Ö.Ç. DALA

COMPARISON OF DEEP LEARNING MODELS FOR SPEECH EMOTION RECOGNITION

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

ÖMER ÇAĞRI DALA

A MASTER OF SCIENCE THESIS
IN
THE DEPARTMENT OF COMPUTER ENGINEERING

MAY 2024

ATILIM UNIVERSITY 2024

COMPARISON OF DEEP LEARNING MODELS FOR SPEECH EMOTION RECOGNITION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

BY

ÖMER ÇAĞRI DALA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

MAY 2024

Approval of the Graduate School of Natural and Applied Sciences, Atilim University.

Prof. Dr. Ender Keskinkılıç
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of **Master of Science in Computer Engineering Department, Atilim University**.

Prof. Dr. Gökhan Şengül
Head of Department

This is to certify that we have read the thesis COMPARISON OF DEEP LEARNING MODELS FOR SPEECH EMOTION RECOGNITION submitted by ÖMER ÇAĞRI DALA and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Murat KOYUNCU
Supervisor

Examining Committee Members:

Prof. Dr. Tansel DÖKEROĞLU
Software Eng. Dept., TED University

Prof. Dr. Murat KOYUNCU
Inform. Systems Eng. Dept., Atılım University

Prof. Dr. Ahmet COŞAR
Comp. Eng. Dept., Ankara Medipol University

Date: May 19, 2024

I declare and guarantee that all data, knowledge and information in this document has been obtained, processed and presented in accordance with academic rules and ethical conduct. Based on these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : ÖMER ÇAĞRI DALA

Signature :

ABSTRACT

COMPARISON OF DEEP LEARNING MODELS FOR SPEECH EMOTION RECOGNITION

Dala, Ömer Çağrı

M.S., Department of Computer Engineering

Supervisor : Prof. Dr. Murat KOYUNCU

May 2024, 67 pages

This thesis focuses on the practical applications of deep learning models on the Speech Emotion Recognition (SER) problem. By combining Convolutional Neural Networks (CNNs) and intermediate layers, the study aims to extract emotional features with contextual awareness from speech signals. The proposed approach automatically learns effective representations of emotional content in speech, addressing the weaknesses of traditional techniques. Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) is used for the SER problem, focusing on techniques to improve accuracy. Deep learning models such as Convolutional Neural Networks (CNNs) are commonly used in SER tasks, alongside traditional machine learning algorithms such as Decision Trees, Adaboost, and Random Forest. Deep Learning models are designed such as local and global emotion-related features are automatically learned from speech and log-mel spectrograms which capture both temporal and spectral information in a compact form, making them suitable input representations for deep learning models. This study demonstrates achievable advances in emotion recognition technology by enhancing the recognition of emotions from speech data through the extraction of deep emotion features. The experimental results of this thesis show that CNN models give very satisfactory results on the SER problem.

Keywords: Speech emotion recognition, Deep learning, Signal processing

ÖZ

KONUŞMA DUYGUSU TANIMA İÇİN DERİN ÖĞRENME MODELLERİNİN KARŞILAŞTIRILMASI

Dala, Ömer Çağrı

Yüksek Lisans, Bilgisayar Mühendisliği

Tez Yöneticisi : Prof. Dr. Murat Koyuncu

Mayıs 2024, 67 sayfa

Bu tez, derin öğrenme modellerinin Konuşma Duygusu Tanıma (SER) problemi üzerindeki pratik uygulamalarına odaklanmaktadır. Evrişimsel Sinir Ağları (CNN'ler) ve ara katmanları birleştirerek, konuşma sinyallerinden bağlamsal farkındalık ile duygusal özelliklerin çıkarılması amaçlanır. Önerilen yaklaşım, geleneksel tekniklerle ilgili zorlukları ele alarak, konuşmadaki duygusal içeriğin etkili temsillerini otomatik olarak öğrenmeye odaklanmaktadır. Ryerson Duygusal Konuşma ve Şarkının Görsel-İşitsel Veritabanı (RAVDESS), doğruluğu artırmaya yönelik tekniklere odaklanan SER probleminde kullanılır. Evrişimli Sinir Ağları (CNN'ler) gibi derin öğrenme modelleri, Karar Ağaçları, Adaboost ve Rastgele Orman gibi geleneksel makine öğrenme algoritmalarının yanısıra SER görevlerinde yaygın olarak kullanılır. Derin Öğrenme mimarisi, yerel ve küresel duygu özelliklerinin, hem zamansal hem de spektral bilgileri kompakt bir biçimde yakalayan konuşma ve log-mel spektrogramlarından otomatik öğrenilmesi ve bunları derin öğrenme modelleri için uygun giriş temsiline getirmesi şeklinde tasarlanmıştır. Bu çalışma, derin duygu özelliklerinin çıkarılması yoluyla konuşma verilerinden duyguların tanınmasını geliştirerek duygu tanıma teknolojisinde ilerleme göstermeyi amaçlamaktadır. Bu tez ile elde edilen deneysel sonuçlar, CNN modellerinin SER problemi üzerinde tatmin edici sonuçlar verdiğini göstermektedir.

Anahtar Kelimeler: Konuşmada duygu tanıma, Derin öğrenme, Sinyal işleme

To my parents

ACKNOWLEDGMENTS

I would like to express my thanks to my thesis advisor Prof. Murat Koyuncu, and committee members for their valuable comments and directions for improvement of the thesis content.

I also would like to thank my friends in Ankara Science University and academic supervisors for their guidance on machine learning models and help with Python speech processing libraries and feature extraction, especially, Dr. Ender Sevinç for his guidance.

Finally, I would like to thank my parents, Nilgun Dala and Ahmet Dala for their endless support and love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Human emotion recognition can be performed on a number of input multi-media types such as videos, captured images, signals collected through brain wave signal measurements, etc [1, 2]. These data can be collected from sources indicating a human being's facial or vocal expressions. In addition to these, complex information sources like signals fired up inside a human subject's brain, and a human subject's muscular tension measurements require extensive sensor setup preparations (electroencephalogram (EEG) machine or EEG device) or Functional Magnetic Resonance Imaging (fMRI) [3] to collect data, can be used. In this thesis, we concentrate on speech data because it can be collected easily and used to support and implement many practical applications.

Human emotion can be identified by many outer physical attributes such as physical responses, body gestures or invisible indicators such as heart rate and body temperature. It can also be detected without physical contact such as speech recorded by using a microphone. In this thesis, we concentrate on recorded human speech, because it is very easy to collect and can be recorded in a compact format digitally. Another advantage is the usability of such an emotion recognizer over a remote telephone conversation where recognizing the emotional state of the other party could be used to enhance user satisfaction and the success of the underlying application (for example, a tutoring application can recognize that the subject is bored and takes appropriate

measures such as presenting the subject a multiple choice question to attract his/her attention).

Speech carries both paralinguistic and linguistic information [4]. The linguistic information includes the context and language of the speech, while paralinguistic information includes the gender, emotional state, age, accent and other unique attributes of the human.

Prosody encoded in the form of intonation, rhythm, and lexical stress patterns of spoken language conveys linguistic and paralinguistic information such as emphasis, intent, attitude, and emotion of a speaker. Prosody in spoken language correlates with acoustic and syntactic features. Acoustic correlates of duration, intensity and pitch are some of the acoustic features that are used to express prosodic prominence or stress in English. Lexical and syntactic features such as parts-of-speech, syllable nuclei identity, and syllable stress of neighboring words have also been shown to exhibit a high degree of correlation with prominence. Speech features such as intensity, intonation, pitch which are prosodic features are extracted from human audio sources. In this thesis mel-frequency cepstral coefficients [5], mel-spectrogram [6], Tonnetz (short for Tonal Centroid Features, is based on the concept of tonal harmony in music theory) [7], short time Fourier transform (STFT) [8] and chromagram [9] are included in the feature vector used by the proposed machine learning model.

The temporal features (time domain features) are simpler to extract and have easy physical interpretations, like: the energy of the signal, zero crossing rate, maximum amplitude, minimum energy, etc.

The spectral features (frequency-based features) are generally obtained by convert-

ing temporal signals into the frequency domain using the Fourier Transform. These features can be used to identify the notes, pitch, rhythm, and melody. Spectral features are obtained by a deeper, and computationally more expensive, processing of audio signals (including both speech and music signals).

In the Literature Review performed for this thesis, the focused topics were: (1) Machine Learning techniques for human Speech Emotion Recognition (SER) (2) determining features (average pitch, standard deviation, skewness, etc) which are widely used for speech analysis (3) available and proven standard software libraries for speech processing (4) techniques for increasing training accuracy in machine learning models.

Efforts in recognition of emotions through human vocal output, collectively known as SER [10] is the process of predicting human emotions from audio signals using artificial intelligence (AI) techniques. SER technologies have a wide range of applications in areas such as education, entertainment, call centres, Human-Computer Interaction (HCI), automatic translation systems, vehicle automation, and healthcare. Extracting relevant features from audio signals is a crucial task in the SER process to correctly identify emotions. This has led to a growing interest in the field of SER, which involves identifying the emotions of speakers from their voices.

An important conclusion of the literature survey is that some speech attributes point to general the characteristics of emotion, rather than being directly involved with individual categories. For example, Anger, Fear, Joy and, to a certain extent, Surprise has positive activation (approach) and hence have similar characteristics such as a much higher average of $F0$ values and a much wider $F0$ range. On the other hand, emotions such as Disgust, Sadness and to a lesser extent Boredom that are associated with

negative activation (withdrawal) have a lower average of $F0$ values and a narrower fundamental frequency range.

The contributions of our work can be listed as:

- New deep 1D-CNN for gender-differentiated SER model is proposed.
- New emotion classification results are reported for the examined speech dataset.
- The proposed model is a new language and speaker-independent model.

Chapter 2 reviews recent studies related to SER. The proposed deep 1D-CNN models for emotion detection are described in Chapter 3. The experimental results of the algorithms are compared and discussed in Chapter 4. Our concluding remarks and possible future work directions are presented in Chapter 5.

CHAPTER 2

PROBLEM BACKGROUND

Human emotion detection in speech is a subset of emotion recognition problem. Tutoring systems used in distance learning which can detect bored or not interested users and can allow changing the style and the level of study material provided, for instance, benefiting from human emotion detection from speech with other further benefits due to its numerous applications, such as audio surveillance, E-learning, clinical studies, detection of lies, entertainment, computer games, and call centres. The advent of decent emotional speech recognition models could significantly improve the user experience in systems involving human-machine interactions, for example in the areas of Artificial Intelligence (AI) or Mobile Health (mHealth) [11].

There are three important factors under consideration when detecting emotion from speech:

1. Contents: "what is said"

2. Style/way: "how it was said"

3. Human(individual): It refers to the actor (male/female/child) "who says it".

The main steps in SER are speech database preprocessing (if needed), the process of feature extraction and setting up the classifier used to detect the emotions (see Figure 2.1).

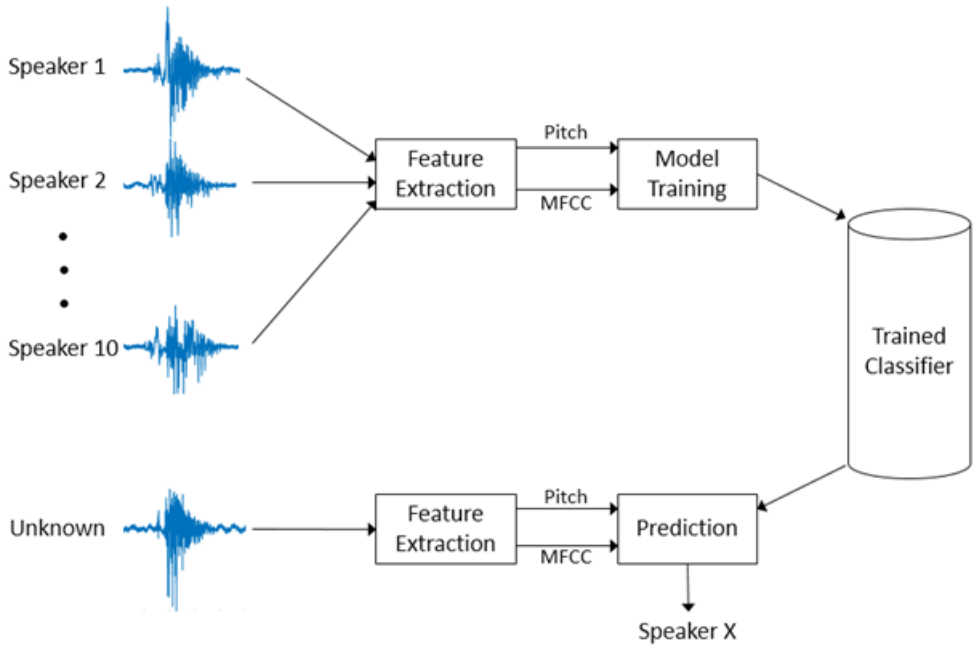

Figure 2.1: Classification Process

As a prerequisite, speech database collection is important for emotion recognition from speech. A lot of work has been done on collection techniques and evaluation of speech databases. There are many criteria that are used to evaluate the suitability of speech databases for different ML Classification methods [11].

According to Lee et al., human emotion classification is ultimately concerned with the Big Six emotions which are, anger, happiness, sadness, disgust, fear, and surprise [12]. These six emotions share similarities (in vocalization) across culture and language according to the consensus of researchers.

