamming window applied on each frame.	2.34 % / 2.36 % (avg. improvement)
DNN, then ELM	-	DNN have been used to produce emotion state probability distribution for each segment, which was to construct utterance-level features. These features were fed into ELM to identify utterance-level emotions (5 categories)	Accuracy rate of 45% through base HMM improved to 54.3% through proposed approach
Multimodal system	-	Both audio and visual information have been used. Results were integrated through fusion. Emotions were classified into 4 categories	Accuracy of 70.9% through acoustic source, 85% through facial source, 89.1% through bimodal system
AST	AudioSet, Speech Commands V2	-	95.6% accuracy on ESC-50, and 98.1% accuracy on Speech Commands V2.

3.3.8 Deep Belief Networks

Deep Belief Networks (DBNs) represent another class of deep learning models, distinct in architecture and training methodology from traditional Deep Neural Networks (DNNs). While both DNNs and DBNs fall under the umbrella of deep learning, the key difference lies in the hierarchical structure and learning approach.

Unlike the feedforward architecture of DNNs, DBNs are composed of multiple layers of stochastic, latent variables, where each layer models dependencies among these variables. DBNs consist of a generative layer known as the Restricted Boltzmann Machine (RBM) and a top discriminative layer for classification tasks.

In the realm of emotion recognition from speech, DBNs offer an alternative approach to capturing intricate patterns in acoustic signals. The training process of a DBN involves layer-wise unsupervised pretraining, where each layer is trained to learn a compact representation of the input data. This pretraining allows DBNs to automatically extract hierarchical features and uncover complex relationships within the data.

The versatility of DBNs lies in their ability to adapt to varying levels of abstraction in the input features. In the context of speech, this can be advantageous for recognizing emotions, as it allows the model to capture both low-level acoustic details and high-level contextual information.

While DNNs have proven effective in learning representations directly from labeled data through supervised learning, DBNs, with their unsupervised pretraining, may excel in scenarios with limited labeled training samples. This characteristic makes them valuable in tasks where obtaining large labeled datasets is challenging [41].

Deep Boltzmann Machine

Deep Boltzmann Machines (DBMs) represent a specialized class of unsupervised deep learning models, sharing some similarities with Deep Belief Networks (DBNs) in their use of Boltzmann Machines. However, DBMs introduce a more complex and interconnected architecture, enabling the modeling of higher-order dependencies in the data.

In the context of emotion recognition from speech, Deep Boltzmann Machines offer a unique approach to capturing the intricate patterns present in acoustic signals. Unlike the layer-wise unsupervised pretraining employed in DBNs, DBMs utilize a joint training approach that considers all layers simultaneously. This allows DBMs to model complex relationships and dependencies across multiple layers, potentially yielding richer representations of emotional cues in speech.

The architecture of a DBM comprises visible and hidden layers, where each layer contains a set of stochastic binary units. The connectivity pattern between layers is symmetric, allowing for bidirectional information flow. This bidirectional connectivity enables DBMs to capture not only the direct relationships between input features but also more abstract and high-level dependencies.

Training a Deep Boltzmann Machine involves adjusting the weights and biases to maximize the likelihood of observed data. This process is inherently unsupervised, making DBMs particularly suitable for scenarios where labeled emotion data is scarce or unavailable. The model learns a probabilistic generative model of the input data, allowing it to capture the underlying structure of the acoustic features associated with different emotional states.

The use of DBMs in emotion recognition highlights their capability to automatically learn hierarchical representations of speech data. By modeling dependencies across multiple layers, DBMs have the potential to capture nuanced and contextually rich patterns, contributing to more sophisticated emotion recognition systems [36].

Restricted Boltzmann Machine

While both RBMs and DBMs are types of Boltzmann Machines used for unsupervised learning, DBMs extend the architecture to include multiple hidden layers with bidirectional connectivity. This architectural difference allows DBMs to capture more complex relationships within the data, making them particularly useful for tasks requiring the modeling of intricate dependencies, such as in emotion recognition from speech [31].

3.3.9 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) represent a class of neural networks designed to effectively capture sequential information, making them well-suited for analyzing time-series data like speech signals. In the context of emotion recognition from speech, RNNs offer a unique approach to understanding the temporal dynamics inherent in spoken language [37].

Key Characteristics of RNNs:

- **Temporal Sequences**: RNNs are specialized in handling sequences of data by maintaining hidden states that capture information from previous time steps. This makes them particularly powerful for tasks where the order and context of input data matter, such as in understanding the emotional nuances expressed in speech.
- **Long Short-Term Memory (LSTM)**: To address challenges like vanishing gradients and the inability to capture long-range dependencies, RNNs often incorporate LSTM cells. LSTMs are capable of learning and remembering information over extended sequences, making them effective for modeling the temporal aspects of speech.
- **Feature Extraction**: RNNs process acoustic features extracted from speech signals, such as Mel-frequency cepstral coefficients (MFCCs), pitch, and energy. These features serve as inputs to the network, enabling it to learn patterns associated with different emotions.
- **Training Process**: During training, RNNs learn to map the sequential acoustic features to corresponding emotion labels. The training process involves adjusting the weights of the network using backpropagation through time (BPTT), allowing the model to capture temporal dependencies and improve its ability to recognize emotions.
- **Real-time Inference**: Once trained, RNNs can perform real-time emotion recognition from speech. Given a new speech sample, the model processes the input sequentially and generates predictions based on the learned temporal dependencies.
- Challenges: Despite their effectiveness, RNNs have limitations, such as difficulties in capturing very long-term dependencies and susceptibility to vanishing or exploding gradients. These challenges have led to the development of more advanced architectures like Long Short-Term Memory networks (LSTMs) and Gated Recurrent Units (GRUs).

Applications in Emotion Recognition:

Temporal Dynamics: RNNs excel in capturing the dynamic nature of emotional expression in speech, where the timing and sequence of acoustic features play a crucial role.

Contextual Understanding: The ability to maintain hidden states allows RNNs to consider context from previous time steps, aiding in the interpretation of emotional cues within a broader context.

Multimodal Integration: RNNs can be employed in multimodal emotion recognition systems, combining information from speech with other modalities like text or facial expressions for a more comprehensive understanding of emotional states.

3.3.10 Long Short-Term Memory

Long Short-Term Memory (LSTM) networks are a specialized type of Recurrent Neural Network (RNN) architecture designed to address the challenges of modeling long-range dependencies and capturing temporal dynamics in sequential data. In the context of emotion recognition from speech, LSTMs offer enhanced capabilities for understanding and interpreting the nuanced patterns associated with different emotional states [16].

Key Characteristics of LSTM Networks:

- **Memory Cells:** LSTMs introduce a memory cell, a fundamental component that allows the network to store and retrieve information over extended time intervals. This mitigates the vanishing gradient problem encountered in traditional RNNs, enabling LSTMs to capture long-term dependencies crucial for understanding emotional expressions in speech.
- **Gates for Information Flow:** LSTMs incorporate gating mechanisms, including the input gate, forget gate, and output gate. These gates regulate the flow of information into, out of, and within the memory cell. The ability to selectively update and forget information enhances the network's capacity to discern relevant emotional cues in speech.
- **Sequential Processing:** LSTMs process sequential input data, such as acoustic features extracted from speech signals, in a step-by-step manner. At each time step, the network considers the current input, updates its hidden state, and makes predictions based on the learned temporal dependencies.
- **Feature Learning:** LSTMs automatically learn hierarchical representations of sequential data, allowing them to extract and emphasize salient features associated with different emotional states in speech.
- **Training Process:** During training, LSTMs adjust their weights through backpropagation through time (BPTT). The architecture's ability to capture long-term dependencies facilitates more effective learning of patterns within emotional expressions, contributing to improved accuracy in emotion recognition.

Applications in Emotion Recognition:

- **Temporal Context**: LSTMs excel in capturing the temporal context of emotional expressions in speech, enabling the model to consider not only the current acoustic features but also the historical context.
- **Complex Dependencies**: The memory cell and gating mechanisms enable LSTMs to capture complex dependencies in sequential data, making them well-suited for tasks where understanding the interplay of various acoustic features is crucial.

- **Real-time Inference**: Trained LSTMs can perform real-time emotion recognition, making them suitable for applications that require immediate feedback based on incoming speech signals.

- **Transfer Learning**: LSTMs can benefit from transfer learning by initializing weights with pre-trained models on large datasets. This is particularly useful in emotion recognition tasks when labeled data is limited.

3.3.11 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a type of deep learning model that has been successfully applied to various tasks, including image and speech processing. In the context of emotion detection from human speech, CNNs can be utilized to automatically learn relevant features from the audio data [44]. Here's an overview of how CNNs can be employed for emotion detection from speech:

1. **Data Representation:** Spectrogram Generation: Audio data is often converted into a spectrogram, which is a visual representation of the spectrum of frequencies in a sound signal as they vary with time. This conversion is crucial for extracting patterns from the audio signals

2. **Convolutional Layers:**

   Feature Extraction: Convolutional layers in a CNN are responsible for learning hierarchical features. In the context of spectrograms, the convolutional filters can learn to detect patterns and features that are indicative of certain emotional characteristics in the speech signal.

3. **Pooling Layers:**

   Downsampling: Pooling layers are often used to reduce the spatial dimensions of the feature maps obtained from convolutional layers. This downsampling helps retain the most important information while reducing computational complexity.

4. **Flattening and Fully Connected Layers:**

   Decision Making: The flattened output from the convolutional and pooling layers is fed into one or more fully connected layers. These layers serve as classifiers and learn to map the features extracted by the earlier layers to specific emotion classes.

5. **Softmax Activation:**

   Output Layer Activation: The final layer typically uses a softmax activation function, which converts the raw output scores into probability distributions over different emotion classes. This allows the model to provide a probability for each emotion class.

6. **Training:**

   Supervised Learning: CNNs are trained in a supervised manner, meaning that they are provided with labeled examples of speech data and their corresponding emotion labels. The model adjusts its parameters during training to minimize the difference between its predicted emotions and the true emotions.

7. **Evaluation:**

   Testing and Validation: The trained model is evaluated on a separate set of data that it has not seen before to assess its ability to generalize to new instances. Metrics such as accuracy, precision, recall, and F1 score are commonly used for evaluation.

8. **Hyperparameter Tuning:**

   Optimization: The model's hyperparameters, such as learning rate, number of layers, and filter sizes, may need to be fine-tuned to achieve optimal performance.

9. **Real-time Inference:**

   Deployment: Once trained, the model can be used for real-time inference, taking in new audio data and predicting the associated emotion.

Considerations:
Data Quality and Quantity: Adequate and diverse training data is crucial for the model to generalize well to different speakers, accents, and emotional expressions. Model Complexity: The architecture of the CNN, including the number of layers and parameters, needs to be carefully chosen to balance complexity and generalization.

In summary, CNNs provide a powerful framework for automatically learning hierarchical features from spectrograms, making them well-suited for tasks like emotion detection from human speech.

3.4 Summary

The following table 3.4 summarizes various deep learning models discussed in the chapter, outlining their advantages and disadvantages in the context of speech emotion recognition.

Table 3.4: Comparison of Deep Learning Models for Speech Emotion Recognition

Model	**Pros**	**Cons**
Deep Neural Networks (DNNs)	Highly effective in identifying complex patterns and relationships, good at generalizing across different emotional expressions.	Requires significant computational resources, potential for overfitting on complex datasets.
Convolutional Recurrent Neural Network (CRNN)	Merges spatial feature detection of CNNs with temporal modeling of RNNs, suitable for data with both spatial and temporal dimensions.	Complex architecture can be challenging to tune and optimize, potentially high computational load.
CNN Bidirectional LSTM (CNN-BiLSTM)	Combines CNN's feature extraction capabilities with LSTM's temporal accuracy, highly effective in complex temporal sequence tasks.	Training can be computationally intensive and slow, may require large datasets to train effectively.
Parallel Convolutional Neural Networks with Squeeze-and-Excitation (PCNSE)	Targets discrete emotion recognition efficiently, shows strong performance on specific benchmarks.	Performance can vary significantly across different datasets, may struggle with generalization across diverse emotional states.
Temporal-aware bI-direction Multi-scale Network (TIM-Net)	Capable of capturing long-range temporal dependencies, adjusts dynamically to variations in temporal scale.	Complexity of the model might lead to difficulties in training and require extensive computational resources.
Audio Spectrogram Transformer (AST)	Utilizes a purely attention-based mechanism without convolutional layers, excellent at handling long-range dependencies and subtle audio features.	As a newer model, may lack extensive real-world testing across varied SER applications.

