BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER SYSTEMS
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

DEVELOPMENT OF AUTOMATED EMOTION RECOGNITION SYSTEM THROUGH VOICE USING PYTHON

VÝVOJ AUTOMATIZOVANÉHO SYSTÉMU ROZPOZNÁVÁNÍ EMOCÍ POMOCÍ HLASU POMOCÍ PYTHONU

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR — **TEREZA MAGERKOVÁ**
AUTOR PRÁCE

SUPERVISOR — **YASIR HUSSAIN**
VEDOUCÍ PRÁCE

BRNO 2024

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Bachelor's Thesis Assignment

153453

Institut:	Department of Computer Systems (DCSY)
Student:	**Magerková Tereza**
Programme:	Information Technology
Title:	**Development of Automated Emotion Recognition System through Voice using Python**
Category:	Speech and Natural Language Processing
Academic year:	2023/24

Assignment:

1. Study human emotions and their effects on the human voice.
2. Identify the challenges and limitations of existing methods through a literature review.
3. Design a machine learning (including deep learning) model for automated detection of human emotion from the human voice.
4. Implement the machine learning model in Python for emotion detection from the human voice.
5. Test and validate the emotion detection model on publicly available voice dataset(s).
6. Conduct critical analysis and discuss achieved results and their contribution.

Literature:

- Based on the supervisor's recommendation.

Requirements for the semestral defence:

- Fulfillment of Items 1 to 3 of the assignment.

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

Supervisor:	**Hussain Yasir**
Head of Department:	Sekanina Lukáš, prof. Ing., Ph.D.
Beginning of work:	1.11.2023
Submission deadline:	9.5.2024
Approval date:	30.10.2023

Abstract

This work presents an in-depth investigation into the design and implementation of deep learning models for speech emotion recognition. It proposes a model based on a comprehensive review of existing techniques from the field. The model is trained and tested on large-scale emotion-labeled speech datasets. Experimental evaluations are conducted to assess the performance of the model in terms of accuracy, robustness, and generalization.

Abstrakt

Táto práca do hĺbky skúma návrh a implementáciu modelov hlbokého učenia na rozpoznávanie emócií z reči. Navrhuje model založený na komplexnom prehľade existujúcich techník z tejto oblasti. Model je trénovaný a testovaný na rozsiahlych sadách rečových dát označených emóciami. Vykonané experimentálne hodnotenia majú za cieľ posúdiť výkonnosť modelu z hľadiska presnosti, robustnosti a schopnosti zovšobecňovat rozpoznávacie schopnosti modelu.

Keywords

Machine Learning, Features, Python, Voice, Emotions, Deep Learning

Klíčová slova

Strojové učenie, Funkcie, Python, Hlas, Emócie, Hlboké učenie

Reference

MAGERKOVÁ, Tereza. *Development of Automated Emotion Recognition System through Voice using Python.* Brno, 2024. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Yasir Hussain

Rozšířený abstrakt

Táto práca sa zaoberá tvorbou systému na rozpoznávanie emócií pomocou hlbokého učenia, obsahuje popis a porovnanie rôznych prístupov k problematike, pochádzajúcich z analýzy súčasného stavu oblasti rozpoznávania emócií. Dátové sady použité na učenie sú EmoDB a RAVDESS, obe nahrané profesionálnymi hercami v nemeckom a anglickom jazyku.

Cieľom práce bolo navrhnúť systém, ktorý bude schopný detekovať emócie na základe vlastností zvukového signálu vedome alebo nevedome spôsobených emóciou. Súčasné riešenia v oblasti rozpoznávania emócií z hlasu využívajú vlastnosti zachytávajúce buď princíp tvorby reči (lineárne prediktívne kepstrálne koeficienty), alebo prijímania zvuku (mel-frekvenčné kepstrálne koeficienty). Algoritmy učenia často stavajú na hlbokých neurónových sieťach, v rôznych kombináciách s metódami spracovania dát. Typy sietí ako konvolučná neurónová sieť alebo rekurentná neurónová sieť, prípadne ich kombinácia sú často využívanými blokmi v existujúcich systémoch podobného charakteru.