Feature selection and extraction is the second and one of the most important steps in SER studies [13]. Three broad types of speech variables have been identified as related to the expression of emotional states. These are fundamental frequency (F0) contour, continuous acoustic variables, and voice quality, respectively. Fundamental frequency contour has been used to describe fundamental frequency variation in

terms of geometric patterns. Continuous acoustic variables include the magnitude of fundamental frequency, intensity, speaking rate, and the distribution of energy across the spectrum. These acoustic variables are also referred to as the augmented prosodic domain. The terms used in a study by Dala to describe voice quality are tense, harsh, and breathy [13]. However, the work in this thesis does not employ these voice quality indicators.

Extraction of features is a critical step in SER, determining the final stage performance which ultimately is the accuracy of Machine Learning classifiers. In the case of relevant features, the combination of features can also be relevant in improving the recognition of emotions.

Classifiers are used for the recognition of subsets of human emotion. Several classifiers can be implemented for SER and their performance depends on the database design and features extracted from the speech. There are many Machine Learning classifiers to classify emotions from speech input.

Issa et al. proposed a framework for SER that uses a one-dimensional deep convolutional neural network (CNN1D) with five different audio features as input data [11]. The models are trained and tested on three datasets: RAVDESS, EMO-DB, and IEMOCAP. The best-performing model achieves higher accuracy than previous approaches for RAVDESS and EMO-DB datasets, setting a new state-of-the-art. For the IEMOCAP dataset, the model achieves a classification accuracy of 64.30%. The framework demonstrates simplicity, applicability, and generality. Han et al. proposed the use of deep neural networks (DNNs) to extract high-level features from raw data for SER [14]. The approach involves producing emotion state probability distributions for each speech segment using DNNs, constructing utterance-level features

from segment-level probability distributions, and using an extreme learning machine (ELM) for utterance-level emotion classification. Experimental results show that this approach improves accuracy by 20% compared to state-of-the-art approaches.

Badshah et al. prepared a discussion on a method for SER using spectrograms and deep CNNs. The proposed model consists of three convolutional layers and three fully connected layers that extract features from spectrogram images and predict the seven emotions [15]. The model was trained on spectrograms obtained from the Berlin emotions dataset and achieved accurate and efficient emotion predictions. The document also investigates the effectiveness of transfer learning using a pre-trained AlexNet model, but the results were not as successful. The proposed method shows promise for accurately predicting emotions from speech signals. Mittal et al. discussed a deep learning approach for real-time multiple-face recognition. The authors highlight the challenges involved in recognizing multiple faces in various conditions and propose a robust face recognition system [16]. They train a 34-layered Residual Network for face detection and an Inception Network for feature extraction using the triplet loss function. The models achieve high accuracy on different datasets. The authors also develop a mobile application for real-time attendance using the proposed face recognition model. The application outperforms fingerprint biometrics in terms of speed, cost, and group size. The document provides detailed information about the methodology, experimental results, and future scope of the research.

Bae et al. discussed a proposed method for voice recognition using Adaptive MFCC (Mel-Frequency Cepstral Coefficient) and Deep Learning [17]. The aim is to improve the voice recognition rate by extracting audio data from the original signal and reducing data loss. The proposed method uses a filter that is compactly built into the

data density area and applies weighted values to the data area to prevent data loss and improve the recognition rate. Deep Learning is also used to enable voice recognition without the need for a database, making it suitable for electronic devices with small memory capacity. The document describes the process of Adaptive MFCC and how it effectively removes noise for robust voice recognition. The method is tested in an experiment using recorded voice data, and the results show a recognition rate of about 96-98% for the Adaptive MFCC method. The document concludes that the proposed method is effective for voice recognition and further advances are being made to remove white noise, particularly for home appliances.

He et al. focused on understanding the behaviours of extremely deep networks, using simple architectures rather than pushing state-of-the-art results [18]. The architectures discussed include plain and residual networks, with an example network taking 32x32 pixel images as input, subtracting the per-pixel mean, and using layers of 3x3 convolutions. The network structure includes stacks of layers for different feature map sizes (32x32, 16x16, 8x8), with a specific number of layers and filters for each size. The number of filters increases as the feature map size decreases. Subsampling in the network is done by convolutions with a stride of 2, and the network ends with global average pooling, followed by a fully connected layer and softmax function. The design philosophy is inspired by VGG nets, with rules for the number of filters per layer and adjustments when the feature map size is halved.

Huang et al. focused on SER and explored the use of both verbal and nonverbal sounds in an utterance [19]. The researchers developed a detector for verbal/nonverbal sounds and used a Prosodic Phrase auto-tagger to extract segments. They then extracted emotion and sound features using CNNs and combined them into a generic feature

vector. Finally, a sequence-to-sequence model was used to generate an emotional sequence as the recognition result. Limet et al. studied the use of deep learning methods, specifically CNNs and Recurrent Neural Networks (RNNs), for SER [20]. The study aims to propose a SER method that does not rely on traditional hand-crafted features. The results of the study show that the proposed method achieves better accuracy in classifying emotions from speech compared to conventional methods. The study also highlights the challenges of recognizing emotions from speech and the limitations of traditional techniques that rely on low-level descriptors. Deep learning methods are presented as a more powerful and efficient approach to SER.

Trigeorgis et al. proposed a solution for recognizing emotions in speech by combining CNNs and Long Short-Term Memory (LSTM) networks [21]. The main objective is to extract context-aware emotional features from speech, which is considered a difficult task. The paper highlights that although LSTM networks have been used to model context, the extraction of emotional features is still an ongoing area of research. The proposed approach aims to automatically learn the most effective representation of emotional content in speech.

Livingstone et al prepared a document about the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [22]. It is a collection of audiovisual recordings of emotional speech and song in North American English. The database consists of recordings by 24 professional actors expressing various emotions such as calm, happy, sad, angry, fearful, surprise, and disgust. These expressions are produced at different levels of emotional intensity, including a neutral expression. The recordings are available in face-and-voice, face-only, and voice-only formats. They have been rated for emotional validity, intensity, and genuineness. The RAVDESS is freely avail-

able for research purposes under a Creative Commons license. Busso et al. prepared a study about the IEMOCAP database, which is a corpus collected by the Speech Analysis and Interpretation Laboratory at the University of Southern California [23]. It consists of approximately 12 hours of data recorded from ten actors in dyadic sessions. The data includes markers on the face, head, and hands, providing detailed information about their facial expressions and hand movements during scripted and spontaneous spoken communication scenarios. This database is a valuable resource for studying and modeling multimodal and expressive human communication.

Burkhardt et al. prepared an article that describes a database of German emotional speech [24]. This database consists of recordings of ten actors simulating different emotions. The article mentions that the database was evaluated in a perception test and is available for public access online. It also highlights that the study of emotional cues in speech has been receiving increased attention in recent years.

Zhao et al. studied the use of deep learning techniques, specifically 1D and 2D CNN LSTM networks, for SER [25]. These networks are designed to learn local and global emotion-related features from speech and log-mel spectrogram data. The architecture of the networks includes four local feature learning blocks and one LSTM layer. The main goal of the article is to enhance the recognition of speech emotion by extracting deep emotion features. Niu et al. prepared a study that proposes a data processing algorithm for SER using deep CNNs [26]. The algorithm utilizes the imaging principle of the retina and convex lens to capture various sizes of spectrogram and gather different training data. The study applies the AlexNet deep learning model to the IEMOCAP database and achieves an average accuracy of 48.8% on six emotions. The results indicate that the proposed algorithm enhances the accuracy of SER.

Tarantino et al. explored the use of self-attention and global windowing in the transformer model for SER [27]. They discussed how to incorporate these techniques to improve the performance of SER on the IEMOCAP database. The paper also investigates the use of soft targets derived from the distribution of annotations in the training data and finds that this approach outperforms majority voting, especially when there is a high level of agreement among annotators. The paper also emphasizes the importance of recognizing the emotional state of clients in automated telephone services.

Eyben et al. studied the Geneva Minimalistic Acoustic Parameter Set (GeMAPS), which is a standard set of acoustic parameters for voice research and affective computing [28]. The purpose of GeMAPS is to establish a common baseline for evaluation and eliminate differences caused by varying parameter sets or implementations. The set has been proven to have high-performance despite its small size. The article also emphasizes the significance of vocal expression in understanding affective states and the role of acoustic parameters in analyzing emotions and other affective dispositions. Triantafyllopoulos et al. explored the use of deep learning architectures for speech enhancement and its impact on SER [29]. The authors demonstrate how a scalable deep learning architecture can effectively remove noise while preserving enough information for accurate emotion identification in speech. The results show that incorporating a speech enhancement architecture is particularly beneficial in low signal-to-noise ratio conditions.

Schuller et al. studied the INTERSPEECH 2016 Computational Paralinguistics Challenge [30]. This challenge focused on three problems: classifying deceptive vs. non-deceptive speech, estimating the degree of sincerity, and identifying the native language of English L2 speakers. Participants were provided with baseline feature ex-

traction and classifiers. This challenge has been held eight times since 2009.

Weiskirchen et al. discussed the use of CNNs for recognizing emotional speech [31]. The authors propose a CNN-based classification architecture that utilizes spectrograms as representations of emotion-affected speech input. The network architecture is tested on three benchmark corpora and achieves impressive results, especially for the SUSAS corpus. The paper also explores the analysis of CNN's internal representations of the input. Chatziagapi et al. focused on the issue of data imbalance in SER and suggested the use of Generative Adversarial Networks (GANs) for data augmentation [32]. The authors propose a conditional GAN architecture to generate synthetic spectrograms for emotions that are underrepresented in the data. The effectiveness of this GAN-based approach is evaluated on two datasets, where it shows a relative performance improvement of 10% in one dataset and 5% in another when augmenting the minority classes.

Demircan et al. explored the use of the fuzzy C-means clustering algorithm for emotion recognition from speech signals [33]. The study focuses on extracting spectral features from speech signals, such as Mel frequency cepstral coefficients and linear prediction coefficients. These features are then used to identify cluster centres using the fuzzy C-means algorithm. The cluster centers are then used as input for classification using supervised classifiers like ANN, NB, kNN, and SVM. The study achieved a maximum success rate of 92.86% using the SVM classifier.

Yoon et al. proposed a deep dual recurrent encoder model that aims to enhance SER by combining audio and text data [34]. The model utilizes dual recurrent neural networks to encode information from both sources and predict the emotion class. The results show that the proposed model performs better than previous methods, achieving

accuracies between 68.8% and 71.8% in assigning data to emotion categories. Huang et al. proposed a method for SER using a semi-CNN [35]. The method involves two stages of training, where unlabeled samples are used to learn candidate features and then these features are used as input to the semi-CNN to learn affect-salient, discriminative features. This approach has shown to be stable and robust recognition performance in complex scenes and outperforms other established SER features.

Wu et al. discussed the use of modulation spectral features (MSFs) for automatic SER [36]. These features are extracted from a long-term spectro-temporal representation that is inspired by auditory processing. The MSFs capture both acoustic frequency and temporal modulation frequency components. The study shows that the MSFs have promising performance in classifying discrete emotion categories, outperforming commonly used short-term spectral representations. Additionally, when combined with prosodic features, the MSFs further enhance recognition performance.

Lampropoulos evaluated the use of MPEG-7 low-level audio descriptors for SER [37]. The authors conducted experiments using RBF-SVM classifiers and found that a combination of basic spectral and timbral features achieved an accuracy of 77.88%. This accuracy is comparable to other approaches that use high-level perceptual descriptors and human perception evaluation. The recognition of emotional states from voice is important for applications like media retrieval systems and call centre applications.

CHAPTER 3

PROPOSED ONE DIMENSIONAL DEEP LEARNING MODELS

3.1 Overview of the Chapter

In Section 3.2 of Chapter 3, dataset waveform graphs and the waveforms' beneficial properties are analyzed. In section 3.3, an ML model based on LSTM and in section 3.4 three ML models based on CNN (with one, two, and three CNN layers) are given. In section 3.5, employed problem-solving methods through the potential use of Python libraries such as Librosa and TensorFlow are sequentially given. Section 3.7 provides optionally comparison of recent ML models such as Random Forest, single Decision Tree employment, and AdaBoost with four Decision Tree estimators. In section 3.8 the mainly proposed ML model along with a brief description of its layers is provided.