Given the various options, the author has chosen to implement the Audio Spectrogram Transformer (AST) in their network. The decision is based on several factors:

- **Attention Mechanism**: AST leverages an advanced attention-based mechanism which is crucial for identifying subtle nuances in speech that are indicative of emotional states.
- **Handling Long-Range Dependencies**: Unlike traditional models that might struggle with long sequences, AST excels in managing long-range dependencies, making it well-suited for continuous speech emotion recognition.
- **Model Efficiency**: Despite its sophisticated capabilities, AST is designed to operate efficiently in terms of computational resources compared to models that combine CNNs and RNNs/LSTMs.
- **Innovative Approach**: The purely attention-based approach without reliance on convolutional layers positions AST at the cutting edge of audio processing technology, promising enhanced performance on SER tasks.

The choice of AST highlights a strategic move towards utilizing state-of-the-art technology to address the intricacies of speech-based emotion recognition, aiming to achieve both high accuracy and efficiency. The proposal is explained in more depth in chapter 4.

Chapter 4

Proposed Methodology

This chapter describes the idea of my implementation, it explains the relevant topics in theory and describes the practical implementations of them.

After thoughtful consideration, I've opted not to create a custom deep learning model for Speech Emotion Recognition (SER) and instead chose to fine-tune a pretrained model. The primary motivation behind this decision is to strive for improved accuracy and efficiency in recognizing emotional cues in speech. By fine-tuning, I can focus particularly on the latter layers of the pretrained model, where intricate details of audio features and nuances related to emotions are likely captured. This tailored approach allows for a more precise adaptation to the specific characteristics of my SER dataset, aiming to enhance the model's ability to discern subtle emotional variations in speech. The emphasis on the final layers during fine-tuning is strategic, seeking to leverage the knowledge encoded in the pretrained model while ensuring it aligns more closely with the unique features of emotions expressed in my dataset. This decision reflects a commitment to achieving a more accurate and contextually relevant SER model by strategically refining the pretrained model's outputs.

Fine-tuning a pretrained model for Speech Emotion Recognition (SER) using deep learning involves leveraging a model that has already been trained on a large dataset and adapting it to the specific characteristics of my target SER dataset. Below is a proposed method for fine-tuning the pretrained model:

- Modifying the Input Layer: The input layer of the pretrained model is adjusted to accommodate the features specific to the SER dataset, ensuring that the input layer matches the dimensionality and type of features present in the dataset, such as mel-frequency cepstral coefficients (MFCCs) or spectrograms.
- Freezing Base Layers: The weights of the initial layers of the pretrained model are frozen. These layers have learned general audio representations that can be useful for SER, and freezing them helps to prevent overfitting.
- Adding Additional Layers: New layers (fully connected or convolutional layers) are appended to the pretrained model to adapt it to the target SER task.
- Initializing Weights: The weights of the newly added layers are initialized randomly or using a suitable initialization technique. This step is crucial to prevent catastrophic forgetting of the pretrained features.

- Modifying the Loss Function: The loss function is adjusted to match the SER task requirements. Cross-entropy loss is commonly used for classification tasks, including emotion recognition.
- Fine-tuning the Model: The model is trained on the SER dataset while keeping the base layers frozen. This allows the model to adapt to the specific emotional characteristics of the dataset without drastically altering the learned audio representations.
- Unfreezing and Continuing Training: Optionally, some of the top layers of the pre-trained model are unfrozen after several epochs to allow fine-tuning on the SER dataset. This can be beneficial if the dataset is large enough to avoid overfitting.
- Applying Regularization Techniques: Regularization techniques like dropout or batch normalization are applied to prevent overfitting during fine-tuning.
- Optimizing Hyperparameters: Experimentation with learning rates, batch sizes, and other hyperparameters is conducted to find the optimal configuration for fine-tuning on the SER dataset.
- Evaluating Performance: The fine-tuned model is evaluated on a validation set to monitor its performance and adjust hyperparameters if necessary. Its performance is assessed on a separate test set to ensure generalization.

4.1 Pre-trained model

For the pretrained model, I have decided to choose the AST model, as it provided with great baseline accuracy and will thus provide a very high basis. The model is described at 3.3.7. The proposed architecture of the implementation is shown in the Figure 4.1.

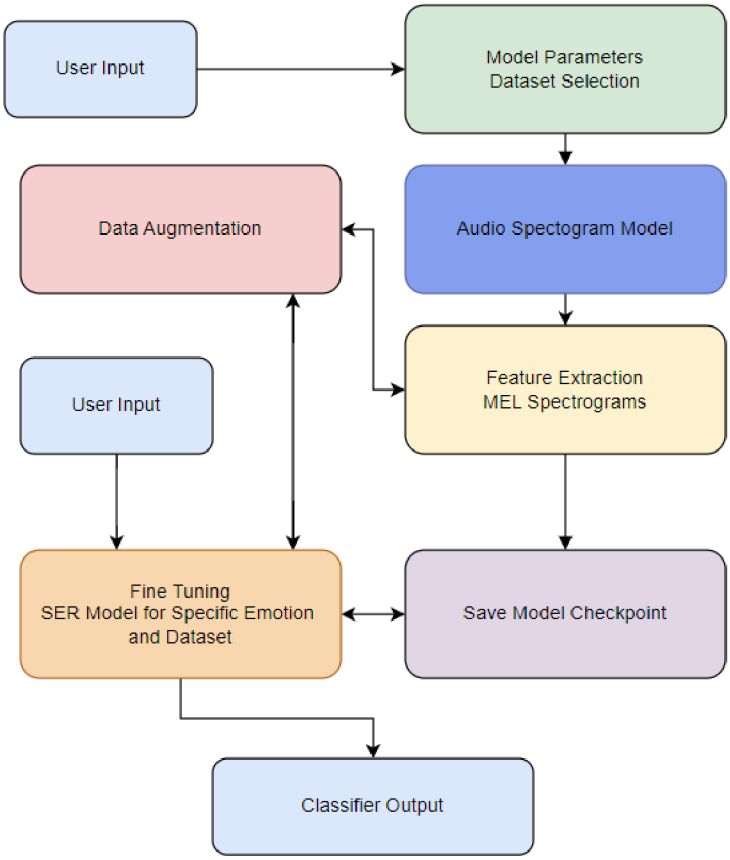


Figure 4.1: Architecture of a Neural Network based on the AST Model. The process begins with user input, where model parameters and dataset selection are specified. Features are extracted by converting audio signals into MEL spectrograms. Data augmentation techniques are applied to improve model generalization. The model undergoes fine-tuning, specifically for enhancing emotion recognition in psychological evaluations. Finally, the classifier outputs a probability distribution across the specified classes.

4.2 Model parameters

In the realm of deep learning, the effectiveness of a model heavily relies on the configuration of various parameters. These parameters dictate how the model learns from data, adapts its internal representations, and ultimately makes predictions. In this section, we delve into key parameters commonly encountered in deep learning frameworks, elucidating their significance and the impact of their manipulation on model behavior and performance. From fundamental parameters like sample rate and batch size to more intricate concepts such as learning rate and cross-validation folds, grasping the nuances of these parameters is essential for fine-tuning models and achieving optimal results in deep learning applications.

- **sample_rate**: Sample rate refers to the number of samples of audio carried per second, measured in Hertz (Hz). It represents the frequency at which audio signals are captured. When changing the sample rate, you're essentially altering the granularity

of the audio data. Higher sample rates capture more detail but also require more computational resources and storage space.

- **n_epochs**: This parameter represents the number of epochs, or complete passes through the entire dataset, during the training phase. Increasing the number of epochs allows the model to see the data more times, potentially improving its performance. However, too many epochs can lead to overfitting, where the model memorizes the training data instead of learning generalizable patterns.

- **batch_size**: Batch size refers to the number of samples processed before the model's parameters are updated. A larger batch size generally leads to faster training because it allows for more parallel computations, but it requires more memory. Smaller batch sizes may result in slower convergence but can help the model generalize better by updating weights more frequently.

- **lr**: LR stands for learning rate, which determines the size of the step the optimizer takes during the parameter update process. A higher learning rate allows for faster convergence but may lead to overshooting and instability. Conversely, a lower learning rate might result in slower convergence but can help the model find a more precise minimum of the loss function.

- **n_folds**: This term typically refers to the number of folds used in cross-validation, a technique for assessing the performance and generalization ability of a model. Increasing the number of folds provides a more robust estimate of the model's performance but also increases computational cost.

- **seed**: Seed is a parameter used to initialize random number generators. Setting a seed ensures reproducibility, meaning that running the model with the same seed will produce the same results each time. This is crucial for experimentation and debugging.

- **mel_filter_banks**: Mel-filter banks are used in audio processing to convert the linear frequency scale of audio signals into the mel scale, which better approximates the human auditory system's response to different frequencies. Adjusting the parameters of the mel-filter banks can affect the spectral representation of the audio data, potentially impacting the model's ability to extract relevant features.

- **frames**: Frames refer to the temporal segmentation of audio signals into smaller chunks. This parameter determines the size of each frame. Changing the frame size can influence the temporal resolution of the input data, affecting the model's ability to capture temporal patterns in the audio signal.

4.2.1 Experiment: Finding the Best Model Configuration

Objective: To identify the optimal configuration of parameters for a deep learning model in the context of audio classification.

Experimental Procedure:

1. **Parameter Initialization:** Initialize the parameter grid with different combinations of values for each parameter. For example:

- Sample rate: [8 kHz, 16 kHz, 22.05 kHz, 44.1 kHz]
- Number of epochs: [50, 100, 150]
- Batch size: [16, 32, 64]
- Learning rate: [0.001, 0.01, 0.1]
- Number of cross-validation folds: [3, 5, 10]
- Mel-filter bank settings: [Default, Custom]
- Frame size: [10 ms, 20 ms, 30 ms]

2. **Model Training and Evaluation:** Train the deep learning model for each parameter combination using the training pipeline. Use a validation set to evaluate model performance for each configuration.

3. **Performance Evaluation:** Assess the performance of each trained model using metrics such as accuracy, precision, recall, and F1-score. Compare the results across different parameter combinations.

4. **Parameter Optimization:** Analyze the performance results to identify the parameter combinations that yield the best performance metrics. Look for trends and patterns in how changes to each parameter affect the model's performance.

5. **Validation:** Validate the final selected model(s) on a separate test set to ensure generalization performance.

Results and Analysis:

1. **Optimal Parameter Configuration:** Identify the parameter combination(s) that result in the best performance metrics based on the validation results.

2. **Insights:** Analyze how variations in each parameter impact the model's performance. Gain insights into which parameters have the most significant influence and how they interact with each other.

3. **Conclusion:** Summarize the findings and recommend the best model configuration based on the experimental results. Provide insights into the relationship between parameter settings and model performance, guiding future research and application development.

Conclusion

By systematically exploring variations in the parameters relevant to deep learning models for audio classification, this experiment enables the identification of the optimal model configuration. Through rigorous experimentation, analysis, and validation, researchers and practitioners can develop highly effective deep learning models tailored to specific tasks and datasets.

4.3 Evaluation Metrics

In the field of machine learning, particularly in classification tasks, it is crucial to accurately measure the performance of models. Evaluation metrics provide insights into different aspects of model behavior, such as its precision in predicting positive labels, sensitivity to

capturing relevant instances, and overall error balance. These metrics are indispensable for tuning models, comparing different models, and ultimately selecting the best model for deployment. The following are key metrics used for evaluating the performance of classification models [30].

Accuracy measures the overall correctness of the model, defined as the ratio of correctly predicted observations (both true positives and true negatives) to the total observations in the dataset. This metric is particularly useful as a general indicator of model performance across all classes. Accuracy provides a quick snapshot of the effectiveness of a predictive model, especially in scenarios where all classes are equally important.

$$\text{Accuracy} = \frac{\text{True Positives (TP)} + \text{True Negatives (TN)}}{\text{TP} + \text{TN} + \text{False Positives (FP)} + \text{False Negatives (FN)}}$$

Simplified the accuracy can be denoted as:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Precision measures the accuracy of the model's positive predictions, defined as the ratio of true positives to the total number of predicted positives. This metric is crucial in situations where the cost of a false positive is high (e.g., spam detection).