Navrhnutý systém je postavený na konvolučnej neurónovej sieti, ktorá na vstup dostáva vybranú sadu mel-frekvenčných kepstrálnych koeficientov. Detekcia prebieha v troch fázach: spracovanie dát, extrakcia a selekcia vlastností a nakoniec učenie siete za pomoci spracovaných dát výsledkom ktorého je finálna klasifikácia. Testovanie výsledkov prebieha na základe predikcií modelu.

Dáta sú spracované odstránením tichých pasáží a normalizáciou. Tento krok napomáha presnejšej extrakcii koeficientov. Extrakcia a selekcia koeficientov slúži na zúženie vlastností vstupu do siete, umožňuje presnejšiu kontrolu vstupných dát a pomerne nízku komplexitu modelu a vďaka tomu aj nižšiu výpočtovú náročnosť. Konvolučná sieť pracuje s vlastnosťami spektrálnej a časovej domény signálu. Výsledkom je pravdepodobnostná hodnota príslušnosti vzorky do jednej z emočných tried.

Experimenty boli vykonané oddelene na oboch sadách. Každá sada je rozdelená na trénovaciu, testovaciu a validačnú časť. Sady sú rozdelené s uvážením rozdelenia tried emócií a reflektujú distribúciu v pôvodnej sade. Presnosť výsledkov trénovania je vyhodnocovaná po každej trénovacej epoche a po ukončení všetkých epoch je najlepší výsledok podrobnejšie vyhodnotený.

Ďalší možný smer tejto práce by mohlo byť otestovanie modelu na väčšej testovacej sade s reálnymi emóciami alebo podrobnejšie skúmanie vplyvu vstupných parametrov na výsledné predikcie.

Development of Automated Emotion Recognition System through Voice using Python

Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of Mr. Yasir Hussain. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

.....
Tereza Magerková
May 7, 2024

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Technology around us is much more prevalent today than it used to be only a few years ago, fundamentally changing how we communicate. Even when we interact with our devices daily, these interactions can feel like there is always something missing to feel truly human-like. To accommodate this requirement, the machine would have to be able to perceive the nature and circumstances of the communication and have the capacity to adjust the response accordingly.

Human voice as an effective way of carrying information, stands out as being our primary and most natural communication tool. Unlike various physical signals and other non-verbal forms of communication, speech is readily and easily producible and easy to collect. However, the problem lies in the fact that we heavily rely on non-verbal components of communication (such as emotions) to convey the full extent of our ideas and intentions.

In response, Speech Emotion Recognition (SER) has received interest in recent years. SER uses machine learning principles, allowing us to build models capable of identifying and categorizing emotions from speech. This can greatly affect many applications from AI assistants that could comprehend and respond to users' emotional states, to call centers enhancing customer satisfaction. Key challenges of SER lie in the vast variability of emotions and their expressions across individuals in various situations. As technology continues advancing, the pursuit of natural, context-aware man-to-machine interactions remains in demand.

1.1 Ethics and Motivation

This thesis examines the current approaches to SER and works with data used to develop an SER. There are some ethical dilemmas concerning this field. SER models can greatly improve critical infrastructures, automatizing the process and removing human error. With all these positive aspects come negatives. Data used to train these models have to be ethically sourced with consent and used for beneficial and not harmful purposes. To date, there is legislation in place to keep consumers from such exploitation and regulate the risks of deploying SER systems in real-world applications.

Chapter 2

Speech Emotion Recognition

Speech emotion recognition builds upon the principle that human emotions can be conveyed through voice. SER systems focus on correctly identifying and classifying correct features in human voice that are capable of conveying most critical characteristics [34]. The general structure of the system is depicted in Figure 2.1. In order to understand the