3.2 RAVDESS Dataset Waveform Analysis

In Figure 3.1, the waveforms of eight different emotions (neutral, calm, happy, sad, angry, fearful, disgusted, and surprised) from the RAVDESS dataset are displayed. This visualization provides a graphical representation of the audio signals corresponding to each emotion, allowing for a comparative analysis of the waveform patterns associated with different emotional states. By examining these waveforms, researchers

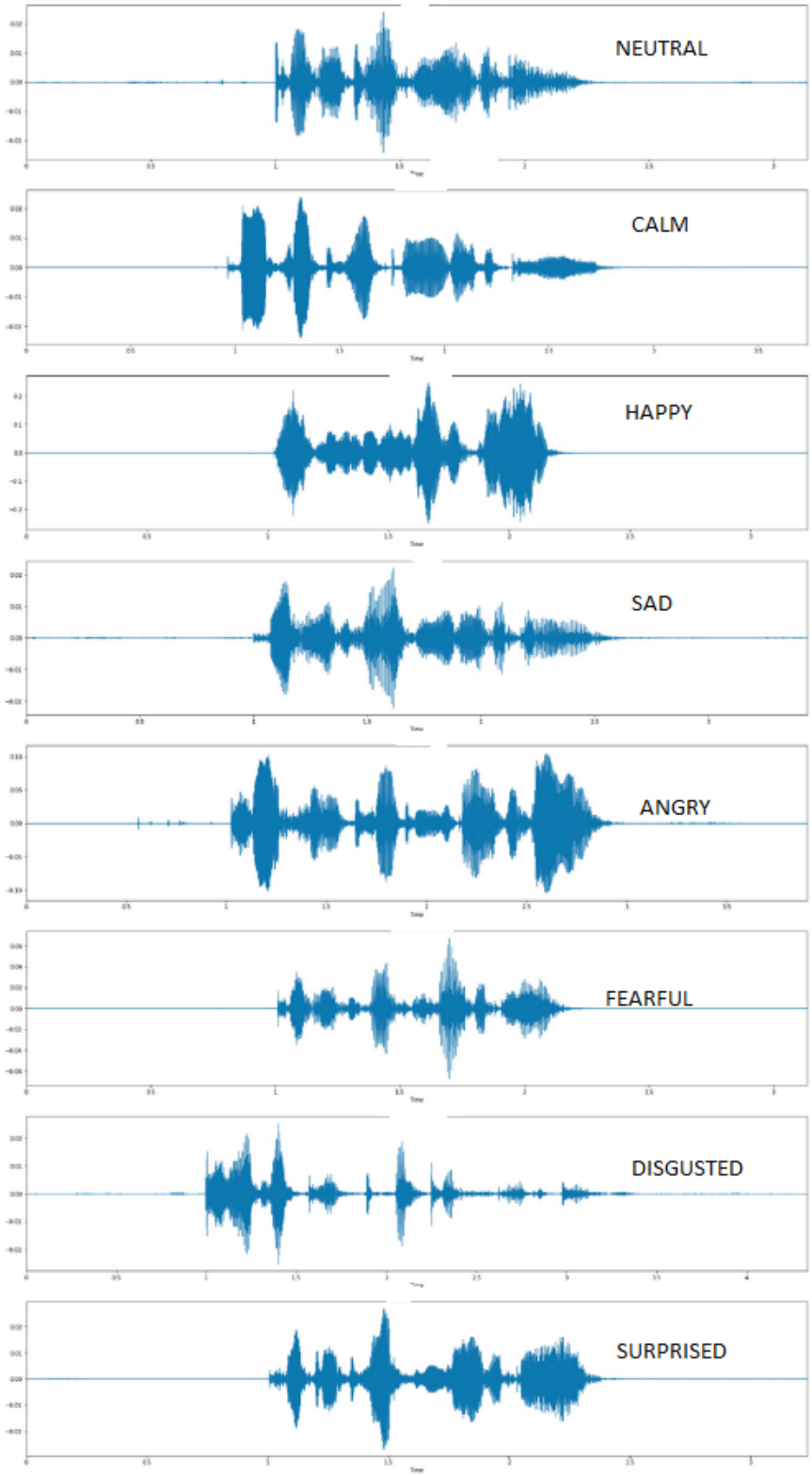

Figure 3.1: The waveforms of eight emotions from RAVDESS dataset

can gain insights into the acoustic characteristics and variations in speech signals that are indicative of various emotions, thereby laying the foundation for developing algorithms and models for emotion recognition in speech processing.

Voice recordings involving the six major emotions (happiness, sadness, anger, fear, surprise, and disgust) can contain a wide range of voice expressions and acoustic clues that express the underlying emotional states. For example, in a voice recording expressing happiness, one may hear a happy and upbeat tone, with variations in pitch, tempo, and intonation expressing feelings of joy and optimism. On the other hand, a voice recording representing sadness may demonstrate a slower pace, lower pitch, and subdued intonation, thus it may become possible to recognize feelings of sorrow and melancholy.

In recordings of anger, the voice may sound intense and forceful, with raised volume, sharp pronunciation, and aggressive speech patterns conveying feelings of frustration and hostility. Fearful voice recordings may feature trembling speech, rapid breathing, and a sense of urgency, reflecting heightened anxiety and worriedness]. Surprise in voice recordings can be characterized by sudden changes in pitch, volume, and tempo, conveying astonishment and scepticism.

Voice recordings expressing disgust may exhibit a tone of distaste or dislike, with cues such as speaking through the nose, being short of breath, and vocal tension signalling feelings of distaste or revulsion. By analyzing voice recordings involving the big six emotions, researchers and practitioners can gain valuable insights into the acoustic properties and vocal characteristics associated with different emotional states, contributing to advancements in emotion recognition, speech analysis, and effective computing applications.

3.3 Sequential LSTM Model

Long Short-Term Memory (LSTM) is a specialized type of recurrent neural network (RNN) designed to efficiently capture and store long-term dependencies in sequential data. Ordinary RNNs have difficulty with learning long-term dependencies due to problems such as the vanishing gradient problem, where gradients become too small, preventing the network from learning from earlier time steps in input data. LSTMs address this by introducing a complex architecture that includes gates(input, forget, and output gates) which regulate the flow of information. These gates allow LSTMs to maintain a memory cell that can selectively remember or forget past information, making them highly effective for tasks like time series prediction, natural language processing, and other problems where analyzing context over long sequences is necessary.

A sample sequential type LSTM model is shown in Figure 3.3. Sequential LSTM networks can handle speech data which is inherently sequential, where the arrangement and progression of data are essential. Time series data, language data (like sentences or documents), or any other ordered data can be effectively processed using sequential LSTMs.

3.4 Proposed One-Dimensional Deep Learning Models

3.4.1 Proposed 1D Deep Learning Model 1: single CNN layer

First, a simple basic model using a single 1D CNN layer, see Figure 3.3, was defined and its performance was evaluated. The results show a validation accuracy of 0.65

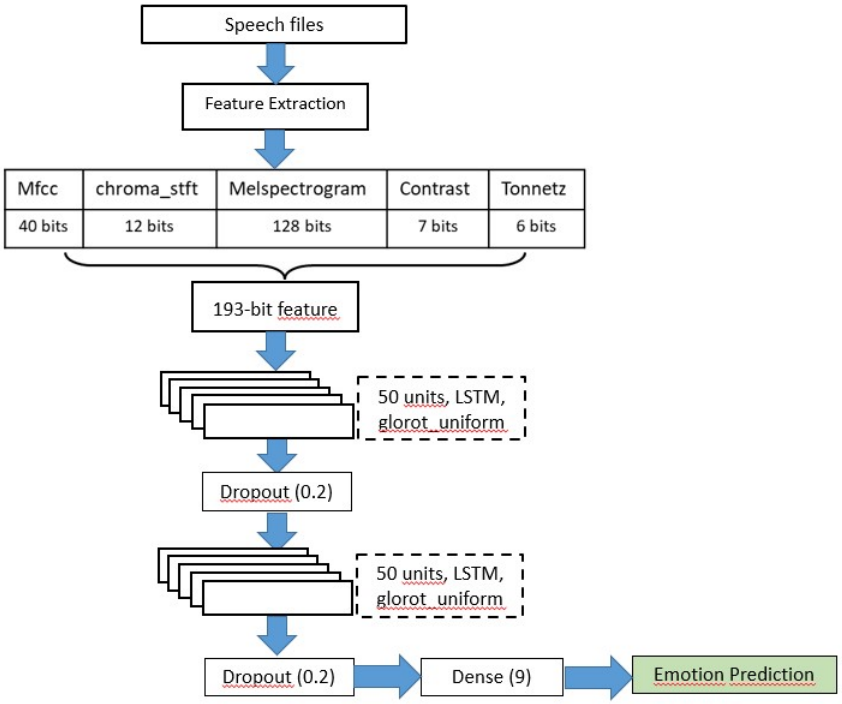

Figure 3.2: A sample Long Short-Term Memory model

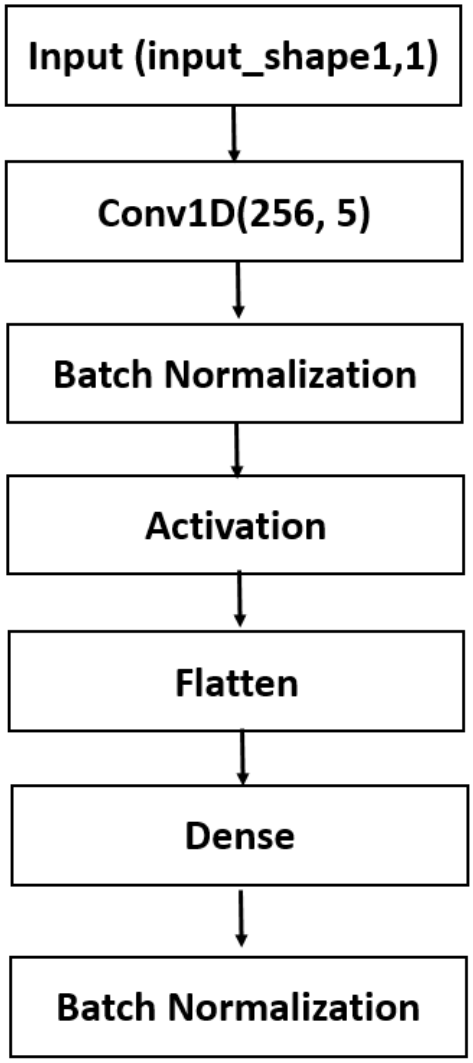

Figure 3.3: Model with single 1D CNN layer

was reached by using a "RELU" activation function. When "SELU" activation was used validation accuracy became 0.66. Finally, when "LeakyReLU" was used the validation accuracy reached the highest value of 0.68.

3.4.2 Proposed 1D Deep Learning Model 2: two CNN layers

Secondly, a model using two 1D CNN layers, see Figure 3.4, was defined and its performance was evaluated. Since "LeakyReLu" gave best results for the first model

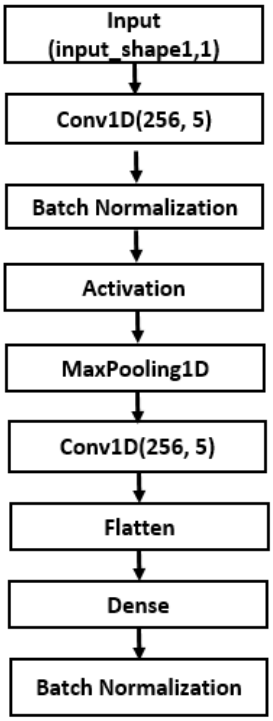

Figure 3.4: Model with two 1D CNN layers.

only this activation function was tested. The resulting validation accuracy is 0.71.

3.4.3 Proposed 1D Deep Learning Model 3: three CNN layers

Thirdly, a model using three 1D CNN layers, see Figure 3.5, was defined and its performance was evaluated. Again "LeakyReLu" activation function was used. The resulting validation accuracy is 0.73.

CNN1D model alternative to the mainly proposed model shown in Figure 3.6:

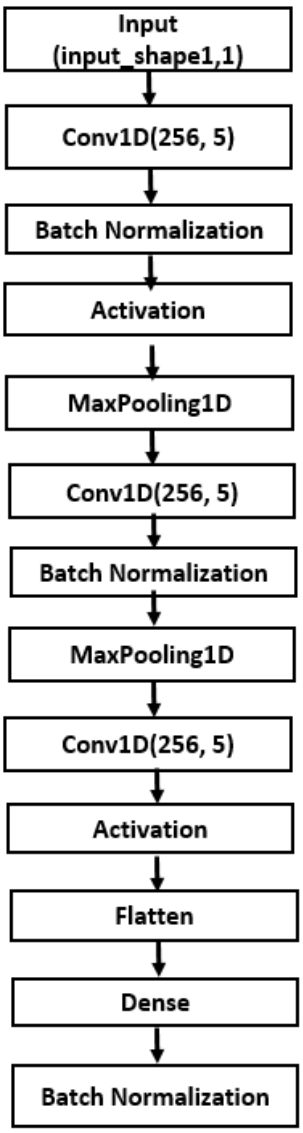

Figure 3.5: Model with three 1D CNN layers.