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{TP} + \text{False Positives (FP)}}$$

Recall (also known as Sensitivity or True Positive Rate) measures the ability of the model to identify all relevant instances, calculated as the ratio of true positives to the total actual positives. High recall is critical in scenarios where missing a positive instance is costly (e.g., disease screening).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{False Negatives (FN)}}$$

F1 Score is the harmonic mean of precision and recall. It is a single metric that combines both precision and recall to provide a balanced view of the model's overall performance, especially useful when the positive class is rare.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Support refers to the number of actual occurrences of each class in the dataset. It is important for understanding the class distribution and ensuring that the evaluation metrics are not biased due to a skewed dataset.

$$\text{Support} = \text{Number of instances for each class}$$

Specificity (also known as True Negative Rate) measures the proportion of actual negatives that are correctly identified by the model, indicating the model's ability to reject false positives. It is particularly relevant in cases where it's crucial to confirm an absence of condition.

$$\text{Specificity} = \frac{\text{True Negatives (TN)}}{\text{TN} + \text{FP}}$$

4.4 Strategies to Improve Evaluation Metrics for an Audio Spectrogram Transformer (AST) Model

To enhance the performance of an Audio Spectrogram Transformer (AST) model in classification tasks, specific strategies can be employed for each key evaluation metric. Below is a list of effective approaches for improving Precision, Recall, F1 Score, Support, and Specificity:

Improving Precision

- **Threshold Adjustment:** Increase the threshold for predicting positive classes to reduce false positives.
- **Data Quality:** Improve the quality of input data, focusing on cleaner, higher-resolution audio spectrograms.
- **Feature Engineering:** Enhance or select features that are more predictive of the positive class.

Improving Recall

- **Threshold Lowering:** Decrease the classification threshold to capture more true positives, at the risk of increasing false positives.
- **Data Augmentation:** Use techniques like time stretching, pitch shifting, and adding background noise to create a more robust model.
- **Model Complexity:** Increase the depth or capacity of the AST model to capture more complex patterns in the data.

Improving F1 Score

- **Model Tuning:** Use grid or random search to find the optimal balance of model parameters that maximize both precision and recall.
- **Ensemble Techniques:** Combine multiple models to leverage their individual strengths, potentially improving both precision and recall.

Improving Support

- **Balanced Datasets:** Ensure the training set is representative of the true population distribution to avoid biases in the model's performance metrics.
- **Resampling Techniques:** Utilize oversampling of minority classes or undersampling of majority classes to balance class distribution.

Improving Specificity

- **Negative Case Enhancement:** Augment the dataset with more varied negative cases to improve the model's learning of what does not constitute a positive class.

- **Anomaly Detection Techniques:** Incorporate methods specifically designed to improve true negative rates, such as anomaly detection algorithms that focus on identifying non-target classes.

4.5 Fine-Tuning for Speech Emotion Recognition

Sd discussed, deep learning models, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have shown promising results in SER. However, fine-tuning an existing deep learning model can further improve its performance, especially when targeting specific emotions.

4.5.1 Fine-Tuning Process

Fine-tuning an existing deep learning model for SER involves adjusting its parameters to better capture the nuances of specific emotions. The process typically consists of the following steps:

1. **Fine-Tuning Strategy:** Define a fine-tuning strategy to adapt the pre-trained model to the target emotion(s). This may involve adjusting hyperparameters, modifying the model architecture, or fine-tuning specific layers of the network.

2. **Data Augmentation:** Apply data augmentation techniques to artificially increase the diversity of the training data. Common augmentation methods for SER include time stretching, pitch shifting, and adding background noise.

3. **Training Procedure:** Train the fine-tuned model using the prepared dataset. Monitor the model's performance on validation data and adjust the fine-tuning strategy as necessary to achieve the desired results.

4.5.2 Fine-Tuning for Specific Emotions

When fine-tuning a model for SER to detect a particular emotion, it's essential to consider the unique acoustic characteristics associated with that emotion. For example, anger may be characterized by high pitch and intensity, while sadness may exhibit lower pitch and slower speech rate.

To fine-tune the model for a specific emotion:

1. **Emotion-Specific Data Selection:** Curate a subset of the dataset containing speech samples predominantly expressing the target emotion. This focused dataset helps the model learn discriminative features for the specific emotion.

2. **Fine-Tuning Parameters:** Adjust the fine-tuning strategy to emphasize features relevant to the target emotion. For instance, increase the weight of emotion-specific loss functions or fine-tune certain layers to extract emotion-specific features more effectively.

3. **Evaluation and Validation:** Evaluate the fine-tuned model's performance on a separate test set containing samples of the target emotion. Use appropriate evaluation metrics, such as accuracy or F1-score, to assess the model's effectiveness in recognizing the desired emotion.

4.5.3 Example: Fine-Tuning for Anger Recognition

As an example, consider fine-tuning a pre-trained AST model for recognizing anger in speech. The fine-tuning process may involve:

- Selecting a pre-trained AST model with high performance on general SER tasks.
- Curating a dataset with speech samples labeled as expressing anger.
- Fine-tuning the model by adjusting hyperparameters and optimizing the model architecture to focus on features indicative of anger, such as high pitch and intensity.
- Evaluating the fine-tuned model's performance on a test set containing anger-labeled speech samples.

Fine-tuning an existing deep learning model for SER to detect specific emotions allows for more targeted and accurate emotion recognition, catering to various applications in affective computing, human-computer interaction, and mental health assessment.

Chapter 5

Implementation

5.1 Top Level Overview

The script is structured to function as the entry point of a Python program that parses command-line interface (CLI) arguments to control various aspects of model training and evaluation.

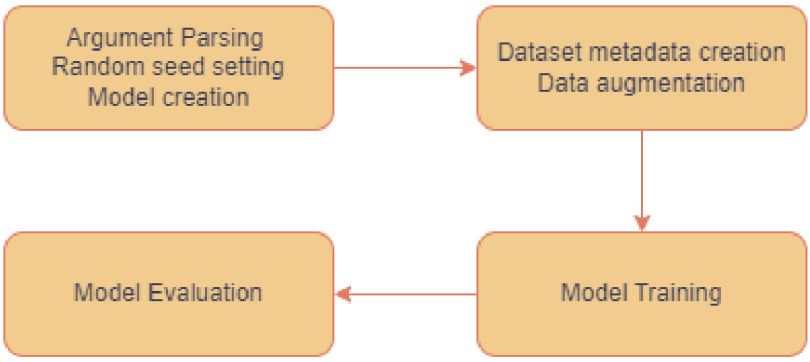


Figure 5.1: The overall flow of the program

5.1.1 Program Arguments

These are the arguments controlling the program settings:

- **-d, –dataset:** Dataset to use for training and evaluating the model. Options include:
  - `RAVDESS`
  - `EMODB`
  - `EMOVO`

  These are the datasets for which automatic metadata file creation is provided, see next argument for other datasets.
- **c, –csv:** If the dataset name is not given as an input, a custom CSV file with dataset metadata can be used.
- **-s, –seed:** Random seed for controlling the random state.

- **-b, –batch-size:** Specifies the number of samples in one training batch.
- **-e, –epochs:** Indicates the number of training epochs.
- **-lr, –learning-rate:** Sets the learning rate for training.
- **-sr, –sampling-rate:** Sampling rate for loading speech recordings.
- **-fr, –frames:** Default: Defines the number of time frames in the Mel Spectrogram.
- **-mel, –mel-filter-banks:** Specifies the size of the mel filter bank.
- **–folds:** Number of training folds for cross-validation.
- **–wandb-key:** (Required) API key for logging into your `wandb.ai` account.
- **–wandb-project:** (Required) Specifies the project in `wandb.ai` for logging experiment runs.

5.1.2 Dataset Metadata

For dataset file the following architecture was chosen

Table 5.1: Metadata File Structure for Neural Network Model

Column Name	Description	Example
recording path	File path to the audio recording	`datasets/RAVDESS/A01/03.wav`
label	Emotional state label	neutral
encoded_label	Numerical encoding of the label	0

5.1.3 Main Function

Within the `if __name__ == „__main__"` block, the script performs several key operations. First, it utilizes `argparse.ArgumentParser` to set up command-line interface (CLI) options, where users must specify the dataset, API keys for Weights & Biases (wandb), and various model parameters such as batch size and learning rate.

To ensure reproducibility across runs, the script fixes the random seed using PyTorch Lightning utilities.

Next, it calls the `train` function from the **training** module, passing user-defined parameters for dataset characteristics and training configuration.

Finally, the script evaluates the model on the specified dataset by calling the `evaluate` function from the **evaluation** module using the trained model paths.

5.2 Model training

5.2.1 Module Description

The Python class, `LigthningAST`, extends `pl.LightningModule` and manages the lifecycle of an audio spectrogram transformer model, including its training, validation, and testing phases.

Initialization

The constructor of the class takes several parameters parsed from the main module arguments defining the model architecture and training process. During initialization, the model, loss function, and various training metrics are set up.

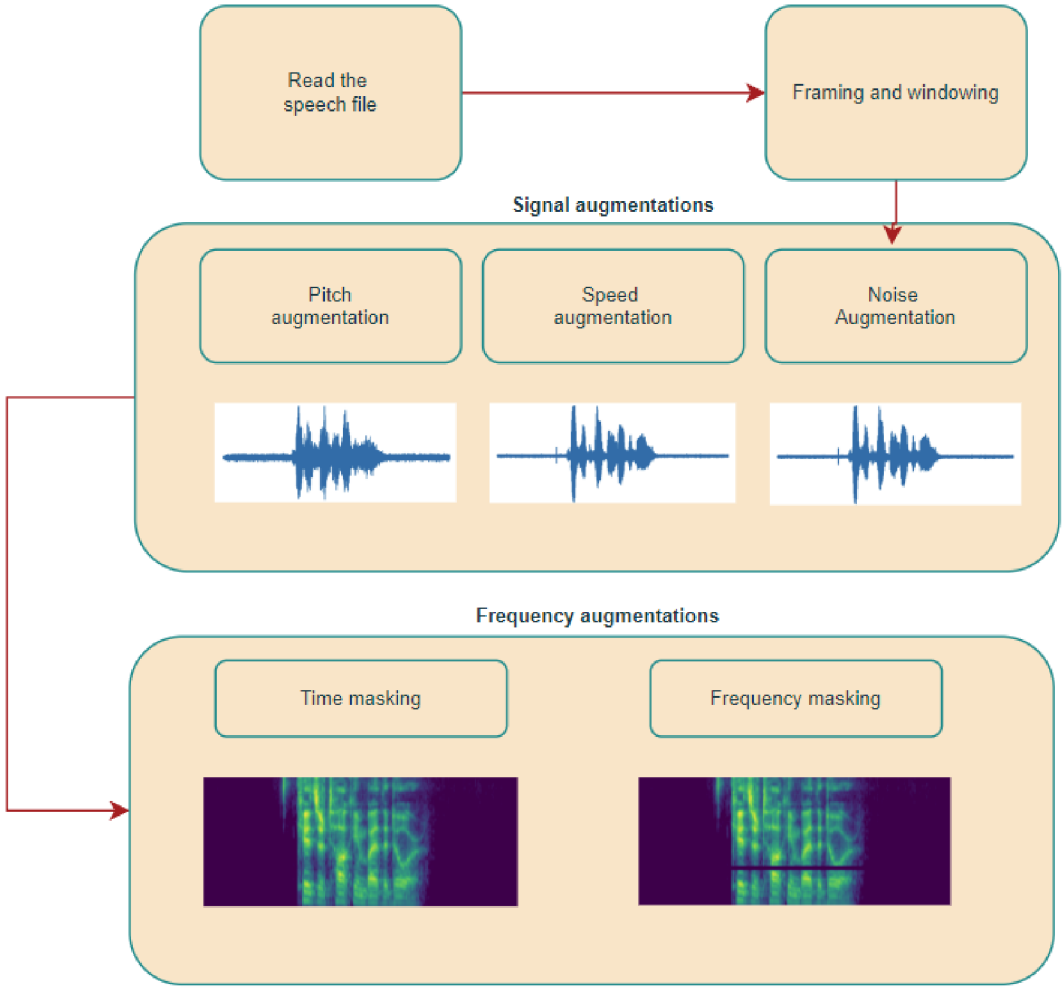
Optimizer Configuration

The `configure_optimizers` method sets up the optimization strategy:

- An Adam optimizer with a configurable learning rate.
- A cosine annealing scheduler for adjusting the learning rate across epochs.