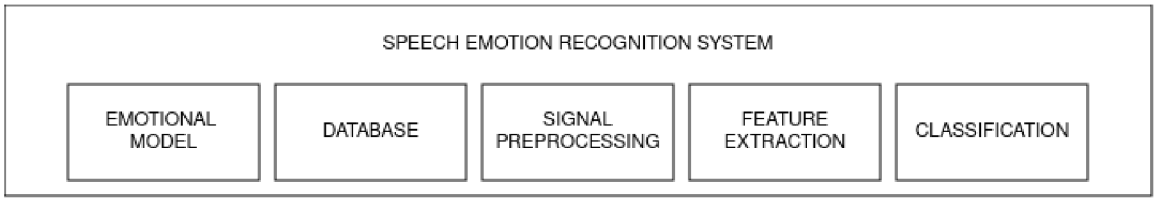


Figure 2.1: The basic structure of a speech emotion recognition system

whole system, it is necessary to first understand areas and challenges presented by each part. This chapter provides theoretical background to the field of SER. It describes the psychological standpoint of defining and categorizing emotions from the view of continuous and discrete models, workings of speech processing, and the use of speech features. Further, it approaches the classification methods used in SER and the paradigms of deep learning used to construct a robust and functional model.

2.1 Emotions

This section provides a fundamental psychological insight into the term emotion, the various attempts to define and understand it from various standpoints. It mainly introduces the idea of grouping emotions into models allowing for more accurate and strict descriptions.

2.1.1 Definition

An emotion can generally be described as a state of belief that results in changes both psychological and in turn physiological that convey one's state of thought [28]. Emotions fascinate research since Darwin [1] and there are different schools of psychology that produce many theories of what emotions are. We can group these theories into three main groups: physiological, neurological and cognitive.

Physiological theory proposes that human body is responsible for the creation of emotion. The James-Lange Theory of Emotions [9] suggests that emotions are a product of

physiological responses to stimuli in the environment. the Environmental stimuli trigger a response of the body and the bodily response in turn triggers a response of the brain which we call emotion. This means that we feel emotions because of the physical changes that occur in response to stimuli. For example, we feel afraid because we tremble and our heart races, rather than trembling and having a racing heart because we experience fear.

Neurological standpoint is that emotions come from activation within brain. This is described by the Facial-Feedback Theory of Emotion [10]. It suggests that facial expressions can influence and even regulate emotional experiences. According to this theory, the act of forming a facial expression, such as smiling or frowning, can trigger physiological and emotional responses associated with that expression our facial muscles send signals to the brain, which then interprets these signals as emotions. For example, if you force yourself to smile, it may lead to feelings of happiness or positivity, even if you weren't feeling that way initially. This theory emphasizes the bidirectional relationship between facial expressions and emotions, suggesting that our expressions not only reflect our emotions but also have the power to shape them.

Lastly, **cognitive theories** propose that thoughts and other mental activity play a role in forming emotions. The Cognitive Appraisal Theory [21] suggests that our emotions are determined by our cognitive appraisal or evaluation of events. When we encounter a stimulus or event, we subconsciously evaluate it based on its relevance to our goals, beliefs, and well-being. The appraisal process comes in two steps. The primary appraisal is the initial assessment of whether the event is positive, negative, or irrelevant. If it's seen as positive or relevant, it can lead to positive emotions. If it's perceived as negative, it can lead to negative emotions like fear, anger, or sadness. In the secondary stage we evaluate our ability to cope with or manage the situation. This assessment influences the intensity and type of emotion we experience. This produces the intensity of the emotion. In the case of positive emotions, it can escalate them to pride or relief. These appraisals trigger emotions and our reactions to them.

2.1.2 Categorization

All theories described emotions as some inner process. When we want to signify emotion on a more objective level, we need a model that can describe it. There are two unique approaches to modeling emotions: the discrete and the dimensional model.

Discrete Model

The discrete model divides emotions into disjunct categories. When choosing an emotion we can simply choose from predefined set of basic categories which one fits best. This model allows us to easily label emotions based on class tags. These are in most cases six: anger, disgust, fear, joy, sadness, and surprise or some variations or additions that better reflect the application of the model (some variations are listed in Table 2.1). Each of the classes is defined by a specific set of features that describe the circumstances leading to emotion and the resulting reactions [28].

The main drawback is that subjects can experience a wider range of emotions than presented. When choosing a response from predefined groups they may encounter a need to select and in doing so not identify the label themselves [28].