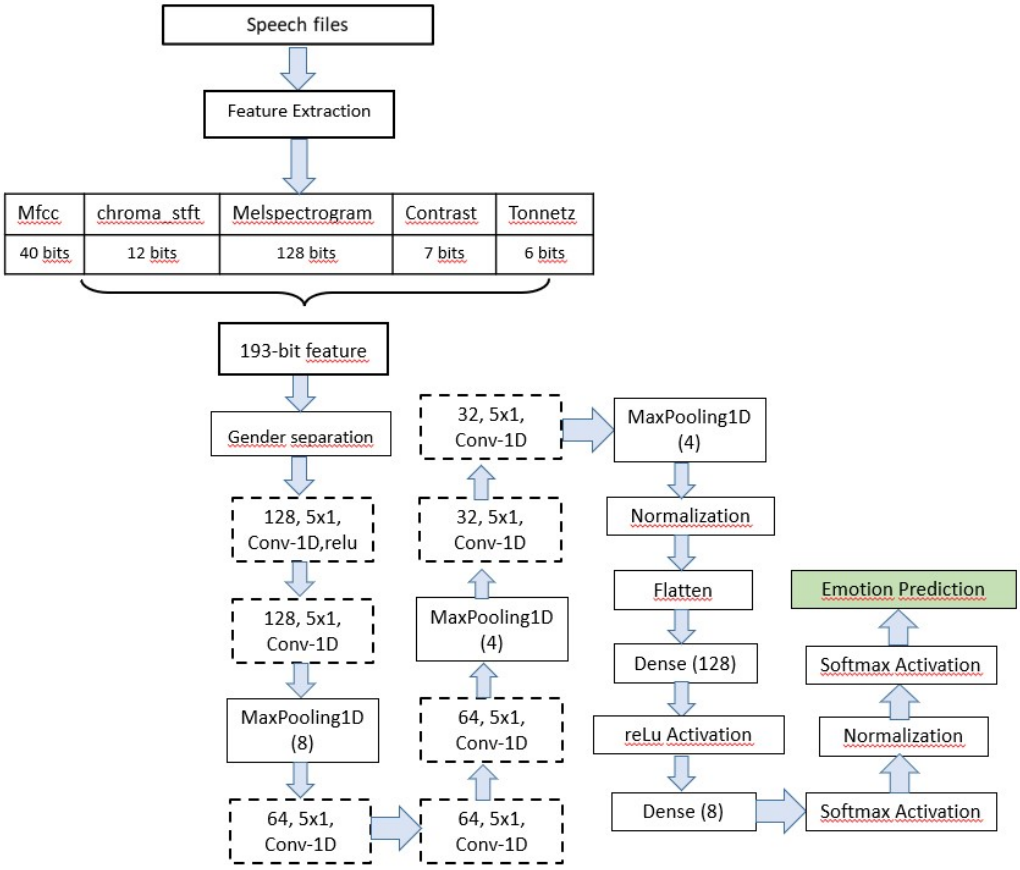

Figure 3.6: Alternative 1D Convolutional Neural Network model

3.5 Implementation Details of the Proposed Models

Librosa is a Python library specifically designed for audio and music signal processing tasks. It provides a wide range of functionalities for loading, analyzing, and manipulating audio data, making it a popular choice for researchers and practitioners working in fields such as speech recognition, music information retrieval, and sound processing. The key features of the Librosa library and how it can be used to prepare speech features for input data in SER CNN models as follows:

Audio Loading and Decoding: Librosa can load audio files in various formats, including WAV, MP3, and FLAC. It provides functions to read audio files from disk and decode them into numpy arrays, which can be directly used as input to deep learning models.

Feature Extraction: One of the primary functionalities of Librosa is extracting various audio features from the input signal. For SER, commonly used features include: Mel-frequency Cepstral Coefficients (MFCCs): MFCCs are a widely used feature representation in speech processing tasks. Librosa provides functions to compute MFCCs from audio signals, capturing spectral characteristics that are important for speech analysis.

Mel Spectrogram: Librosa can compute the Mel spectrogram representation of an audio signal, which represents the power spectral density of the signal on a Mel-scale.

Chroma Features: Chroma features capture the pitch content of the audio signal and are useful for tasks involving musical or tonal aspects of speech.

Rhythm Features: Librosa also offers functions to compute rhythm-related features such as tempo, beat tracking, and onset detection, which can be relevant for certain types of emotion recognition tasks.

Preprocessing and Normalization: Librosa provides utilities for preprocessing audio data, such as resampling, normalizing, and applying transformations like log scaling or power normalization to the extracted features. Preprocessing is crucial for ensuring that the input data is standardized and suitable for the neural network model.

Integration with Deep Learning Frameworks: Librosa seamlessly integrates with popular deep learning frameworks like TensorFlow and PyTorch. The extracted audio features can be easily fed into CNN models for training and evaluation.

In the context of preparing input data for SER CNN models, Librosa can be used to extract relevant features such as MFCCs or Mel spectrograms from the raw audio signals. These features capture important spectral and temporal characteristics of the speech signals, which are then used as input to the CNN models. By leveraging Librosa's functionalities, researchers can efficiently preprocess and extract informative features from audio data, facilitating the development of accurate and robust SER systems.

3.6 One-dimensional CNN Components

One-dimensional Convolutional Neural Networks (1D CNNs) are deep learning models commonly used for sequential data processing, such as time-series data or one-dimensional signals like speech audio. 1D CNN model components can be described briefly as follows:

Convolutional Layers: Similar to their 2D counterparts used in image processing, 1D CNNs employ convolutional layers to extract local patterns or features from the input sequence. The convolution operation involves sliding a filter/kernel across the input sequence and computing dot products to produce feature maps capturing different aspects of the input data.

Pooling Layers: After convolutional layers, pooling layers are often used to downsample the feature maps, reducing the dimensionality of the data while retaining important information. *Max pooling* and *average pooling* operations are commonly used to aggregate information from neighboring regions of the feature maps.

Activation Functions: Non-linear activation functions like ReLU (Rectified Linear Unit) are applied after convolutional and pooling operations to introduce non-linearity into the model, enabling it to learn complex patterns and representations from the input data.

Fully Connected Layers: Towards the end of the model, one or more fully connected layers are typically used to integrate the learned features and make predictions. These layers combine the extracted features from earlier layers and map them to the output classes through their learned weights and biases.

Regularization Techniques: Regularization techniques like dropout or batch normalization may be applied to prevent overfitting (memorization of all training instances) and can improve the generalization ability of the model.

Output Layer: The output layer of the network typically consists of softmax activation for classification tasks, producing probability distributions over the possible classes. For tasks like SER, the output layer might have multiple units corresponding to dif-

Table 3.1: Emotion labels column

Feature sample id	Emotion label
0	1
1	1
2	1
3	1
4	2
..	..
1435	8
1436	8
1437	8
1438	8
1439	8

ferent emotion classes.

Table 3.1 contains the emotion label for each wave file and its value is obtained by the Numpy function: $y = data.iloc[:,-1].copy()$

In Table 3.1 the second column containing emotion labels can be used as target emotion.

3.7 Traditional Machine Learning Classifiers

Machine learning classifiers play a pivotal role in pattern recognition and predictive modeling tasks by categorizing data points into distinct classes based on their features. These classifiers utilize algorithms to learn patterns from labelled training data and make predictions on unseen data. Commonly used classifiers include *Random Forest*, *Decision Tree*, and *AdaBoost*, each with unique characteristics and performance metrics. *Random Forest* is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes as the prediction. *Decision Tree* models create a tree-like structure where each internal node represents a fea-

ture, and each leaf node corresponds to a class label. *AdaBoost*, on the other hand, is an adaptive boosting algorithm that combines multiple weak classifiers to create a strong classifier, with a focus on improving accuracy by assigning higher weights to misclassified data points.

These machine learning classifiers are essential tools in various domains such as image recognition, sentiment analysis, and medical diagnosis, offering valuable insights into complex datasets and enabling automated decision-making processes. The choice of classifier depends on the nature of the data, the desired accuracy level, and the interpretability of the model. Researchers and practitioners often experiment with different classifiers to determine the most suitable approach for a given task, considering factors like computational efficiency, model complexity, and generalization capabilities. By understanding the strengths and limitations of different classifiers, practitioners can effectively leverage machine learning techniques to solve classification problems and extract meaningful patterns from data, contributing to advancements in artificial intelligence and data-driven decision-making processes.

3.7.1 Random Forest Classifier

The Random Forest Classifier is an ensemble machine learning algorithm that operates by constructing a multitude of decision trees during training and outputs the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Figure 3.7 details the creation of a Random Forest Classifier model for SER. Each tree in the forest is grown using a subset of the training data and a subset of features, introducing randomness to the model. This randomness helps in reducing overfitting and improving generalization. Predictions are made by aggregat-

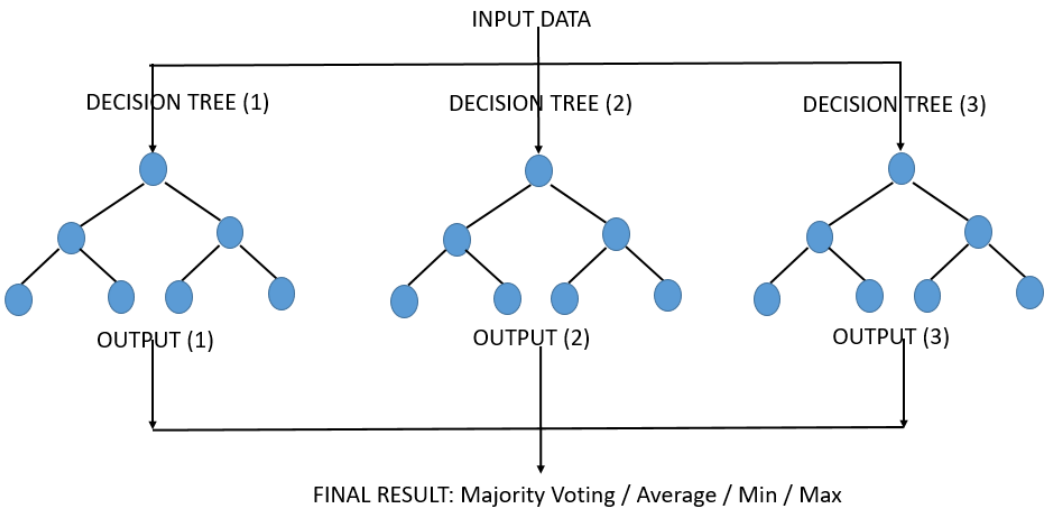

Figure 3.7: Random Forest Classifier.

ing the predictions of all the trees, providing a robust and accurate classification or regression model. The algorithm is known for its flexibility, scalability, and ability to handle high-dimensional datasets, making it widely used in various applications, including classification, regression, and feature importance analysis.

Figure 3.7 outlines the steps involved in setting up and training the Random Forest model, including parameter selection, data splitting for training and testing, and model evaluation. The Random Forest algorithm can produce a model capable of effectively distinguishing between different emotional states based on the extracted audio features. The creation of the Random Forest Classifier model is an important stage in the development of the emotion recognition system, presenting the utilization of advanced machine learning techniques to enhance the classification performance and overall effectiveness of the system

The max_depth parameter in a random forest classifier controls the depth of each individual decision tree in the forest. It limits the number of levels a node can be expanded

to. A larger max_depth value allows each tree to grow deeper, potentially capturing more complex patterns in the data. However, it also increases the risk of overfitting, as the trees become more specific to the training data and less generalizable to unseen data. Choosing the optimal max_depth value depends on the specific dataset and problem. It is often chosen through a process of experimentation and validation.

Here are some general guidelines for setting max_depth:

Start with a small value: A good starting point is to set max_depth to a small value, such as 5 or 6. This helps to prevent overfitting and ensures that the trees remain relatively simple.

Increase the value gradually: Once you have a baseline model, you can gradually increase the max_depth value and observe the impact on the model's performance.

Use cross-validation: Use a cross-validation technique such as k-fold cross-validation to evaluate the model's learning performance for different max_depth values. This helps to select the value that provides the best balance between overfitting and underfitting.

3.7.2 Decision Tree Model

The Decision Tree Classifier is a machine learning algorithm that creates a tree-like structure by recursively partitioning the dataset based on the features, making decisions at each node to maximize information gain or minimize impurity. It is a powerful and interpretable model used for both classification and regression tasks. The tree structure consists of nodes representing conditions on features, branches representing the possible outcomes of those conditions, and leaves representing the final

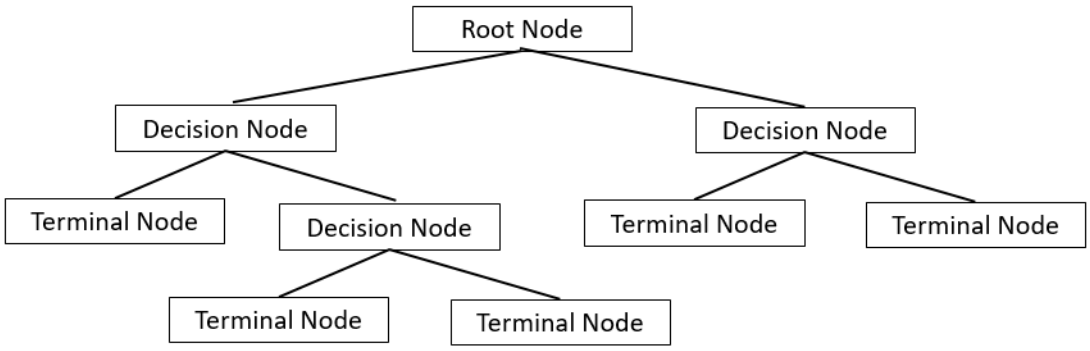

Figure 3.8: Creation of a single Decision Tree classifier

class labels or regression values. Decision trees are capable of capturing complex relationships in the data and are resistant to irrelevant features. However, they are susceptible to overfitting, and techniques like pruning are often applied to control the tree's depth. Decision Tree Classifier excels in providing transparent insights into the decision-making process, making it valuable in scenarios where interpretability is crucial.

Figure 3.8 presents an example machine learning model that incorporates a single Decision Tree classifier. The Decision Tree Classifier is a fundamental machine learning algorithm that creates a tree-like structure by recursively partitioning the dataset using feature-based conditions to make predictions. Figure 3.8 illustrates the implementation of a Decision Tree model. Implementation of a Decision Tree classifier provides parameters for specifying features such as the maximum depth of the tree and other hyperparameters essential for model training. By utilizing a single Decision Tree classifier, researchers can analyze the decision-making process of the model and understand how it partitions the feature space to classify different emotional states in speech signals. Figure 3.8 serves as a practical demonstration of configuring a Decision Tree model within the machine learning framework, highlighting the simplicity

and interpretability of Decision Trees in the context of SER tasks.