5.2.2 Training Preparation

1. **Data Retrieval:** Metadata for the dataset is fetched, including file paths and labels, which are then split according to stratified k-folds to ensure balanced representation in each fold.
2. **Normalization:** Computes and applies normalization statistics (mean and standard deviation) for the dataset to ensure that input features are on a similar scale.
3. **Augmentation:** Employs both signal and spectrogram augmentations to enhance model robustness against variations in audio inputs.



5.2.3 Implemented Augmentations

Data augmentation is crucial for enhancing the robustness and generalization of models in machine learning, particularly in audio processing [11] [40]. This subsection discusses various techniques implemented in Python for augmenting audio data.

These augmentations are designed as subclasses of a base augmentation module, which selectively applies a specific augmentation to an audio recording based on a predefined probability. Below are detailed descriptions of each augmentation technique:

Signal augmentations

Signal augmentations are applied directly to the raw audio waveform, that is, the time-domain signal before any transformation into frequency or time-frequency representations.

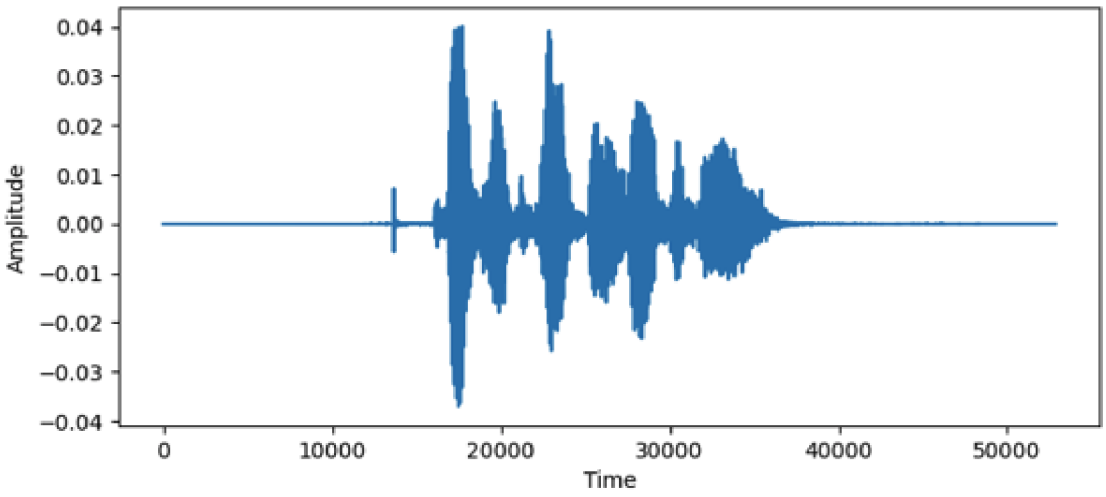


Figure 5.2: Orignal waveform

- Adding **Gaussian Noise** to audio recordings introduces non-specific background noise, which is common in real-world scenarios, thus making the model more resilient to such disturbances. [40]

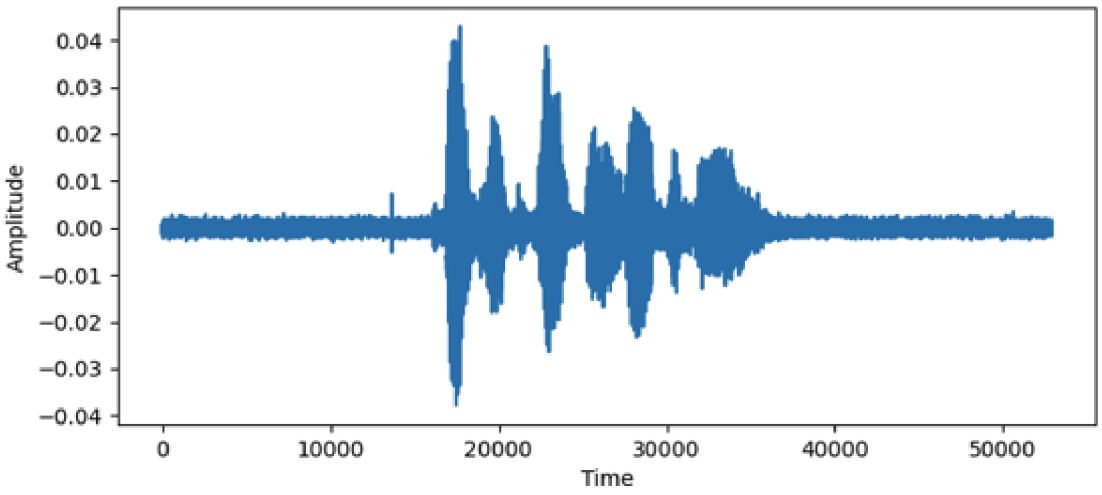


Figure 5.3: Random noise ugmentation of the signal

- **Random Speed Change** This augmentation alters the playback speed of audio recordings, affecting their temporal properties without changing the pitch. It trains the model to recognize features that are invariant to speed variations. [40]

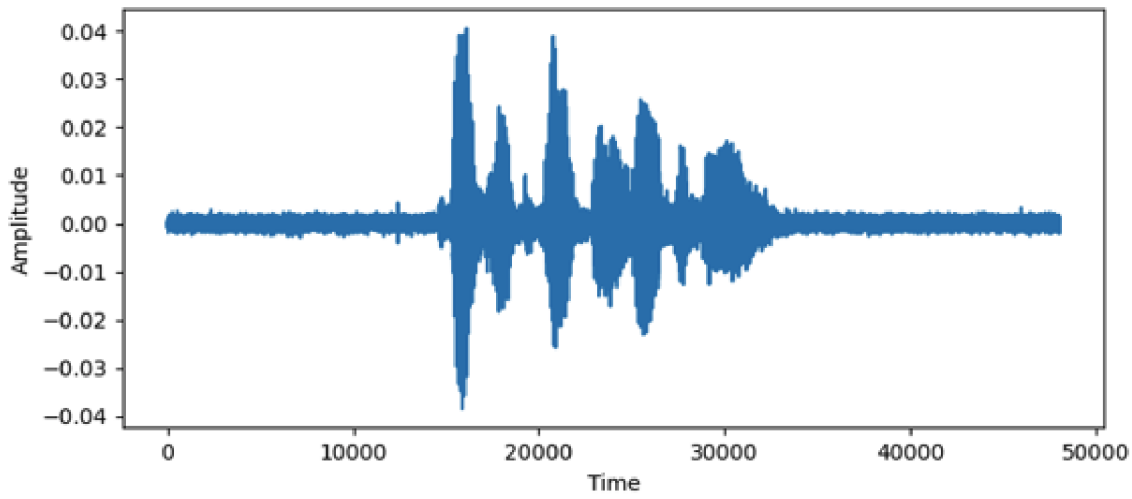


Figure 5.4: Random speed change augmentation

- **Room Impulse Response (RIR) Augmentation** simulates different acoustic environments by convolving the audio signal with a room's impulse response. This helps the model perform well across varied recording conditions. [40]

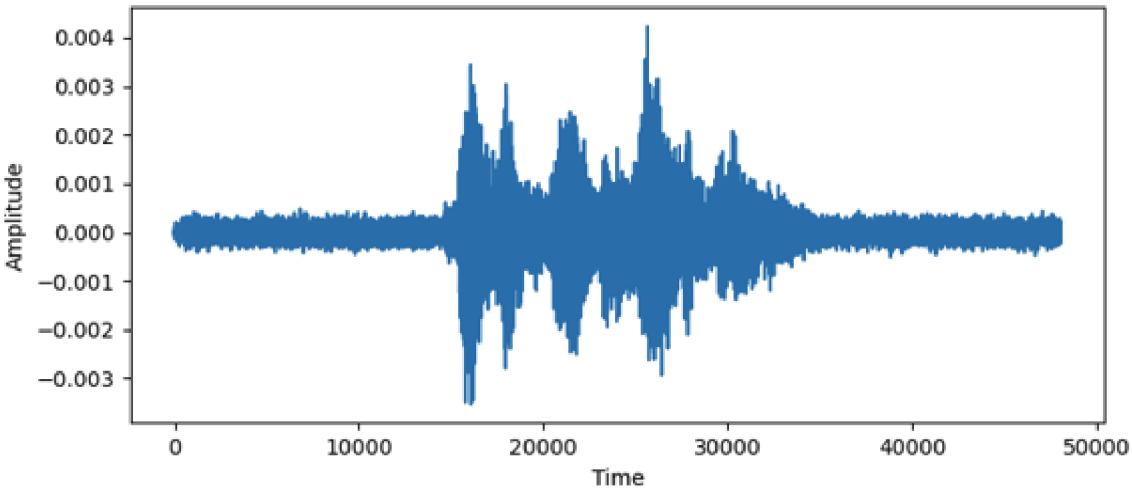


Figure 5.5: Random Impulse Response augmentation (RIR)

Spectrogram augmentations

Spectrogram augmentations are applied after the audio signal has been converted into a spectrogram, a visual representation of the spectrum of frequencies of a signal as it varies with time. [40]

- **Time Masking** randomly masks consecutive time segments in the spectrogram, similar to frequency masking but along the time axis. It challenges the model to rely on partial temporal information. [40]

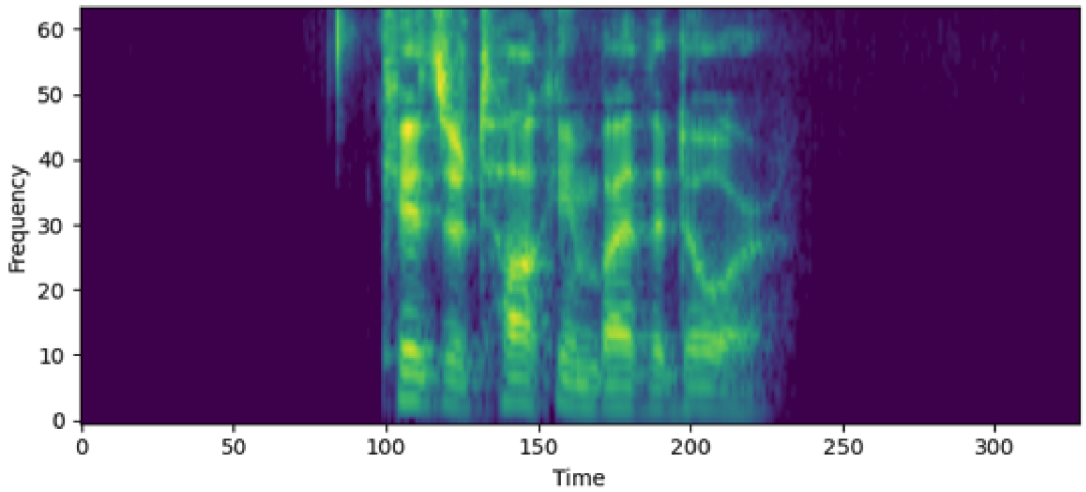


Figure 5.6: Time masking spectogram augmentation

- **Frequency Masking** is used in processing spectrograms by masking random frequency bands. This technique forces a model to learn from parts of the data where key frequency components might be missing, enhancing general robustness. [40]

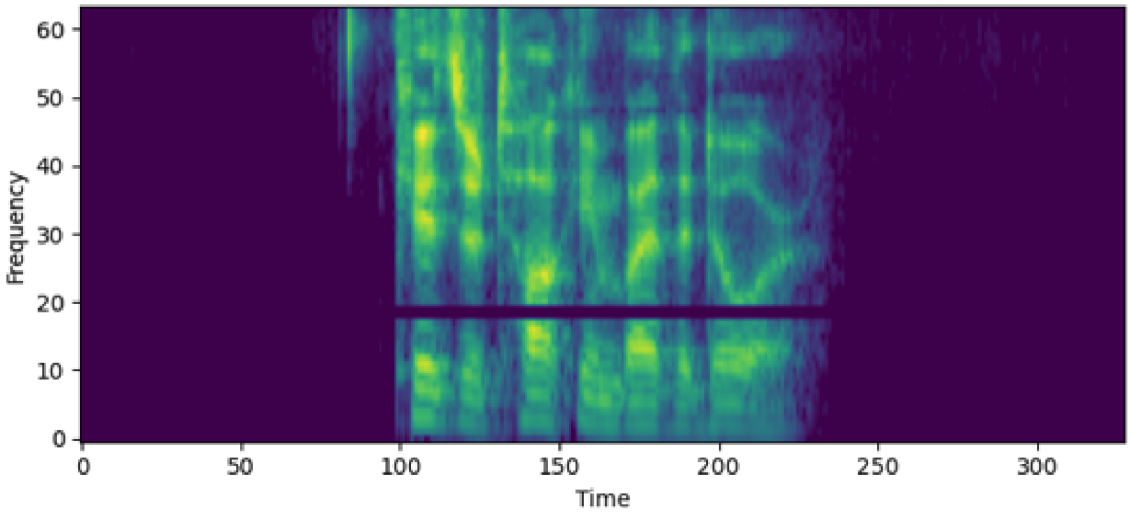


Figure 5.7: Frequency masking spectogram augmentation

Python Implementation

Here is a Python implementation Listong 5.2.3 showing the structure of the base augmentation class and how it probabilistically decides whether to apply an augmentation:

```
class BaseAugmentation(torch.nn.Module):
    def __init__(self, p, sample_rate):
        super().__init__()
        self._p = p
        self._sample_rate = sample_rate

    def forward(self, recording):
        should_apply = torch.bernoulli(torch.tensor(self._p))
        if should_apply:
            return self._apply_augumentation(recording)
        return recording

    def _apply_augumentation(self, _):
        raise NotImplementedError
```

Training and Validation Steps

The class implements specific methods for handling training and validation batches:

- `training_step`: Calculates and logs training loss and accuracy.
- `validation_step`: Calculates and logs validation loss and accuracy, storing them for later analysis.