Table 2.1: Proposed discrete categories of emotions

Izard	**Ekman**	**Plutchik**
enjoyment	happiness	joy
sadness	sadness	sorrow
fear	fear	fear
anger	anger	anger
disgust	disgust	disgust
surprise	surprise	surprise
interest	acceptance	
shame	anticipation	
shyness		
guilt		

Dimensional Model

The dimensional or attribute models utilize individual variables used to describe emotion along multiple dimensions defined by these variables [34]. These models often use two dimensions: valence and arousal or add a third dimension as depicted in Figure 2.2. Emotion is then defined as a point in a defined space. These models are therefore capable of capturing fine differences and allow to observe a similarity of emotions [28] contrary to broad categories of the discrete model.

The circumplex model proposed by Russell [29] utilizes two primary dimensions: valence and arousal. The valence axis refers to whether an emotion is perceived as positive or negative. Emotions with positive valence, such as happiness or excitement, are located on one side of the circumplex, while emotions with negative valence are located on the opposite side. The arousal axis reflects the level of physiological activation or intensity associated with an emotion.

The PAD (Pleasure-Arousal-Dominance) model chooses three dimensions. First, is Pleasure-Displeasure defining whether an emotion is positive or negative. Second is the Arousal-Non-arousal axis indicating psychological activation and alertness. And third is the Dominance-Submissiveness axis describing whether the subject feels in control of the emotion. Result is then evaluated by a PAD score defined by emotion scales [23].

2.2 Data Sourcing

Datasets provide collections of data that has been collected, organized and labeled. The choice of a dataset and the quality of the data in it affects the success of the recognition process [4]. A good dataset usable for SER is required to have diversity of speakers namely their genders, ages and requires a balanced number of recordings of each class of emotion per speaker to prevent data imbalance in the dataset.

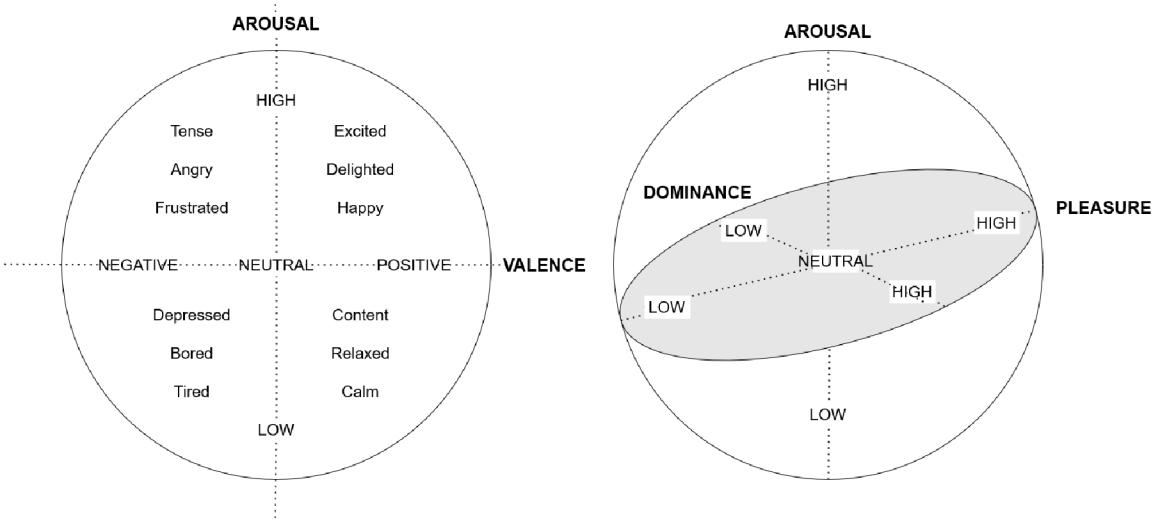


Figure 2.2: Valence-Arousal and PAD model

Table 2.2: Emotion models - overview

	Discrete model	**Dimensional model**
Emotion definitions	Disjunct categories	Points in multi-dimensional space
Real-life applications	Natural way of giving names to emotions	More descriptive of feelings
Labeling	Intuitive creation of classes	Classes need to be constructed from variables

Databases often use one of the emotion models 2.1.2 to signify the emotion captured in a recording. The labels contain information about speaker, the contents of a sample and most importantly the collected emotion. When choosing we then must take into account not only the quality of data but also the range of emotions it can provide and whether that range is suitable or necessary for the application.

Emotion recognition datasets can be divided into speech datasets (audio recordings, text) or visual datasets (facial expression images, video recordings).