3.7.3 AdaBoost with Decision Tree Estimators

The AdaBoost (Adaptive Boosting) Classifier is an ensemble learning algorithm that combines the predictions of multiple weak learners, typically decision trees, to create a strong, accurate classifier. AdaBoost assigns weights to instances in the dataset and adjusts them at each iteration to focus on misclassified samples, making subsequent weak learners prioritize the difficult-to-classify instances. The algorithm then combines the predictions of all weak learners through a weighted sum, where the weights are determined by each learner's accuracy. This iterative process continues until a predetermined number of weak learners are trained or a perfect model is achieved. AdaBoost is known for its ability to improve the performance of weak learners, adapt to complex datasets, and mitigate overfitting. It is widely used in classification tasks and demonstrates robustness across various domains.

Figure 3.9 shows the function calls for defining a machine learning model using AdaBoost with four Decision Tree Estimators. AdaBoost, short for Adaptive Boosting, is an ensemble learning algorithm that combines the predictions of multiple weak learners, often decision trees, to create a robust and accurate classifier. In this context, AdaBoost is configured to utilize four Decision Tree Estimators as base learners, leveraging their individual predictive capabilities to enhance the overall classification performance. The code includes the initialization of the AdaBoost classifier, specifying parameters such as the number of estimators, learning rate, and other hyperparameters crucial for the boosting process.

```
#ADABOOST
#
import pandas as pd
import sklearn.model_selection
datal = pd.read_csv('/content/drive/MyDrive/test_dat.csv')
numcol = datal.shape[1]

Xdata = datal.iloc[:, :numcol-1]

y=datal.iloc[:,-1].copy()

# Split the dataset into train and test sets.
X_train, X_test, y_train, y_test = sklearn.model_selection.train_test_split(Xdata,
y, test_size = 0.2, random_state = 30)
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import AdaBoostClassifier
tdepth = 3
learning_rate=0.1
dtc1 = DecisionTreeClassifier(max_depth=tdepth)
dtc2 = DecisionTreeClassifier(max_depth=tdepth)
dtc3 = DecisionTreeClassifier(max_depth=tdepth)
dtc4 = DecisionTreeClassifier(max_depth=tdepth)
dtc1.fit(X_train, y_train)
dtc2.fit(X_train, y_train)
dtc3.fit(X_train, y_train)
dtc4.fit(X_train, y_train)
base_learners = [dtc1,dtc2,dtc3,dtc4]

# Create an AdaBoost classifier with the specified base learners
adaboost_model = adaboost_classifier = AdaBoostClassifier(
    estimator=DecisionTreeClassifier(max_depth=tdepth),
    n_estimators=len(base_learners),
    learning_rate=learning_rate,
    random_state=30
)

# Fit the AdaBoost classifier on the training data with explicit base learners
adaboost_model.estimators_ = base_learners
adaboost_classifier.fit(X_train, y_train)

# Make predictions on the test set
y_pred = adaboost_classifier.predict(X_test)

# Evaluate the accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')
```

Figure 3.9: Code to define a model using adaBoost with four Decision Tree Estimators

The hyperparameters for AdaBoost are as follows:

n_estimators: This parameter determines the number of weak learners (e.g., decision trees or classifiers) to be used in the ensemble.

learning_rate: This parameter controls the contribution of each weak learner to the final combination. A smaller learning rate typically requires more weak learners to achieve similar performance, but it can lead to better generalization.

base_estimator: AdaBoost can work with any base estimator. The base estimator is typically a decision tree, but other classifiers can also be used.

algorithm: This parameter specifies the algorithm to be used for the boosting process. It can be either "SAMME" or "SAMME.R". "SAMME" uses a discrete boosting algorithm, while "SAMME.R" uses the real boosting algorithm.

random_state: This parameter sets the random seed for reproducibility.

In the model proposed by this thesis, four Decision Trees are incorporated within the AdaBoost framework, with the aim of exploiting the strengths of both algorithms, such as the interpretability of Decision Trees and the boosting capabilities of AdaBoost, to improve the model's ability to classify emotions in speech signals effectively.

3.8 Proposed 1D CNN based Machine Learning model

Figure 3.10 presents the proposed model used for solving the SER problem. We also use it as a reference point for evaluating the performance of other machine learning models used in the literature, such as Random Forest, Decision Tree, and Decision Trees augmented with AdaBoost. The proposed model represents a simple and straightforward approach to emotion classification, providing a benchmark against

```
#Proposed model: CNN1D
#

def model1(input_shape1):
    model = Sequential()

    model.add(Conv1D(256, 5,padding='same',
                     input_shape=(input_shape1,1)))
    model.add(BatchNormalization())
    model.add(Activation('relu'))


    model.add(Conv1D(256, 5,padding='same'))
    model.add(Activation('relu'))
    model.add(Dropout(0.1))

    model.add(BatchNormalization())

    model.add(MaxPooling1D(pool_size=(8)))
    model.add(Conv1D(256, 5,padding='same'))


    model.add(Activation('relu'))
    model.add(Dropout(0.2))



    model.add(Flatten())
    model.add(Dropout(0.2))
    model.add(Dense(8))
    model.add(BatchNormalization())
    model.add(Activation('softmax'))

    return model
```

Figure 3.10: Proposed 1D CNN based model

which the effectiveness of more complex models can be compared. By visualizing the proposed model, researchers can establish a starting point for assessing the accuracy and efficiency of their classification system before implementing more sophisticated algorithms. This figure underscores the importance of establishing a baseline performance level to gauge the improvements achieved by subsequent model iterations and enhancements in the SER task. Understanding the proposed model's capabilities and limitations is essential for refining and optimizing the overall classification system to achieve superior accuracy and reliability in emotion detection from speech signals.

The provided model is a Convolutional Neural Network (CNN) designed for 1D data, commonly used in tasks like time series analysis and signal processing. The model architecture consists of multiple Conv1D layers with varying numbers of filters (128, 64, 32) and kernel sizes of 5, each followed by MaxPooling1D layers to downsample the data. The use of MaxPooling helps in reducing the spatial dimensions and extracting key features from the input data.

The model includes BatchNormalization layers after some Conv1D layers to improve training stability and convergence by normalizing the activations. After the final Conv1D layer, the data is flattened to be fed into Dense layers for classification. Two Dense layers are added with 128 and 8 units, respectively, with ReLU activation in the first Dense layer and softmax activation in the second Dense layer for multi-class classification tasks.

CHAPTER 4

EXPERIMENTAL RESULTS AND BENCHMARK DATASET

The experiments of this thesis are conducted on a PC with Ryzen 5 3600 CPU @ 3.60GHz and 16 GB of memory. The code is developed in Jupyter-notebook with version 6.5.4, and executed on Google's Colab research environment. One well-known dataset RAVDESS is used in the experiments. Data augmentation is not used on the datasets in order to make a fair comparison with other algorithms' results reported in the literature. Five-fold cross-validation is used for correctly calculating model performance in the experiments. There was no over or under-fitting problem detected since the testing and training datasets' cross-validation and inclusion of regularization results in similar performance results. Each deep learning model is executed with 200, 500, and 700 epochs for RAVDESS and results are compared against those reported in recent literature, to make a fair comparison.

4.1 RAVDESS Dataset Used in the Experiments

The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) is chosen as the dataset for our model because of its great availability [38]. This dataset contains audio and visual recordings of 12 male and 12 female actors pronouncing English sentences with eight different emotional expressions. For our task, we utilize only speech samples from the database with the following eight different emotion

classes: sad, happy, angry, calm, fearful, surprised, neutral and disgust. The overall number of utterances is 1440 [11].

RAVDESS is a multimodal dataset of emotional speech and singing. The database includes 24 professional actors who vocalize lexically-matched statements in a neutral North American accent. There are 720 male and 720 female speech files in this dataset. "Calm", "happy", "sad", "angry", "fearful", "surprise", and "disgust" expressions can be found in a speech file, while calm, happy, sad, angry, and fearful emotions can be found in a song.

The RAVDESS dataset was used to train the Neural Network and measure the accuracy performance of our model. The success rate of 5-fold crossover employed training for the model is examined. This means the RAVDESS dataset will be divided into five 20% sets. Four pieces which make up the 80% will be used to train the model and over the remaining 20% the model's accomplishment rate will be measured.

4.2 Feature extraction

In this section, we give information about the feature set we used in our study. The best structure of the feature set in terms of prediction accuracy was identified through a literature review and experimental results for our 1D-CNN model.

Librosa (a Python library) is used for feature extraction on voice files. Totally, 193 features are obtained from each file. The extracted features can be summarized as given below:

- MFCC is a signal composed of a small number of features (typically 10-20)

that concisely describe the overall shape of a spectral envelope. It identifies the components of the audio signal that are useful for identifying linguistic content while discarding everything else that carries information such as background noise, emotion, and so on.

- Chroma STFT: (A chromagram from a waveform or power spectrogram) Computing discrete Fourier transforms over short overlapping windows, short-time Fourier transform (STFT) represents a signal in time-frequency domain.

- Mel-scaled spectrogram: Mel-scale is a pitch unit in which equal distances in pitch appear equally distant to the listener. These characteristics, to some extent, mimic the human reception pattern of sound frequency.

- Spectral contrast: Each spectrogram S frame is divided into sub-bands. The energy contrast for each sub-band is calculated by comparing the mean energy in the top quantile (peak energy) to that of the bottom quantile (valley energy). High contrast values are generally associated with clear, narrow-band signals, whereas low contrast values are associated with broad-band noise.

- Tonnetz: is similar to the chromagrams according to the harmony and pitch classes' representation. The tonal centroids of a voice instance are measured in six-dimensional pitch space. Pitch classes with tight harmonic linkages have shorter Euclidean distances because the tonal centroid is based on the Harmonic Network, which depicts pitch relationships in a planar manner.

40, 12, 128, 7, and 6 features from MFCC, chroma, Mel-Spectrogram, contrast, and tonnetz are obtained respectively.

Table 4.1: Accuracy with increasing Random Forest maximum depth

RF Depth	Accuracy
2	0.32
3	0.36
4	0.39
5	0.43
6	0.48
7	0.49
8	0.52
9	0.55
10	0.56
11	0.58

Table 4.2: Accuracy with increasing Decision Tree depth

Decision Tree depth	Accuracy
2	0.26
3	0.32
4	0.38
5	0.36
6	0.37
7	0.38
8	0.38
9	0.39

4.3 Experimental Performance Results of Machine Learning Classifiers

4.3.1 Random Forest Classifier

The accuracy of the Random Forest model could be improved from 0.32 up to 0.58 as seen in Table 4.1. But, degradation in accuracy was observed when maximum depth was increased beyond 11.

Table 4.3: Accuracy of adaBoost with four Decision Trees with increasing depths from 6 to 10

AdaBoost maximum depth	Accuracy
6	0.43
7	0.43
8	0.45
9	0.44
10	0.43

4.3.2 Decision Tree Model

The accuracy of the Decision Tree model was improved from 0.26 up to 0.39 as seen in Table 4.2. But no further improvement could be observed when tree depth was increased beyond 9, causing even a slight reduction.

4.3.3 AdaBoost with Decision Tree Estimators

The accuracy improvement on Decision Trees by AdaBoost was experimentally measured to be from 0.43 up to 0.45 as seen in Table 4.3.

4.4 Experimental Evaluation of LSTM model

This model consisting of two LSTM layers and Dropout layers in succession to them was intended as an experiment of using only LSTM layers with premonition of low accuracy rate.

The results given in the following section were obtained with the setup given in Section 3.4:

Table 4.4: Training of 1D CNN model for 500 epochs: best validation accuracy: 0.7326, validation loss: 1.0222

Epoch/500	Accuracy	Loss	Validation Accuracy	Validation Loss
1	0.1407	2.2275	0.1354	14.2557
2	0.2825	1.8467	0.1354	8.6633
3	0.3272	1.7759	0.1354	5.5915
4	0.3695	1.6694	0.1632	3.9756
5	0.3896	1.6377	0.2535	2.2615
10	0.5018	1.3948	0.3056	1.7103
50	0.9426	0.3875	0.5764	1.2017
100	0.9990	0.0846	0.7188	0.9034
200	1.0000	0.0221	0.7118	1.0150
300	0.9983	0.0126	0.6667	1.4148
400	**1.0000**	**0.0043**	**0.7326**	**1.0222**
500	0.9959	0.0086	0.7118	1.1226

This result goes on to show that an LSTM layer-only based model is not on par with models employing Convolutional Neural Network layers. Although LSTM-based models are typically used in Recurrent Neural Networks with the task of classifying sequential or temporal data, LSTM layers can be preferably used in conjunction with CNN layers to obtain acceptable results.

4.5 Experimental Evaluation of the Performance of 1D Deep Learning Models

Figure 4.1 presents the code for performing Principle Component Analysis (PCA) and producing a Dendrogram as a result of Hierarchical Clustering Analysis (HCA).