Both methods utilize a helper function to compute predictions, loss, and accuracy from the input batch.

Visualisation, Experiment Tracking and Model Saving

Thanks to the python implementation, the structure allows for integration with the wandb [4] platform for logging training metrics and saving model checkpoints. This allows for monitoring model performance and saving the best-performing models.

The initialization happens on creation of model class, and on every n_step of model training, a log is sent into the wandb module. These „checkpoints“ are then available locally and the model training can be resumed from these partly trained models.

In the same fashion, results (accuarcy, loss function value) are logged and sent to the wandb database, where they are available to see and analyse. 5.8

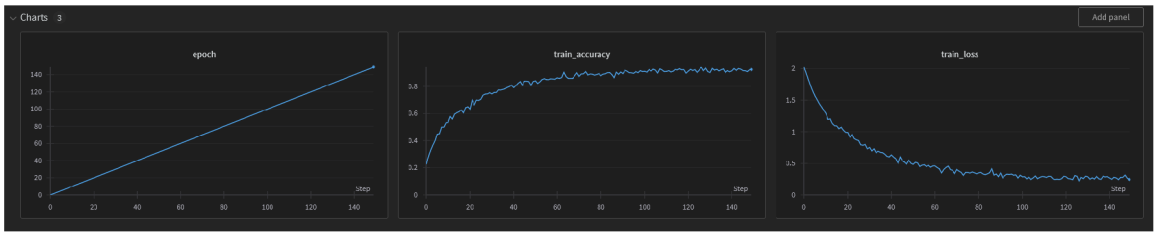


Figure 5.8: WanDB training run charts

5.3 Fine Tuning

This section details the implementation of the fine-tuning process in the Speech Emotion Recognition (SER) Python code. The primary focus is on ensuring the correctness of labels across a cross-corpus dataset and optimizing the model to improve the detection of specific emotions.

5.3.1 Dataset Label Mapping

To maintain consistency and accuracy in the labels across different datasets, a systematic label mapping approach is employed. Each dataset may have its own set of emotion labels, which can lead to discrepancies if not properly aligned. The following steps are taken to ensure correct label mapping:

1. **Standardization of Emotion Labels:** Each emotion label from the datasets is standardized to a common set of labels. For instance, labels like „happy“, „joy“, and „elation“ are all mapped to a single label „happy“.

2. **Creation of Label Index Mapping:** Once standardized, these labels are mapped to unique indices. This allows for consistent referencing across different datasets and ensures that the model correctly interprets each label during training and evaluation.

3. **Cross-Verification:** The mapping is cross-verified to ensure that each label from all datasets is correctly translated to its respective index. This step is crucial to avoid mislabeling and to maintain the integrity of the training process.

The standardized labels and their corresponding indices are then used to convert emotion labels into indices for the model weights. This ensures that the model can correctly interpret and process the emotion data from different datasets.

5.3.2 Custom Weight List for Loss Function

To enhance the detection of specific emotions, a custom weight list is implemented. This list, with a size equal to the number of emotion labels, is used to adjust the loss function, giving more importance to certain classes. The steps involved in this process are as follows:

1. **Initialization of Custom Weights:** A custom weight list is created, where each weight corresponds to a specific emotion label. For example, if the goal is to improve the detection of „anger“, the weight for the „anger“ label is increased relative to other labels.

2. **Integration with Loss Function:** The custom weight list is integrated into the loss function. During training, the loss for the „anger“ class is weighted more heavily, encouraging the model to pay extra attention to correctly classifying instances of „anger“.

3. **Model Training:** With the adjusted loss function, the model is trained on the SER dataset. The custom weights guide the model to focus more on the specified emotion, improving its recall and precision for that class.

4. **Evaluation and Adjustment:** The performance of the model is evaluated, particularly focusing on the detection of the targeted emotion. If necessary, the custom weights are adjusted iteratively to optimize performance.

5.3.3 Data Augmentation by Increasing Emotion Instances

Another method employed for fine-tuning involves feeding the training model more instances of a particular emotion. This is implemented during the dataset loading process from the CSV metadata file. The steps include:

1. **Metadata Loading:** The metadata for the dataset is loaded from a CSV file, which includes paths to the audio recordings and their corresponding emotion labels.

2. **Duplication of Target Emotion Instances:** To improve detection of a specific emotion, such as „anger“, instances of this emotion are duplicated in the dataset. This increases the representation of the target emotion in the training data.

3. **Balanced Dataset Creation:** Care is taken to ensure that the dataset remains balanced and that the increased instances of the target emotion do not lead to overfitting. Proper validation techniques are used to monitor model performance.

4. **Model Training with Augmented Data:** The augmented dataset is used to train the model, with the increased number of target emotion instances helping the model to better learn the characteristics of that emotion.

This approach of augmenting the dataset with more instances of the target emotion, in combination with the custom weight list for the loss function, allows for a robust fine-tuning process. By strategically increasing the focus on the desired emotional class, the model becomes more sensitive to its nuances, thereby enhancing its overall performance in speech emotion recognition tasks.

5.4 Computational Complexity

5.4.1 Used hardware

In the early stages, all training of the model was done locally on a machine using intel i7-8750h cpu running at 2.20GHz. For faster iteration frequency and lower waiting times, all remaining training was done remotely on the supercomputer Barbora [1] located in Ostrava, Czech Republic using the Brno University of Technology access.

Barbora's advanced computing infrastructure, equipped with high-performance GPUs and CPUs, offered the necessary computational horsepower to efficiently handle calculations and large datasets. The superior processing capabilities of the supercomputer drastically

reduced the model training times, enabling more rapid iterations and enhancements With access to greater memory and storage capacities, the model could be scaled up without compromising on the performance or accuracy.

In this section, the cost and time complexity will be analysed for future reference of training models such as the one regarded in this thesis.

5.4.2 GPU accelerators

Overview of CUDA by NVIDIA (Compute Unified Device Architecture) is a parallel computing platform and application programming interface (API) model created by NVIDIA [26]. It allows software developers to use a CUDA-enabled graphics processing unit (GPU) for general purpose processing – an approach known as GPGPU (General-Purpose computing on Graphics Processing Units).

Key Features of CUDA CUDA provides a comprehensive development environment for performing complex calculations on NVIDIA GPUs, offering several key features:

- **Parallel Computing Model:** CUDA enables developers to create algorithms that can process large blocks of data in parallel, significantly accelerating complex computations compared to sequential processing on CPUs.
- **Memory Management:** It provides various memory hierarchies and management techniques, including global, shared, constant, and texture memory, allowing for efficient data handling and optimization.
- **Direct Hardware Access:** Developers have direct access to the virtual instruction set and memory of the parallel computational elements in GPUs. This allows for higher performance and more efficient resource utilization.

Initially developed for scientific and engineering computing, CUDA has found widespread use across various domains that require intensive computational resources, such as machine learning and deep Learning where CUDA accelerates neural network training and inference, reducing the time required to train complex models.

CUDA's ability to manage and accelerate computations by leveraging the power of GPUs has made it an indispensable tool in the field of high-performance computing, enabling advancements in science, engineering, and data analysis.

5.4.3 Measuring the power used

This subsection presents a comparison of power consumption between CPU-only training and GPU training on the supercomputer Barbora. The data were collected using the Carbon library, which provides a comprehensive assessment of the environmental impact of machine learning models.

CPU-only Training

Training the model using only CPUs on the supercomputer Barbora involved the following specifications:

- **CPU:** Intel Xeon Gold 6240 CPU (72 cores)

- **RAM:** 192GB DDR4
- **Disk:** 10TB HDD (per user) with 5GB/s throughput
- **Training Time:** 6.25 hours (circa 15 sec * 150 epochs * 10 folds)
- **Energy Consumption:**
  - **CPU:** 180W (avg), 1125Wh total
  - **RAM:** 6W (avg), 37,5Wh total
  - **Disk:** 4W (avg), 25Wh total
- **Total Energy Consumption:** 1187Wh

GPU Training

Training the model on the supercomputer Barbora with an NVIDIA Tesla V100 GPU involved the following specifications:

- **GPU:** NVIDIA Tesla V100 (32GB)
- **CPU:** Intel Xeon Gold 6240 CPU (72 cores)
- **RAM:** 192GB DDR4
- **Disk:** 2TB NVMe SSD
- **Training Time:** 5 hours
- **Energy Consumption:**
  - **GPU:** 250W (avg), 1250Wh total
  - **CPU:** 120W (avg), 600Wh total
  - **RAM:** 6W (avg), 30Wh total
  - **Disk:** 4W (avg), 20Wh total
- **Total Energy Consumption:** 1900Wh

Impact of Sampling Rate on Training Time

During testing, it was observed that at a sampling rate of 16000Hz, one epoch took 15 seconds for both CPU and GPU training due to the overhead associated with the CUDA environment. This suggests that the performance advantage of GPUs may not always be fully realized at lower sampling rates.

Proposed Solution: To optimize GPU training and reduce epoch times at lower sampling rates, the following steps are recommended:

- **Batch Size Adjustment:** Increase the batch size to maximize the utilization of GPU resources. This ensures that more data is processed in parallel, reducing epoch time.
- **Data Preprocessing Optimization:** Preprocess and cache the audio data in batches to minimize the preprocessing overhead during training.

- **Asynchronous Data Loading:** Implement data loaders with multiple workers to load data asynchronously and minimize the waiting time for the next batch.
- **Mixed Precision Training:** Use mixed precision training (FP16) to reduce memory consumption and accelerate training.
- **CUDA Graphs:** Leverage CUDA Graphs to capture and replay training loops, reducing kernel launch overhead.