Figure 4.2 shows the results of Hierarchical Clustering Analysis in the form of a Dendrogram, showing the distance between components at which these Principle components split.

```
from sklearn.decomposition import PCA
from scipy.cluster.hierarchy import dendrogram, linkage


# Perform hierarchical clustering
linkage_matrix = linkage(Xdata, method='ward')

# Plot the dendrogram
plt.figure(figsize=(10, 5))
dendrogram(linkage_matrix)
plt.title('Hierarchical Clustering Dendrogram')
plt.xlabel('Sample index')
plt.ylabel('Distance')
plt.show()


# Create PCA object
pca = PCA(n_components=data.shape[1])

# Fit PCA to the data
pca.fit(data)

# Access the explained variance ratio
explained_variance_ratio = pca.explained_variance_ratio_

print("Explained variance ratio for each principal component:", explained_variance_ratio)

# Get the loadings
loadings = pca.components_

# Find the number of principal components needed to retain 95% of variance
n_components = np.argmax(np.cumsum(explained_variance_ratio) >= 0.95) + 1

# Get the most important components
important_components = loadings[:n_components]

# Find the original features
original_features = np.argsort(np.abs(important_components), axis=1)[:, ::-1]

print("Original features for each principal component:")
for i, features in enumerate(original_features):
    print(f"Principal Component {i+1}: {features}")
```

Figure 4.1: Principle Component Analysis and Hierarchical Clustering Analysis

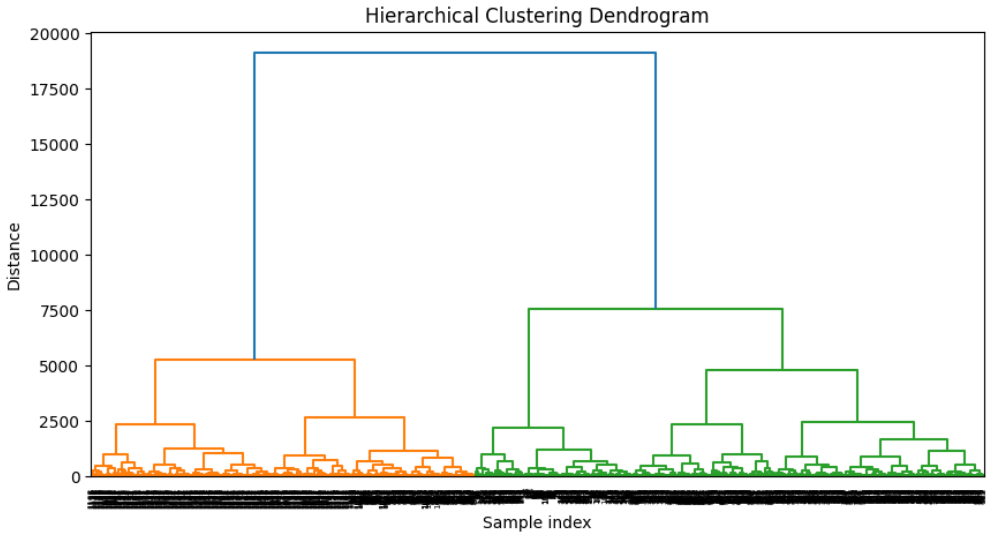

Figure 4.2: Hierarchical Clustering Analysis dendrogram for RAVDESS data

4.6 SER Accuracy Performance with Female Actors Data

In order to measure the SER performance of the proposed model, first training is done using female actors data in RAVDESS dataset and then 5-fold crossover validation are calculated to prevent the effects of the selection of easy instances for validation. Figure 4.3 shows the validation results for five different training and validation data sets. These results show that a validation accuracy of above 70% is achieved. The achieved validation accuracy for each fold are: 0.78, 0.75, 0.71, 0.73, 0.78.

Figure 4.4 shows the validation results for five different training and validation data sets of male actors. These results show that a validation accuracy of about 70% is achieved. Figure 4.4 shows that the achieved accuracy levels using male actors' data are about 5% less than the accuracy levels for female actors. The achieved validation accuracy values for each fold are: 0.81, 0.72, 0.66, 0.68, and 0.72. We conclude that the performance of the model proposed in this thesis is robust and independent of the

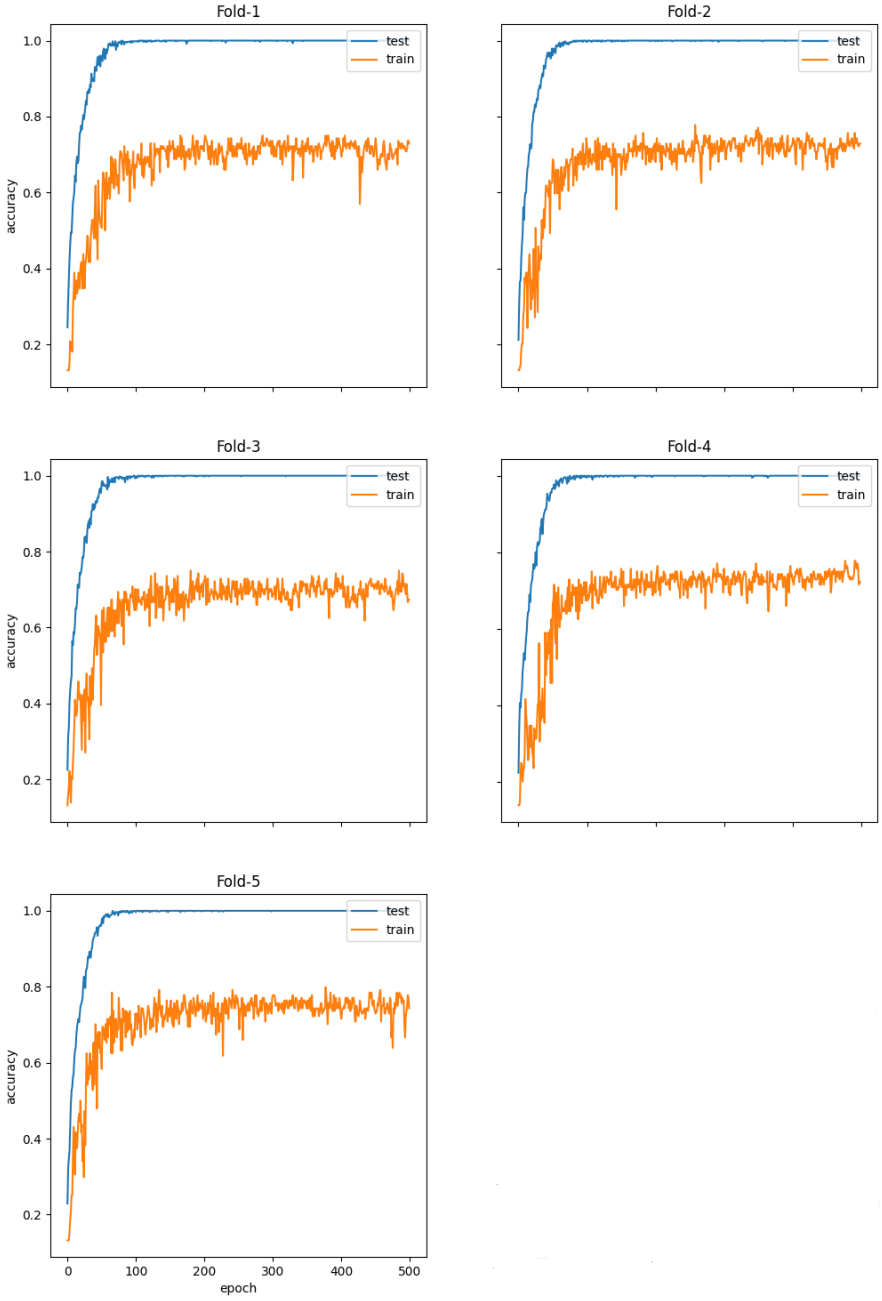

Figure 4.3: The validation scores of female actors (500 epochs).

Table 4.5: Best accuracy and validation accuracy values using various machine learning models

Algorithm	Accuracy	Loss	Validation Accuracy	Validation Loss
Decision Trees	-	-	0.39	-
Random Forest	-	-	0.58	-
1 layer 1D CNN	0.6875	0.9500	0.5556	1.2345
2 layer 1D CNN	0.8559	0.4228	0.6250	1.1331
3 layer 1D CNN	1.00	0.0043	0.7326	1.0222
2 layer LSTM	0.63	1.03	0.125	3.31

subset of training data used for training.

If all of the male and female actors data are used for training and validation using 5-fold crossover validation, the validation scores are: 0.69, 0.71, 0.68, 0.67, and 0.64. In Figure 4.5 the validation accuracy scores for all of the five folds are displayed. These values are a little below those obtained by all female training data, but they provide an acceptable accuracy level for SER. In the future, tests can be made for male data, while model training is done using only female data.

The accuracy results given in Table 4.6 clearly show the superiority of the 1D CNN model with 3 convolution layers. The number of convolution layers in 1D CNN models has been increased first from 1 to 2 and then to 3 and each layer has resulted in an improvement in the validation accuracy. Random Forest also has resulted in a promising accuracy value of 58 %, which is remarkable for such a simple model. The LSTM model requires the time dimension to be represented in input. Since all of the 193 features are derived from the complete speech data file, the time dimension has been modeled by assigning each one of the 193 features as a time step. The LSTM performance shows that the modeling of time in this way is reasonable.

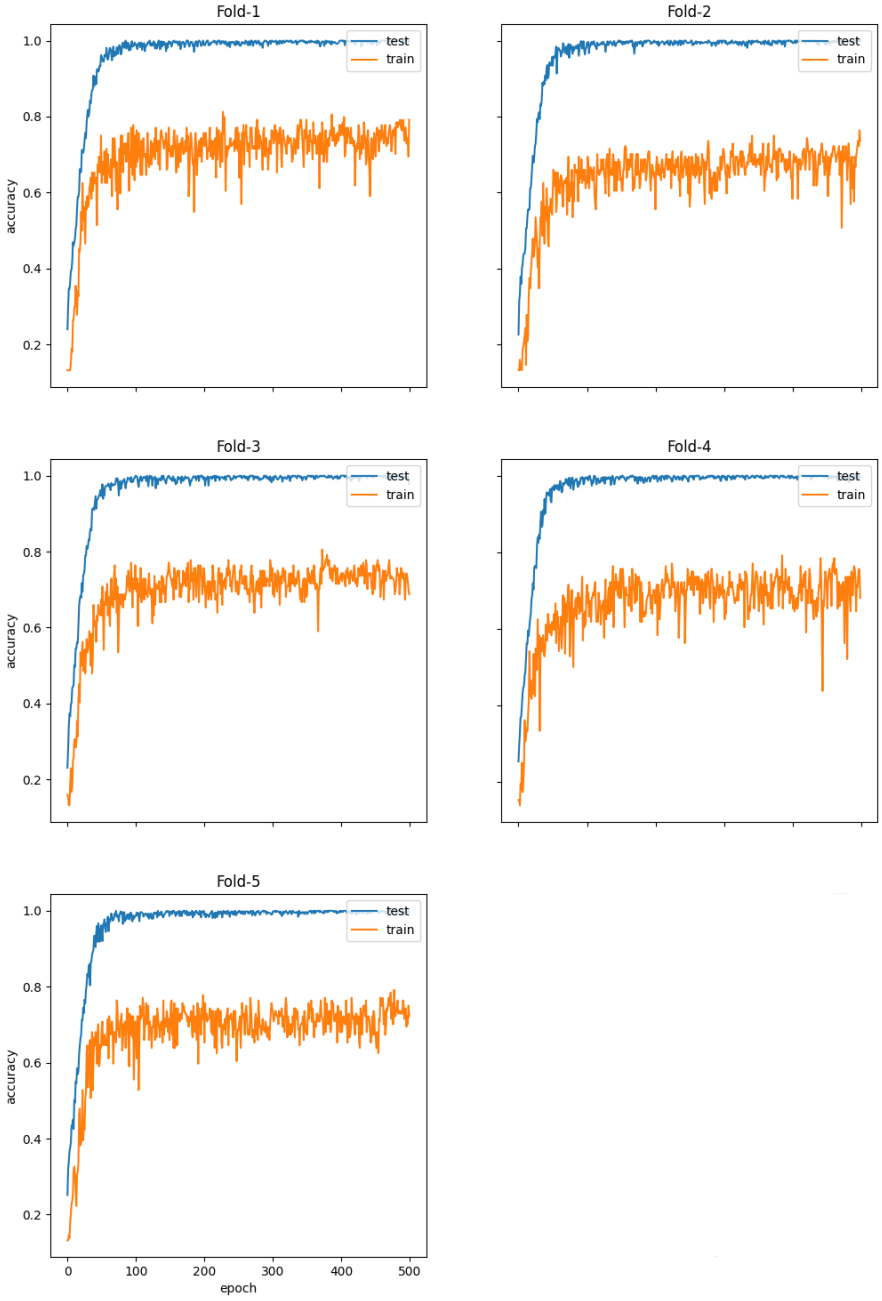

Figure 4.4: The validation scores of male actors (500 epochs).