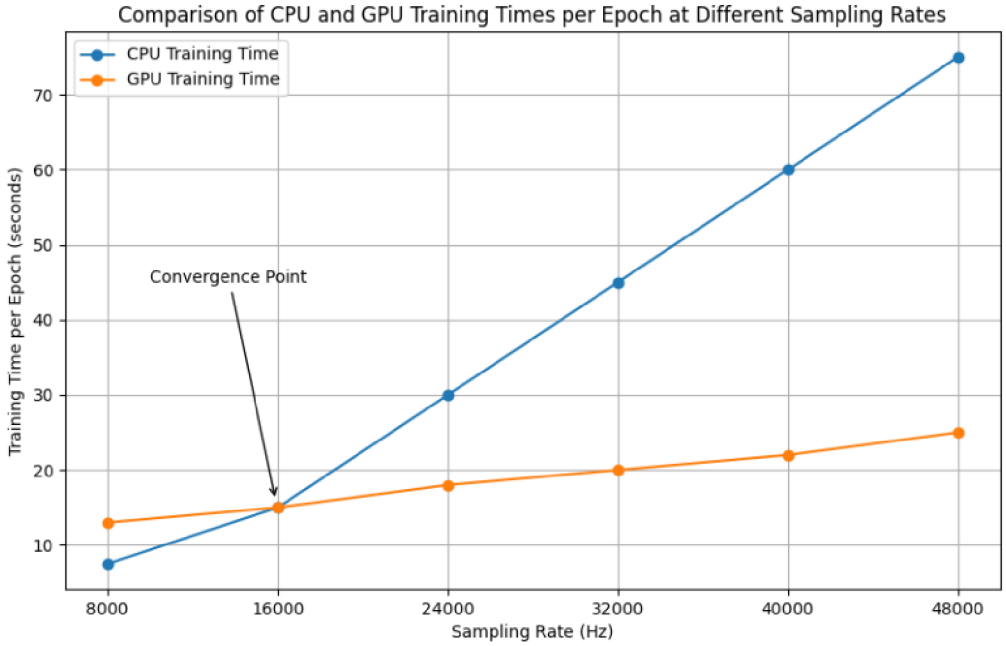


Figure 5.9: Comparison of one epoch time training on CPU vs GPU(CUDA activated)

5.4.4 Summary

As described in the subsection above, using the supercomputer or any other powerful system brings a lot of advantages for a relatively cheap price. That being said, fine tuning described in this thesis is doable on a „daily use“ computer system given some accommodations are performed.
Is it based on the specific configuration, pricing and time schedule to determine the best hardware to use for task such as this one. At the time of development of this thesis, training the models on a CPU only cluster of a supercomputer proved to be the best option.

Chapter 6

Results and Discussion

6.1 Optimizing Recall for Disease Recognition Tasks

In the context of disease recognition, ensuring minimal false negatives is crucial to avoid undetected cases, making Recall (Sensitivity) a critical metric in model evaluation. This section discusses experimental setups and strategies used to fine-tune models for high Recall performance in disease recognition tasks, particularly focusing on the emotion Anger.

6.1.1 Importance of Recall

Recall is essential in disease recognition as it measures a model's ability to identify all relevant instances of a class. For disease recognition, high Recall ensures maximum detection of true disease cases, crucial for effective diagnosis and treatment. Low Recall could lead to undetected illnesses, worsening patient outcomes.

6.1.2 Comparison of Confusion Matrices

This subsection presents confusion matrices from the evaluations of the Base Model and the Fine-Tuned Model, as shown in Figures 6.1 and 6.2, respectively. These visual representations allow for a straightforward comparison of how each model performs in classifying various emotional states.

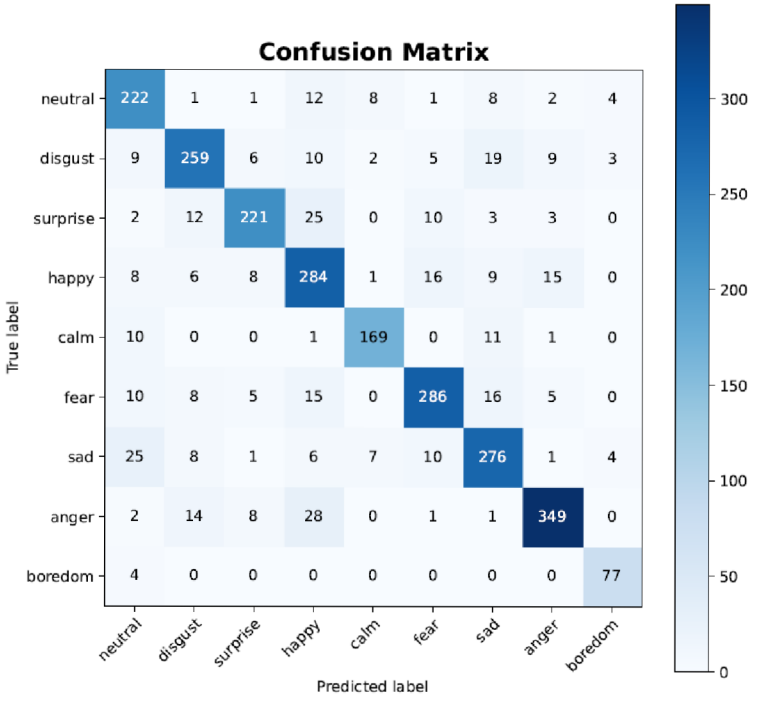


Figure 6.1: Confusion Matrix of the Base Model

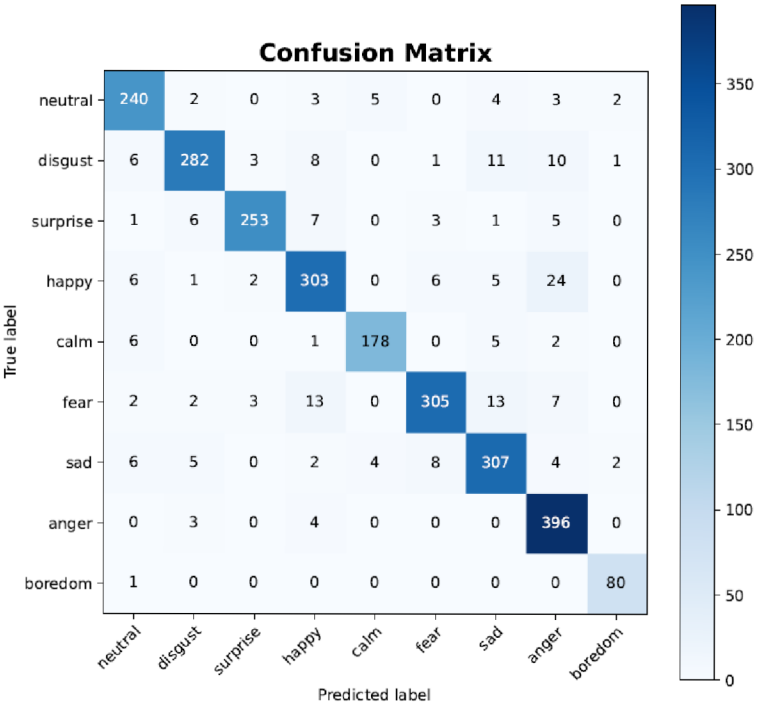


Figure 6.2: Confusion Matrix of the Fine-Tuned Model, with emphasis on Anger detection

Performance Metrics Comparison

Tables 6.1 and 6.2 display the performance metrics for the Base Model and the Fine-Tuned Model, respectively. Notably, the Recall for the emotion Anger shows marked improvement in the Fine-Tuned Model, demonstrating the effectiveness of the fine-tuning process on a cross-corpus dataset.

Table 6.1: Performance Metrics of the Base Model

Emotion	Precision	Recall	F1-score	Support	Specificity
Anger	0.759	0.854	0.804	48	0.966

Table 6.2: Performance Metrics of the Fine-Tuned Model

Emotion	Precision	Recall	F1-score	Support	Specificity
Anger	0.933	0.965	0.949	96	0.990

Analysis of Performance Across Two Models

As illustrated in Table 6.1 and Table 6.2, and further depicted in Figures 6.1 and 6.2, the Fine-Tuned Model demonstrates significant enhancements in its ability to accurately and reliably detect various emotions, particularly Anger. These improvements are critical for deploying the model in real-world applications where precise emotion recognition can lead to better outcomes in user interactions, safety protocols, and therapeutic settings.

6.2 Cross-Corpus Training and Fine-Tuning on Datasets

This section discusses the training and fine-tuning processes of our Audio Spectrogram Transformer (AST) model on a cross-corpus dataset, consisting of three distinct datasets. Initially, the model was trained on this aggregated dataset to learn general features applicable across different emotional expressions and recording conditions. Subsequently, to enhance its performance on specific data, the pre-trained model was fine-tuned directly on the dataset.

6.2.1 Testing on RAVDESS Dataset

Training Procedure

The training began with the AST model exposed to a diverse range of emotional states and acoustic environments presented by the combined datasets. This initial phase aimed at equipping the model with robust, generalized capabilities for emotion recognition.

Fine-Tuning Process

After the initial training, the model underwent a fine-tuning process on the RAVDESS dataset. Fine-tuning adjusted the model's weights specifically to the acoustic and emotional characteristics present in RAVDESS, thus optimizing its performance for this particular set.

Convergence and Performance

The fine-tuning phase was notably efficient, with the model converging to optimal performance in less than 10 epochs. This rapid convergence highlights the effectiveness of leveraging a pre-trained model that has already captured a broad understanding of emotional cues, requiring only minor adjustments to specialize for a particular dataset.

Results Visualization

To illustrate the improvement brought by the fine-tuning process, confusion matrices before and after fine-tuning on the RAVDESS dataset are presented. These matrices provide a visual representation of the model's performance on classifying different emotional states.

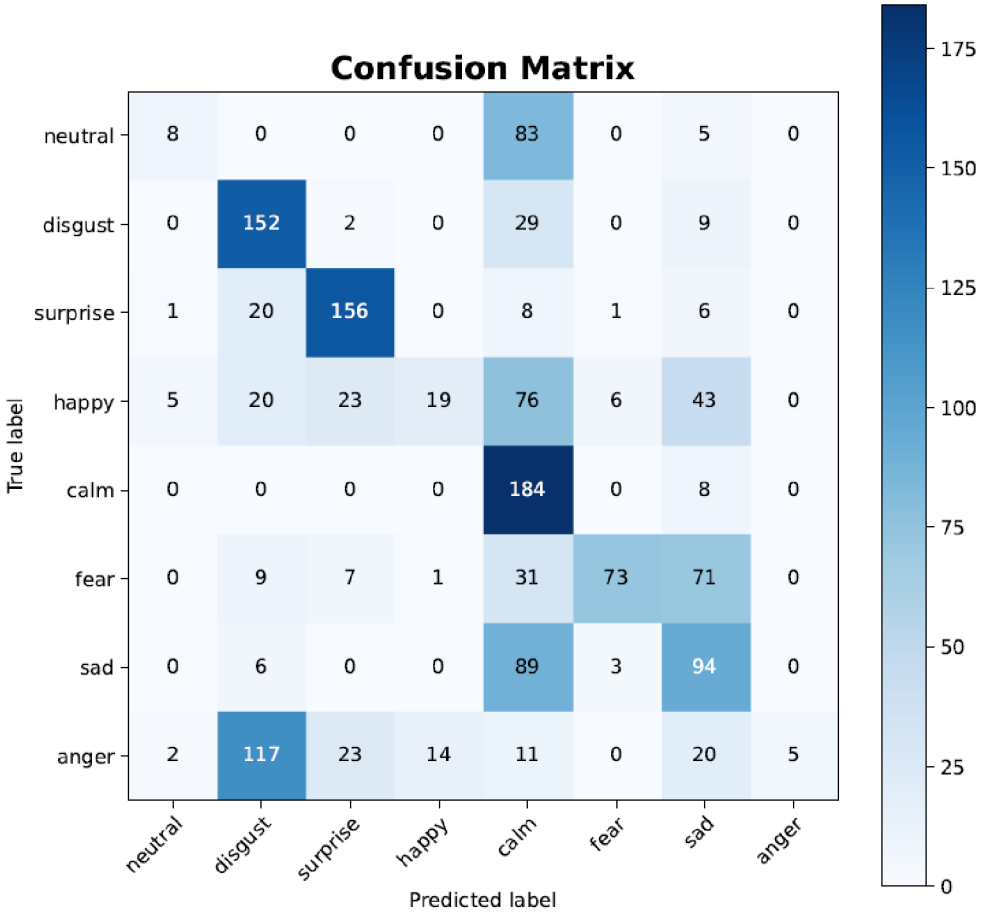


Figure 6.3: Confusion matrix of the cross-corpus model tested on the RAVDESS dataset

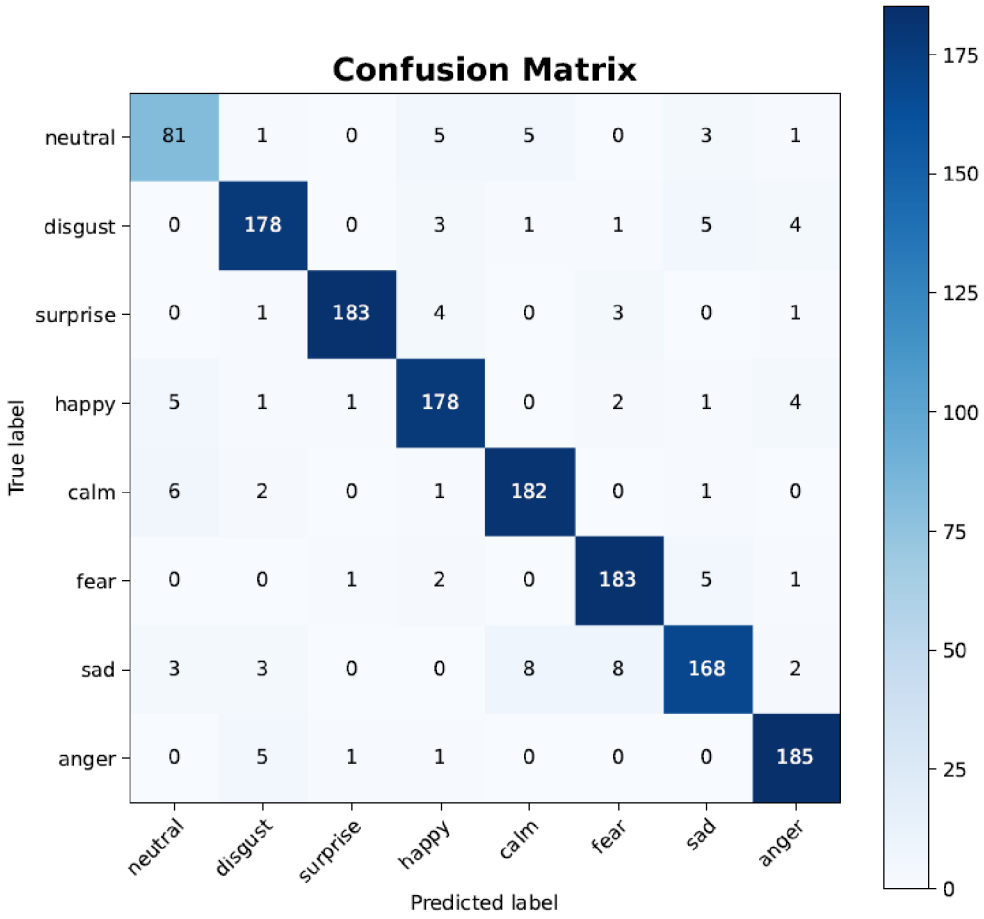


Figure 6.4: Confusion matrix of the model after fine-tuning on the RAVDESS dataset

6.2.2 Testing on EMODB Dataset

Following the successful fine-tuning on RAVDESS, the model was also tested on the EMODB dataset, which only includes 7 of the 9 emotions that the model was initially trained to recognize, specifically lacking 'surprise' and 'calm'. Despite this limitation, the results were promising and demonstrate the model's adaptability and functionality with minimal additional computational investment for training.

Performance Analysis

The confusion matrices for the EMODB dataset illustrates how the model managed to adjust its predictions in the absence of 'surprise' and 'calm'. This analysis helps in understanding the model's capability to handle datasets with varied emotional labels effectively.

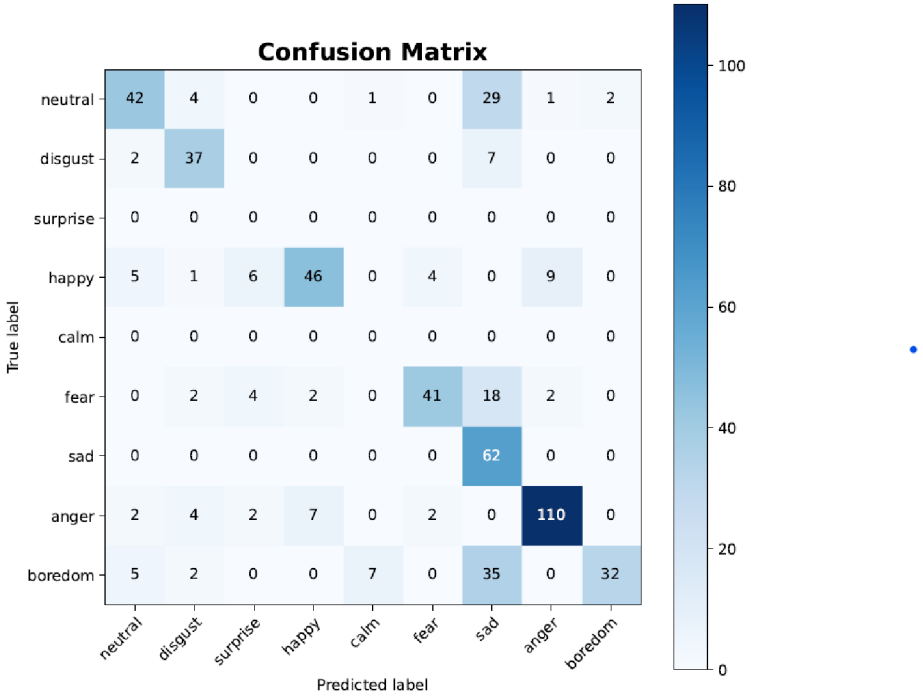


Figure 6.5: Confusion matrix of the model tested on the EMODB dataset

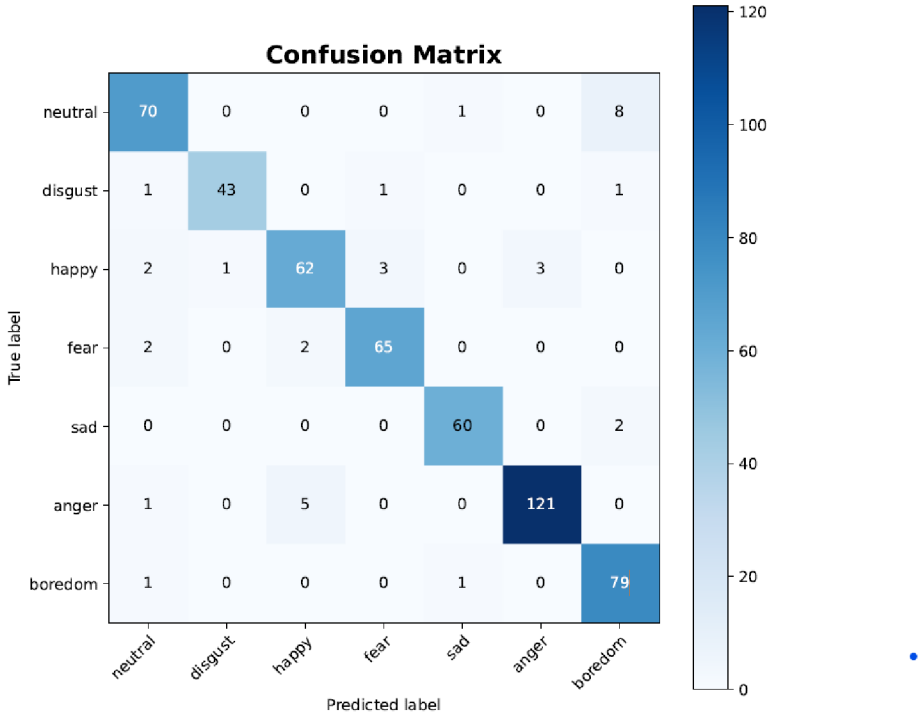


Figure 6.6: Confusion matrix of the fine-tuned model tested on the EMODB dataset

This testing phase underscores the model's flexibility and robustness, proving its efficacy even when the training and testing datasets do not perfectly align in terms of emotion categories. The confusion matrices (Figures 6.3, 6.4, and 6.5) visually represent the model's performance across different datasets, highlighting its strengths and areas for potential improvement.

6.2.3 Testing on EMOVO Dataset

The model was further evaluated on the EMOVO dataset to assess its adaptability and performance across a diverse set of emotional expressions. EMOVO provides a distinct context due to its unique composition of Italian emotional speech, which challenges the model to demonstrate its robustness and generalization capabilities.

Dataset Challenges

The EMOVO dataset presents a different set of emotional expressions, some of which were not as prominently featured in the training datasets. This variation tests the model's ability to generalize learned emotional cues to new, context-specific scenarios.

Results Analysis

To visually represent the model's performance on EMOVO, confusion matrices before and after any additional tuning or retraining are provided. These matrices help illustrate the initial adaptability of the model to EMOVO and the improvements in classification accuracy after fine-tuning.

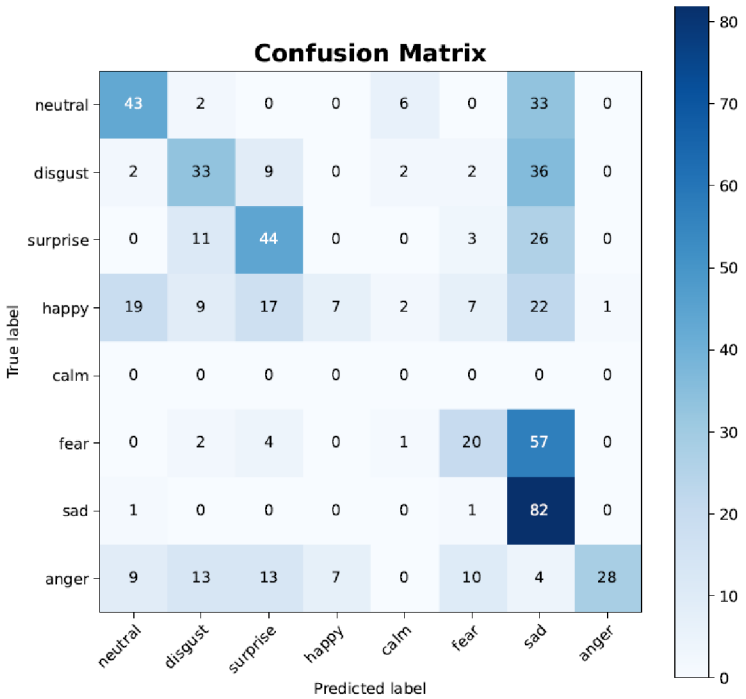


Figure 6.7: Initial confusion matrix of the model tested on the EMOVO dataset.

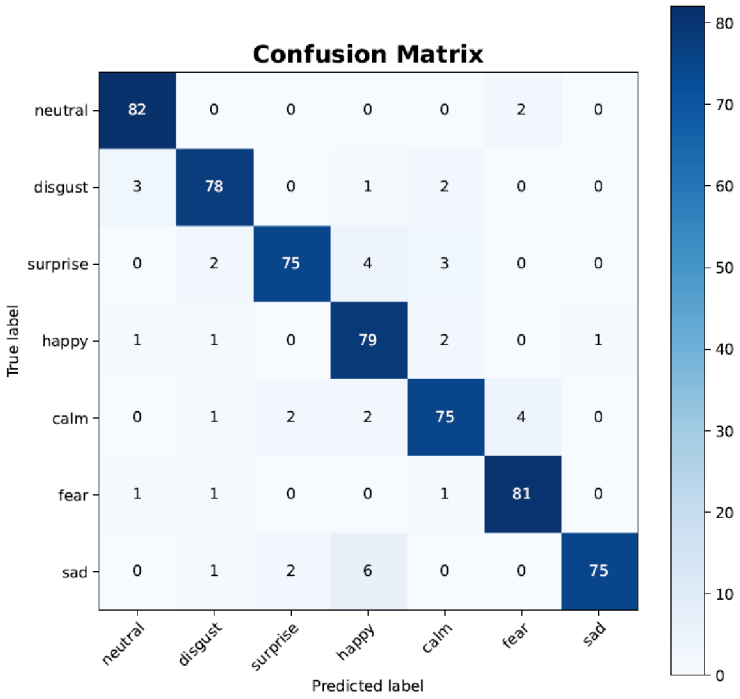


Figure 6.8: Confusion matrix of the model after additional tuning on the EMOVO dataset.

The visual data from Figures 6.7 and 6.8 confirm the model's capability to adapt to the EMOVO dataset's characteristics and highlight the effectiveness of fine-tuning in achieving better specificity and overall accuracy. This testing phase not only showcases the model's flexibility but also its efficiency in adapting to datasets with different linguistic and emotional compositions.

6.3 Known issues and complications

Apart from the usual problems connected to software development, some other issues came up that should be addressed in this section.