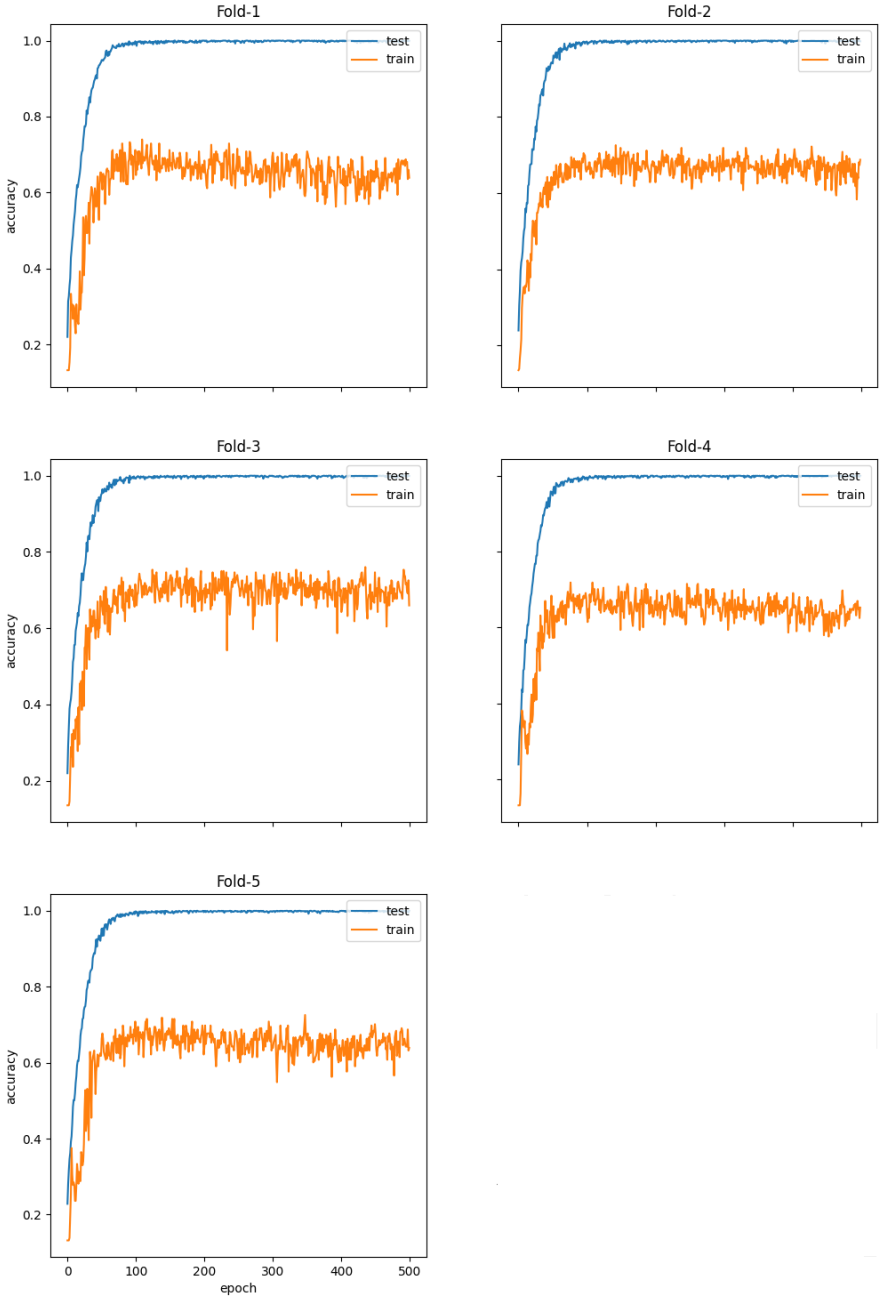

Figure 4.5: The validation scores of male and female actors combined (500 epochs).

CHAPTER 5

CONCLUSION AND FUTURE WORK

The use of machine learning models to identify emotions from speech data reliably is examined in the thesis on emotion recognition in speech. The study effectively recovers local and global emotion-related information from voice signals by utilizing deep learning approaches, such as Convolutional Neural Networks (CNN). The experimental findings demonstrate how well the model captures and analyzes spectrogram data to improve Speech Emotion Recognition (SER), yielding promising accuracy rates for a range of emotions.

The thesis highlights the increasing importance of emotion recognition technology in the digital age, where multimedia communication is crucial, in addition to its technological features. In addition to improving human-computer interaction, the capacity to identify and understand emotions from speech signals creates new opportunities for applications in a variety of domains, including social media analysis, mental health monitoring, and tailored user experiences. The study emphasizes how crucial emotional intelligence is to human-computer interactions and how emotion recognition technologies may influence how communication paradigms evolve in the future.

In summary, by showcasing the effectiveness of deep learning models in precisely recognising emotions from speech data, the thesis adds significant knowledge to the subject of Emotion Recognition in Speech. The experimental findings support the

suggested machine learning approach's capacity to increase recognition accuracy and extract significant emotion-related information. The study makes recommendations for future directions for investigation, such as examining the model's scalability and real-time applicability, growing the dataset for thorough emotion analysis, and fusing emotion recognition technology with other fields like natural language processing. In summary, the thesis emphasizes how important emotion recognition technology is to the advancement of digital-era communication and human-computer interaction.

For future work, it is suggested to explore the scalability and real-time applicability of the proposed model in practical settings. Additionally, further research could focus on enhancing the model's performance by incorporating more advanced machine learning techniques or by expanding the dataset to include a wider range of emotions and speech patterns. Moreover, investigating the integration of emotion recognition in speech with other technologies, such as natural language processing or sentiment analysis, could open up new avenues for research and application in fields like human-computer interaction, customer service, and mental health support.

Data augmentation can be used to improve the performance of SER as a future research. The augmentation technique is frequently employed, as seen by recent studies, and it has the potential to significantly improve the quality of the solutions. In addition, advanced feature selection techniques can be applied to choose the best subset of features to train our model, and to improve the accuracy of the SER model.

As further future work, it is possible to make use of global windowing and self-attention strategies in the SER transformer model. The RAVDESS database's emotion recognition ability can be greatly enhanced by using these techniques. Research reports in the recent literature demonstrates how enhanced attention mechanisms can

be used to improve the accuracy and resilience of emotion identification systems in practical settings.

REFERENCES

[1] M. Swain, A. Routray, and P. Kabisatpathy, "Databases, features and classifiers for speech emotion recognition: a review," *International Journal of Speech Technology*, vol. 21, pp. 93–120, 2018.

[2] H.M. Fayek, M. Lech, and L. Cavedon, "Evaluating deep learning architectures for speech emotion recognition," *Neural Networks*, vol. 92, pp. 60–68, 2017.

[3] P.M. Matthews and P. Jezzard, "Functional magnetic resonance imaging," *Journal of Neurology, Neurosurgery & Psychiatry*, vol. 75, no. 1, pp. 6–12, 2004.

[4] M. Ephratt, "Linguistic, paralinguistic and extralinguistic speech and silence," *Journal of pragmatics*, vol. 43, no. 9, pp. 2286–2307, 2011.

[5] B. Logan *et al.*, "Mel frequency cepstral coefficients for music modeling." In *Ismir*, vol. 270. Plymouth, MA, 2000, p. 11.

[6] J. Shen, R. Pang, R.J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions". *IEEE international conference on acoustics, speech and signal processing (ICASSP)*. In IEEE, 2018, pp. 4779–4783.

[7] G. Bernardes, D. Cocharro, M. Caetano, C. Guedes, and M.E. Davies, "A multilevel tonal interval space for modelling pitch relatedness and musical consonance," *Journal of New Music Research*, vol. 45, no. 4, pp. 281–294, 2016.

[8] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 32, no. 2, pp. 236–243, 1984.

[9] H. Patni, A. Jagtap, V. Bhoyar, and A. Gupta, "Speech emotion recognition using mfcc, gfcc, chromagram and rmse features". *8th International conference on signal processing and integrated networks (SPIN)*. In IEEE, 2021, pp. 892–897.

[10] M. El Ayadi, M.S. Kamel, and F. Karray, "Survey on speech emotion recognition: Features, classification schemes, and databases," *Pattern recognition*, vol. 44, no. 3, pp. 572–587, 2011.

[11] D. Issa, M.F. Demirci, and A. Yazici, "Speech emotion recognition with deep convolutional neural networks," *Biomed. Signal Process. Control.*, vol. 59, p. 101894, 2020.

[12] C.M. Lee, S. Yildirim, M. Bulut, A. Kazemzadeh, C. Busso, Z. Deng, S. Lee, and S. Narayanan, "Emotion recognition based on phoneme classes". *Interspeech 2004*. In ISCA, 10 2004.

[13] C. Dala, “A literature review on emotion recognition in speech,” *Researcher*, vol. 59, p. 101894, 2023.

[14] K. Han, D. Yu, and I. Tashev, “Speech emotion recognition using deep neural network and extreme learning machine,” *15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*, H. Li, H.M. Meng, B. Ma, E. Chng, and L. Xie, Eds. In ISCA, 2014, pp. 223–227.

[15] A.M. Badshah, J. Ahmad, N. Rahim, and S.W. Baik, “Speech emotion recognition from spectrograms with deep convolutional neural network”. *International conference on platform technology and service (PlatCon)*. In IEEE, 2017, pp. 1–5.

[16] S. Mittal, S. Agarwal, and M.J. Nigam, “Real time multiple face recognition: A deep learning approach”. *International Conference on Digital Medicine and Image Processing*, ser. In DMIP ’18. ACM, Nov. 2018.

[17] H.S. Bae, H.J. Lee, and S.G. Lee, “Voice recognition based on adaptive mfcc and deep learning,” *11th Conference on Industrial Electronics and Applications (ICIEA)*. In IEEE, Jun. 2016.

[18] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. In IEEE Computer Society, 2016, pp. 770–778.

[19] K.Y. Huang, C.H. Wu, Q.B. Hong, M.H. Su, and Y.H. Chen, “Speech emotion recognition using deep neural network considering verbal and nonverbal speech sounds”. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. In IEEE, May 2019.

[20] W. Lim, D. Jang, and T. Lee, “Speech emotion recognition using convolutional and recurrent neural networks”. *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. In IEEE, Dec. 2016.

[21] G. Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, M.A. Nicolaou, B. Schuller, and S. Zafeiriou, “Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network”. *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. In IEEE, Mar. 2016.

[22] S.R. Livingstone and F.A. Russo, “The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english,” *PLOS ONE*, vol. 13, no. 5, p. e0196391, May 2018.

[23] C. Busso, M. Bulut, C.C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J.N. Chang, S. Lee, and S.S. Narayanan, “Iemocap: interactive emotional dyadic motion capture database,” *Language Resources and Evaluation*, vol. 42, no. 4, pp. 335–359, Nov. 2008.

[24] F. Burkhardt, A. Paeschke, M. Rolfes, W.F. Sendlmeier, and B. Weiss, “A database of German emotional speech”. *Interspeech 2005*, 2005, pp. 1517–1520.

[25] J. Zhao, X. Mao, and L. Chen, “Speech emotion recognition using deep 1d and 2d cnn lstm networks,” *Biomedical Signal Processing and Control*, vol. 47, pp. 312–323, Jan. 2019.

[26] Y. Niu, D. Zou, Y. Niu, Z. He, and H. Tan, “Improvement on speech emotion recognition based on deep convolutional neural networks”. *International Conference on Computing and Artificial Intelligence*, ser. In ICCAI 2018. ACM, Mar. 2018.

[27] L. Tarantino, P.N. Garner, A. Lazaridis *et al.*, “Self-attention for speech emotion recognition”. *Interspeech*, 2019, pp. 2578–2582.

[28] F. Eyben, K.R. Scherer, B.W. Schuller, J. Sundberg, E. Andre, C. Busso, L.Y. Devillers, J. Epps, P. Laukka, S.S. Narayanan, and K.P. Truong, “The geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing,” *IEEE Transactions on Affective Computing*, vol. 7, no. 2, pp. 190–202, Apr. 2016.

[29] A. Triantafyllopoulos, G. Keren, J. Wagner, I. Steiner, and B. Schuller, “Towards robust speech emotion recognition using deep residual networks for speech enhancement”. *20TH Annual Conference of the International Speech Communication Association (Interspeech 2019)*, 2019.

[30] B. Schuller, S. Steidl, A. Batliner, J. Hirschberg, J.K. Burgoon, A. Baird, A. Elkins, Y. Zhang, E. Coutinho, and K. Evanini, “The interspeech 2016 computational paralinguistics challenge: Deception, sincerity & native language”. *17TH Annual Conference of the International Speech Communication Association (Interspeech 2016), Vols 1-5*, vol. 8. In ISCA, 2016, pp. 2001–2005.

[31] N. Weiskirchen, R. Bock, and A. Wendemuth, “Recognition of emotional speech with convolutional neural networks by means of spectral estimates”. *Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*. In IEEE, Oct. 2017.

[32] A. Chatziagapi, G. Paraskevopoulos, D. Sgouropoulos, G. Pantazopoulos, M. Nikandrou, T. Giannakopoulos, A. Katsamanis, A. Potamianos, and S. Narayanan, “Data augmentation using gans for speech emotion recognition”. *Interspeech 2019*. In ISCA, Sep. 2019.

[33] S. Demircan and H. Kahramanli, “Application of fuzzy c-means clustering algorithm to spectral features for emotion classification from speech,” *Neural Computing and Applications*, vol. 29, no. 8, pp. 59–66, Nov. 2016.

[34] S. Yoon, S. Byun, and K. Jung, “Multimodal speech emotion recognition using audio and text,” *IEEE Spoken Language Technology Workshop (SLT)*. In IEEE, Dec. 2018.

[35] Z. Huang, M. Dong, Q. Mao, and Y. Zhan, “Speech emotion recognition using cnn”. *22nd ACM international conference on Multimedia*, ser. In MM ’14. ACM, Nov. 2014.

[36] S. Wu, T.H. Falk, and W.Y. Chan, “Automatic speech emotion recognition using modulation spectral features,” *Speech Communication*, vol. 53, no. 5, pp. 768–785, May 2011.

[37] A.S. Lampropoulos and G.A. Tsihrintzis, “Evaluation of mpeg-7 descriptors for speech emotional recognition”. *Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. In IEEE, Jul. 2012.

[38] S.R. Livingstone and F.A. Russo, “The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english,” *PloS one*, vol. 13, no. 5, p. e0196391, 2018.

Appendix A

Implementation of the Proposed Models

Figure A.1 illustrates the libraries used for manipulating wave files, displaying wave features, and obtaining statistical information, all showcased using the matplotlib library. This visual representation highlights the tools and resources employed in the preprocessing and the analysis of audio data for emotion recognition tasks. By leveraging these libraries, researchers can extract relevant features from speech signals, visualize wave characteristics, and perform statistical analyses to enhance the understanding of emotional cues present in the audio data. Figure A.1 underscores the importance of using appropriate software libraries and visualization techniques to facilitate the effective processing and interpretation of speech data in emotion recognition systems.

The "import librosa.display" code imports the display module from the Librosa library. The librosa.display module provides functions for visualizing various audio-

```
import librosa
import librosa.display
import matplotlib as plt
from pathlib import Path
import matplotlib.pyplot as plt
import statsmodels.api as sm
from scipy.signal import find_peaks
import os
```

Figure A.1: Libraries used for manipulating wave files.

```
from pathlib import Path
audio_path = '/content/drive/MyDrive/Colab Notebooks/Data'
x , sr = librosa.load(audio_path)
```

Figure A.2: Path module import.

related information, such as waveforms, spectrograms, and chromagrams. It allows you to generate plots and visual representations of audio data to aid in analysis and understanding.

Librosa is a Python package for music and audio analysis. It provides tools to load and analyze audio files, extract various features from audio signals, and perform tasks such as *pitch estimation*, *beat tracking*, and more. librosa.load( ) method is used to load audio files.

Figure A.2 shows Path module import before setting the audio files' path and loading the main folder with the help of Librosa library returning the audio signals' folder path and sample rate.

Figure A.3 demonstrates the import declarations of various Python libraries for processing and manipulating data, creating mathematical representations, setting dataset paths, and building machine learning models. This visual depiction showcases the essential tools and steps involved in the development of machine learning algorithms for SER. By incorporating these libraries and functionalities, researchers can prepro-

```
import librosa
import librosa.display
import matplotlib as plt
from pathlib import Path
import matplotlib.pyplot as plt
import statsmodels.api as sm
from scipy.signal import find_peaks
import numpy as np
import tensorflow as tf
from matplotlib.pyplot import specgram
import pandas as pd
from sklearn.metrics import confusion_matrix
from keras.preprocessing import sequence
from keras.models import Sequential
from keras.layers import Dense, Embedding, Activation
from keras.layers import LSTM
from keras.preprocessing.text import Tokenizer
#from keras.preprocessing.sequence import pad_sequences
from keras.utils.data_utils import pad_sequences
from tensorflow.keras.utils import to_categorical
from keras.layers import Input, Flatten, Dropout#, Activation
from keras.layers import Conv1D, MaxPooling1D, AveragePooling1D
from keras.models import Model
from keras.callbacks import ModelCheckpoint
from tensorflow.keras.layers import BatchNormalization
from keras.utils import np_utils
from sklearn.preprocessing import LabelEncoder
from keras import optimizers
from sklearn.model_selection import StratifiedKFold, KFold
from sklearn.metrics import accuracy_score
```

Figure A.3: Python libraries for processing data, doing mathematics, building OS paths and ML models.

cess audio data, extract relevant features, construct mathematical representations for modeling, and train machine learning models to classify emotions in speech signals. Figure A.3 emphasizes the significance of leveraging appropriate libraries and techniques to streamline the workflow and enhance the efficiency of building emotion recognition systems based on machine learning approaches.

NumPy is a powerful numerical computation library for Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. NumPy is a fundamental package for scientific computing in Python and is widely used in various fields, including machine learning, data science, signal processing.

statsmodels is a Python library that provides classes and functions for estimating and

testing various statistical models. It includes functionalities for regression analysis, time-series analysis, hypothesis testing.

The statement "import statsmodels.api as sm" is used to import the "statsmodels" library in Python and gives it the alias "sm". This allows referring to the library using the shorter name "sm" in code.

The statement "from scipy.signal import find_peaks" imports the find_peaks function from the scipy.signal module in Python. The function is part of the SciPy library, which is a collection of scientific computing tools built on top of NumPy. The find_peaks function is actually designed for detecting peaks (local maxima) in 1-D arrays.

The "import os" statement in Python brings OS module support to sequential code. The OS module provides a way of interacting with the operating system, allowing to perform various operating system-related tasks such as file and directory manipulation, environment variable access.

Pandas is a powerful data manipulation and analysis library for Python, and using "pd" as an alias is a widely adopted convention. Mainly, the purpose for import was the use of dataframes and data analysis, processing.

The *piptrack* method in the pitch library is a valuable tool for pitch tracking in audio signals, particularly in the context of speech and music analysis. This method utilizes advanced signal processing techniques to accurately estimate the pitch of a sound signal over time, providing insights into the fundamental frequency variations present in the audio data. By employing sophisticated algorithms, piptrack can identify and track the pitch trajectory of a signal, enabling researchers and practitioners to analyze

```
@staticmethod
def getActors(AP):
    directory = AP
    read_path = ""
    x = []

    for d in os.listdir(AP):
     if os.path.isdir(os.path.join(AP, d)):
        if(d[0]=="A"):
         x.append(d)
    x.sort()
    return x
```

Figure A.4: Finding the list of all male and female actors using getActors function

pitch-related characteristics such as intonation, melody, and vocal expression. The method's ability to robustly track pitch variations in complex audio signals makes it a crucial component in tasks like speech recognition, music transcription, and emotion detection from speech, offering a deeper understanding of the acoustic properties of sound signals. Overall, the piptrack method serves as a powerful tool for extracting pitch information from audio data, facilitating in-depth analysis and interpretation of pitch-related features in various applications within the field of audio signal processing and analysis.

Mainly, pitch library was imported for its piptrack method to be employed.

To give a summary of methods used in this study, the following code segments in Figure A.4 and explanations are provided.

In the getActors method given in Figure A.5, dataset actor names starting with the

```
@staticmethod
def getSoundFilesList(adir, audio_path):
    X = []

    fdir = audio_path + "/" + adir
    for r, d, f in os.walk(fdir):
     for file in f:
         if file.endswith(".wav"):
             print(file)
             X.append(file)

    #print(X)
    return X
```

Figure A.5: getSoundFilesList method

letter "A" are appended to a list and sorted with the method returning the resulting list. (Actor1, Actor2, etc.)

The sound files path, "fdir", is built up by concatenating audio path and the corresponding Actors directory. The method then checks for files with ".wav" extension in that directory. A list of sound files within the current Actors directory is built and returned by this method.

Using the $mfcc$ method, given the audio signal and sample rate, Mel-freq. Cepstral Coefficients are calculated.

Figure A.7 in the thesis presents the "extractFeatures" method, excluding MFCC extraction which was shown as implemented. This code outlines the process of extracting relevant features from audio data for SER. This method is crucial in the feature extraction stage of the machine learning pipeline, where acoustic features indicative of

```
def displayMFCC(mfc, sr):

    #print(mfc.shape)

    #Displaying  the MFCCs:

    librosa.display.specshow(mfc, sr = sr, y_axis='linear', x_axis='time')


@staticmethod

def calcMFCCs(y, sr):

    mfccs=librosa.feature.mfcc(y=y, sr=sr, n_mfcc=40)

    mfccScaled = np.mean(mfccs.T, axis=0)

    return mfccScaled
```

Figure A.6: displayMFCC and calcMFCCs methods

```
@staticmethod
def extractFeatures(audio_path):



    y, sr = librosa.load(audio_path)


    mfccs = []
    mfccs = calcMFCCs(y, sr)
    SF["MFCC"] = mfccs

    spec = np.abs(librosa.stft(y, n_fft=1012))
    chroma_cq = librosa.feature.chroma_cqt(y=y, sr=sr)
    cqt = np.mean(chroma_cq,axis=1)
    SF["CHROMA_CQT"] = cqt

    mel = np.mean(librosa.feature.melspectrogram(y=y, sr=sr,n_mels=128).T, axis=0)
    SF["MEL_SPECTROGRAM"] = mel

    contrast = np.mean(librosa.feature.spectral_contrast(S=spec, sr=sr).T, axis=0)
    SF["CONTRAST"] = contrast

    X = librosa.effects.harmonic(y)
    tonnetz = np.mean(librosa.feature.tonnetz(y=X, sr=sr).T, axis=0)
    SF["TONNETZ"] = tonnetz

    return mfccs, cqt, mel, contrast, tonnetz
```

Figure A.7: Extracting relevant features from audio data for SER using *extractFeatures* method

different emotions are computed and utilized for training emotion classification models. By detailing the steps involved in the extractFeatures method, researchers can gain insights into the specific feature extraction techniques applied to the audio data, such as MFCCs, pitch values, spectral features, or other relevant parameters. This figure underscores the importance of feature extraction in capturing the discriminative characteristics of speech signals that aid in accurately recognizing and classifying emotions, thereby contributing to the overall effectiveness of the emotion recognition system.

In this thesis, LIBROSA feature extraction library was used in order to produce the following features: MFCCs, CHROMAGRAM, MEL_SPECTROGRAM, CONTRAST, TONNETZ

Figure A.8 code extracts features not employed in training showing the extraction of additional features from audio signals, including linear prediction coefficients, zero-crossings count, autocorrelation factors, signal peaks, and pitch values. Although these features were not utilized in training and testing the dataset, the figure highlights the comprehensive approach taken to explore various acoustic attributes that could potentially contribute to SER. By calculating metrics such as median pitch, pitch range, and average pitch frequency, researchers can delve deeper into the acoustic properties of speech signals and investigate additional parameters that may enhance the understanding and classification of emotional states in audio data. This figure underscores the importance of considering a wide range of features in the feature extraction process to capture diverse aspects of speech signals for more robust emotion recognition systems.

Linear prediction coefficients, zero-crossings count of the signal were extracted. Au-

```
lpc_result = librosa.lpc(y, order=15)
#SF[actorName][filename]["LPC"] = lpc_result
#print(lpc_result)


pitch_component = []
n0 = 0 #
n1 = 5000
zero_crossings = librosa.zero_crossings(y[n0:n1], pad=False)
#print(len(zero_crossings))
#SF[actorName][filename]["ZC_SUM"] = sum(zero_crossings)

auto = sm.tsa.acf(y) ################??????
peaks = find_peaks(auto)[0] # Find peaks of the autocorrelation
if np.any(peaks):
 lag = peaks[0] # Choose the first peak as our pitch component lag
 pitch = sr / lag
 #print(pitch)
 #SF[actorName][filename]["PITCH_COMPONENT"] = pitch

chroma = librosa.feature.chroma_stft(S=spec, sr=sr)

#img = librosa.display.specshow(chroma, y_axis = 'chroma', x_axis='time')
#fig, ax = plt.subplots(figsize=(20,10))
#fig.colorbar(img, ax)
#plt.show()

#SF[actorName][filename]["CHROMA"] = chroma
#chroma_arr.append(chroma)

pitches, magnitudes = librosa.core.pitch.piptrack(y=y, sr=sr)
#print(pitches.squeeze())
max_indexes = np.argmax(magnitudes, axis=0)
pitches = pitches[max_indexes, range(magnitudes.shape[1])]
l=(int)((len(pitches)-1)/2)
```

Figure A.8: Other features that could be used for SER.

```
X= pd.DataFrame()
cc=0
for i in sub_dir:
    filename = os.listdir(main_dir + '/' + i)

    for f in filename:
      if f.endswith('wav'):
        part = f.split('.')[0].split('-')
        print(part)
        emotion.append(int(part[2]))
        actor.append(int(part[6]))

        file_path.append(main_dir + i + '/' + f)
        file_pathShort.append(i + '/' + f)
        SF= {}
        AL = getActors(main_dir + '/' + i)



        ASFL = getSoundFilesList(i, main_dir)
        a,b,c,d,e=extractFeatures(main_dir + '/' + i + '/' + f)

        tot= []
        for x in a:tot.append(x)
        for x in b:tot.append(x)
        for x in c:tot.append(x)
        for x in d:tot.append(x)
        for x in e:tot.append(x)

        #X4[cc]=pd.DataFrame([tot])
        X[cc]=np.asarray(tot)
        cc+=1;
        #print(tot)
```

Figure A.9: Building the dataframe using extracted features from audio data

tocorrelation factors for the signals were figured out using the "acf" method, signal peaks were detected and these two givens were employed to calculate pitch values. This can be an alternative to automated pitch value extraction methods.

Figure A.9 illustrates the process of building a dataframe using extracted features from audio data for SER. The dataframe construction involves organizing and structuring the extracted features, such as MFCCs, CQT, mel, contrast, and tonnetz, into a tabular format suitable for machine learning analysis. By creating a structured dataframe, researchers can effectively manage and manipulate the feature data, facilitating fur-

```
import tensorflow as tf
from matplotlib.pyplot import specgram
from sklearn.metrics import confusion_matrix
from keras.preprocessing import sequence
from keras.models import Sequential
from keras.layers import Dense, Embedding, Activation
from keras.layers import LSTM
from keras.preprocessing.text import Tokenizer
from keras.utils.data_utils import pad_sequences
from tensorflow.keras.utils import to_categorical
from keras.layers import Input, Flatten, Dropout#, Activation
from keras.layers import Conv1D, MaxPooling1D, AveragePooling1D
from keras.models import Model
from keras.callbacks import ModelCheckpoint
from tensorflow.keras.layers import BatchNormalization
```

Figure A.10: Reading in the dataframe

ther preprocessing steps and model training. Additionally, the figure highlights the importance of data organization and preparation in the machine learning pipeline, as the dataframe serves as a foundational component for training emotion classification models. This step underscores the significance of data handling and structuring in ensuring the efficiency and accuracy of the emotion recognition system.

Necessary module imports before the upcoming steps as seen in A.10 must be carried out.

(This page has been intentionally left blank.)