- GPU cluster time - during the training of a model checkpoint, the GPU allocated time for the university project ran out, which later complicated the calculation of time complexity. Older logs were used to calulate the necessary metrics.

6.4 Future work

Based on the problem on described at 5.4.3, the proposed solution could be implemented and a benchmark created to find out the best setting for fine tuning models such as the one in this thesis, making the task of specific emotion recognition more available, even for people with less or no experience with neural networks.

Chapter 7

Conclusion

Speech Emotion Recognition (SER) is an inherently challenging task due to the complexity and variability of human emotions, the diverse expression of these emotions across different speakers, and the influence of various contextual and environmental factors. This thesis set out to address this challenge by developing a fine-tuned solution specifically tailored for SER tasks, demonstrating that customized models can yield better results than broadly generalized approaches, the resulting values of weighted accuracy are as follows: 93.5% for the EMODB dataset, 92.8% for EMOVO, and 92.9% for the RAVDESS dataset.

The primary goal of this thesis was to provide a solution for fine-tuning SER models for specific tasks. By leveraging transfer learning techniques and pre-trained models, the work presented here has shown that customizing models to suit a particular dataset or application can enhance performance compared to generalized models. Fine-tuning allows the model to capture the specific nuances and characteristics of the target dataset, ultimately leading to improved emotion classification accuracy.

By providing clean, reusable code in Python and extensive documentation of the source files enriched by guide-style readme files, the secondary goal of this thesis was to provide a guideline for fine tuning a model for a specific task in a way that is obtainable without extensive technical research, thus laying grounds for more development in the field of software emotion recognition.

Bibliography

[1] *Barbora Supercomputer* [https://www.it4i.cz/en/infrastructure/barbora]. IT4Innovations, National Supercomputing Center, 2024. Accessed: 2024-05-07.

[2] Akçay, M. B. and Oğuz, K. Speech Emotion Recognition: Emotional Models, Databases, Features, Preprocessing Methods, Supporting Modalities, and Classifiers. *Speech Communication*. 2020, vol. 116, p. 56–76. DOI: 10.1016/j.specom.2019.12.001. ISSN 0167-6393. Available at: https://www.sciencedirect.com/science/article/pii/S0167639319302262.

[3] Barrett, L. F. Discrete emotions or dimensions? The role of valence focus and arousal focus. *Cognition and Emotion*. 1998, vol. 12, no. 4, p. 579–599.

[4] Biewald, L. *Experiment Tracking with Weights and Biases*. 2020. Software available from wandb.com. Available at: https://www.wandb.com/.

[5] Burkhardt, F., Paeschke, A., Rolfes, M., Sendlmeier, W. F. and Weiss, B. A Database of German Emotional Speech. *Proceedings of the 9th European Conference on Speech Communication and Technology (Interspeech)*. 2005, p. 1517–1520. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.61.9375&rep=rep1&type=pdf.

[6] Busso, C., Bulut, M., Lee, C.-C., Kazemzadeh, A., Mower, E. et al. IEMOCAP: Interactive Emotional Dyadic Motion Capture Database. In: European Language Resources Association (ELRA). *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC)*. Marrakech, Morocco: [b.n.], 2008, p. 335–338. Available at: http://sail.usc.edu/publications/files/bussoLREC2008.pdf.

[7] Busso, C., Lee, S. and Narayanan, S. Analysis of Emotionally Salient Aspects of Fundamental Frequency for Emotion Detection. *Audio, Speech, and Language Processing, IEEE Transactions on*. june 2009, vol. 17, p. 582 – 596. DOI: 10.1109/TASL.2008.2009578.

[8] Cannon, W. B. The James-Lange Theory of Emotions: A Critical Examination and an Alternative Theory. *The American Journal of Psychology*. 1927, vol. 39, 1/4, p. 106–124.

[9] Darwin, C. The expression of the emotions in man and animals. *John Murray, London*. 1872. Available at: http://dx.doi.org/10.1037/10001-000.

[10] Davis, S. B. and Mermelstein, P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*. 1980, vol. 28, no. 4, p. 357–366.

[11] DeVries, T. and Taylor, G. W. Improved regularization of convolutional neural networks with cutout. *ArXiv preprint arXiv:1708.04552*. 2017.

[12] Flanagan, J. L. *Speech Analysis Synthesis and Perception*. Springer-Verlag, 1972.

[13] Gong, Y., Chung, Y.-A. and Glass, J. PSLA: Improving Audio Tagging with Pretraining, Sampling, Labeling, and Aggregation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. 2021. DOI: 10.1109/TASLP.2021.3120633.

[14] Han, K., Yu, D. and Tashev, I. Speech Emotion Recognition Using Deep Neural Network and Extreme Learning Machine. In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. 2014. DOI: 10.21437/Interspeech.2014-57.

[15] Harris, F. J. On the use of windows for harmonic analysis with the discrete Fourier transform. *Proceedings of the IEEE*. 1978, vol. 66, no. 1, p. 51–83.

[16] Jahangir, R., Teh, Y. W., Hanif, F. and Mujtaba, G. Deep learning approaches for speech emotion recognition: state of the art and research challenges. *Multimedia Tools and Applications*. july 2021, vol. 80, no. 16, p. 23745–23812.

[17] James, W. What Is an Emotion? *Mind*. 1884, vol. 9, no. 34. DOI: 10.1093/mind/os-IX.34.188. Available at: http://dx.doi.org/10.1093/mind/os-IX.34.188.

[18] Kasuya, H., Ogawa, H., Yoshida, K. and Kikuchi, J. Mutual relationships among speaking fundamental frequency, voice perturbations, and glottal airflow characteristics. *Journal of Speech and Hearing Research*. 1986, vol. 29, p. 149–157.

[19] Koolagudi, S., Vempada, R. and Rao, K. Emotion recognition from speech signal using epoch parameters. In:. July 2010. DOI: 10.1109/SPCOM.2010.5560541.

[20] Kumaran, U., Radha Rammohan, S. and Nagarajan, S. Fusion of Mel and Gammatone Frequency Cepstral Coefficients for Speech Emotion Recognition Using Deep C-RNN. *International Journal of Speech Technology*. 2021, vol. 24, p. 303–314. DOI: 10.1007/s10772-020-09792-x.

[21] Lippmann, R. Neural network classifiers for speech recognition. *Lincoln Laboratory Journal*. january 1988, vol. 1.

[22] Livingstone, S. R. and Russo, F. A. The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS). *PLoS ONE*. Public Library of Science. 2018, vol. 13, no. 5, p. e0196391. DOI: 10.1371/journal.pone.0196391. Available at: https://doi.org/10.1371/journal.pone.0196391.

[23] Lotfian, R. and Busso, C. Building a Naturalistic Emotion Database from Online Media: The MSP-Podcast Corpus. In: IEEE. *Proceedings of the 9th International Conference on Affective Computing and Intelligent Interaction (ACII)*. 2019, p. 1–7.

[24] Lövheim, H. A new three-dimensional model for emotions and monoamine neurotransmitters. *Medical Hypotheses*. 2012, vol. 78, no. 2, p. 341–348.

[25] Marsella, S. and Gratch, J. EMA: A process model of appraisal dynamics. *Cognitive Systems Research*. 2009, vol. 10, no. 1, p. 70–90.

[26] NVIDIA, Vingelmann, P. and Fitzek, F. H. *CUDA, release: 10.2.89*. 2020. Available at: https://developer.nvidia.com/cuda-toolkit.

[27] Oppenheim, A. V., Willsky, A. S. and Nawab, S. H. *Signals and Systems*. 2ndth ed. Prentice-Hall, 1996. ISBN 978-0138147570.

[28] Plato and Sachs, J. *Republic*. Focus Pub., 2007. Focus philosophical library. ISBN 9781585102617. Available at: https://books.google.sk/books?id=cZz9GQAACAAJ.

[29] Plutchik, R. *Emotions in the practice of psychotherapy: Clinical implications of affect theories*. American Psychological Association, 2000. Available at: https://doi.org/10.1037/10366-000.

[30] Powers, D. M. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation. *Journal of Machine Learning Technologies*. DMW & Associates. 2011, vol. 2, no. 1, p. 37–63.

[31] Prima, I. and Ahmad, D. ANALISIS CONDITIONAL RESTRICTED BOLTZMAN MACHINE UNTUK MEMPREDIKSI HARGA SAHAM BANK SYARIAH INDONESIA. *Jurnal Lebesgue : Jurnal Ilmiah Pendidikan Matematika, Matematika dan Statistika*. april 2023, vol. 4, p. 409–416. DOI: 10.46306/lb.v4i1.266.

[32] Rabiner, L. R. and Schafer, R. W. *Digital Processing of Speech Signals*. Prentice Hall, 1978.

[33] Roberts, W. and Ross, W. *Rhetoric*. Cosimo, Incorporated, 2010. Cosimo Classics Philosophy. ISBN 9781616403089. Available at: https://books.google.sk/books?id=zDGG2e8e0JsC.

[34] Russell, J. A Circumplex Model of Affect. *Journal of Personality and Social Psychology*. december 1980, vol. 39, p. 1161–1178. DOI: 10.1037/h0077714.

[35] Schachter, S. and Singer, J. E. Cognitive, Social, and Physiological Determinants of Emotional State. *Psychological Review*. 1962, vol. 69, no. 5, p. 379–399.

[36] Shah, K., Shah, K., Chaudhari, A. and Kothadiya, D. Comprehensive Analysis of Deep Learning Models for Brain Tumor Detection from Medical Imaging. In:. February 2024, p. 339–351. DOI: 10.1007/978-981-99-7820-5_28. ISBN 978-981-99-7819-9.

[37] Surdeanu, M. and Valenzuela Escárcega, M. Recurrent Neural Networks. In:. February 2024, p. 147–164. DOI: 10.1017/9781009026222.011.

[38] Walinga, J. *Introduction to Psychology: 1st Canadian Edition*. BCcampus, 2010. Online access: Center for Open Education Open Textbook Library. Available at: https://books.google.sk/books?id=jyGPzQEACAAJ.

[39] Wang, X., Wang, M., Qi, W., Su, W. and Wang, X. A Novel End-to-End Speech Emotion Recognition Network with Stacked Transformer Layers. In: *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2021, p. 6289–6293. DOI: 10.1109/ICASSP39728.2021.9414314.

[40] Wei, S., Zou, S., Liao, F. and Lang, W. A Comparison on Data Augmentation Methods Based on Deep Learning for Audio Classification. *Journal of Physics: Conference Series*. january 2020, vol. 1453, p. 012085. DOI: 10.1088/1742-6596/1453/1/012085.

[41] Wu, A., Huang, Y. and Zhang, G. Feature Fusion Methods for Robust Speech Emotion Recognition Based on Deep Belief Networks. In:. December 2016, p. 6–10. DOI: 10.1145/3033288.3033295.

[42] Ye, J., Wen, X., Wei, Y., Xu, Y. and Liu, K. Temporal Modeling Matters: A Novel Temporal Emotional Modeling Approach for Speech Emotion Recognition. *ArXiv*. 2022. DOI: 10.48550/ARXIV.2211.08233. Available at: https://arxiv.org/abs/2211.08233.

[43] Zagalo, N., Torres, A. and Branco, V. Emotional Spectrum Developed by Virtual Storytelling. In:. November 2005, p. 105–114. DOI: 10.1007/11590361_12. ISBN 978-3-540-30511-8.

[44] Zhang, X., Zhang, X. and Wang, W. Convolutional Neural Network. In:. October 2023, p. 39–71. DOI: 10.1007/978-981-99-6449-9_2. ISBN 978-981-99-6448-2.

[45] Zhao, Z., Li, Q., Zhang, Z., Cummins, N. and Wang, H. Combining a parallel 2D CNN with a self-attention Dilated Residual Network for CTC-based discrete speech emotion recognition. *Neural Networks*. 2021, vol. 141, p. 52–60. DOI: 10.1016/j.neunet.2021.03.013. ISSN 0893-6080. Available at: https://www.sciencedirect.com/science/article/pii/S0893608021000939.

A SD card content **65**

Appendix A

SD card content

The attached memory card has the following structure:

- Thesis.pdf - This PDF with thesis text.
- src/ - Folder containing the implementation of neural network
- thesis_source/ - Latex source codes for PDF generation
- README.MD - A readme file for this folder