Datasets can be sourced in various ways. The main three include acted, elicited and naturally sourced recordings. When sourcing the data it is important that the actors or other participants consent to the data being collected.

Acted Speech Datasets

Are recorded by professional actors in low-noise environments. These datasets can regulate and plan the recorded emotion classes beforehand and recruit actors to make the data diverse. Emotions in acted speech are greatly exaggerated making models trained on such

databases less successful in detecting real-life emotions [4]. This unwanted effect can be reduced by hiring semi-professional actors [12]

Elicited Speech Datasets

We can produce a viable dataset by placing a recording device in an environment where emotions are artificially induced. This method does not create real emotions but gets close to them and provides a wider range of categories. Usually, the stimuli are presented in forms of videos, images, stories, music clips, or other media types that are designed to induce an emotion of varying intensity. A study from 2005 [17] proposed to use events in computer games to induce a more realistic emotional reaction from players.

Natural Speech Datasets

Lastly, these are collections of enormous numbers of recordings obtained from various call-center recordings, talk shows, podcasts, or public conversations. Data sourced this way usually contains more noise and the emotions are not as consistent as in elicited or acted datasets but it has a limited effect. A famous example is the collection of radio broadcasts happening during the Hindenburg crash [12]. Obtaining such spontaneous speech can be ethically and legally challenging.

2.3 Sound and Speech

Sound and speech play are integral components of human communication. Speech is a specialized form of sound exclusive to humans characterized by the production of vocal sounds through the coordination of respiratory, phonatory, and articulatory systems. It carries emotional information through various acoustic features such as pitch, intensity, rhythm, and spectral characteristics. Different emotions are associated with distinct patterns in these acoustic features, making speech an effective medium for conveying emotions.

This section provides an overview of processes used to extract information from speech signals and overcome challenges such as variability in emotional expression, cultural differences, and noisy environments.

2.3.1 Preprocessing

When working with audio signals we have to take into account that the data can contain unwanted components like background noise, inconsistencies in energy levels or variations of voice characteristics. All of these aspects can have an effect on extracting relevant features especially when working with real-world speech [20]. The speech signal is usually preprocessed in steps indicated in Figure 2.3.

Voice Activity Detection (VAD)

Voice activity detection is used to separate parts where voiced speech can be detected from silent (unvoiced) parts. The process divides signal into short frames and determining the presence of activity withing these frames. When determining the activity we can use two categories of features: time-domain and frequency-domain [20].

The time-domain features are used due to their simplicity but degrade in a noisy signal [20]. One of these features is Zero-Crossing Rate (ZCR). This method utilizes the way how

Table 2.3: Brief description of selected databases

	Language	*Size*	*Emotions*	*Type*
IEMOCAP	English	5 531 utterances	Anger, Happiness, Excitement, Sadness, Frustration, Fear, Surprise, Neutral	Elicited
BAUM-1s	Turkish	1 222 utterances	Joy, Anger, Sadness, Disgust, Fear, Surprise, Boredom, Contempt	Natural
EMO-DB	German	535 utterances	Neutral, Anger, Sadness, Fear, Boredom, Happiness, Disgust	Acted
RAVDESS	English	1056 utterances	Neutral, Anger, Sadness, Fear, Happiness, Disgust, Surprised, Calm	Acted
MSP-Podcast (version 1.7)	Mixed	62 140 utterances	Anger, Happiness, Sadness, Disgust, Surprise, Fear, Contempt, Neutral	Natural

vocal tract constricts and opens during voiced speech [7]. When producing voiced speech, the vocal tract produces a periodic flow which shows a low zero-crossing count, whereas unvoiced speech is produced by constricting the airflow and shows higher zero-crossing count. The detection of the crossings is shown in Figure 2.4.

Noise Reduction

When any sound signal is being used to obtain information, it cannot be corrupted by any background or ambient noise. Ambient noise is any signal not being monitored [16] and can affect the later stages of recognition. By removing noise, we are left with clean speech.

Framing

Framing or segmentation is a process of apportioning the continuous voice signal into fixed-length segments. Speech is considered an unstable signal, however can remain stable for a sufficiently short period, such as 20 to 30 milliseconds [4]. In this short time, we can examine local and quasi-stationary features.

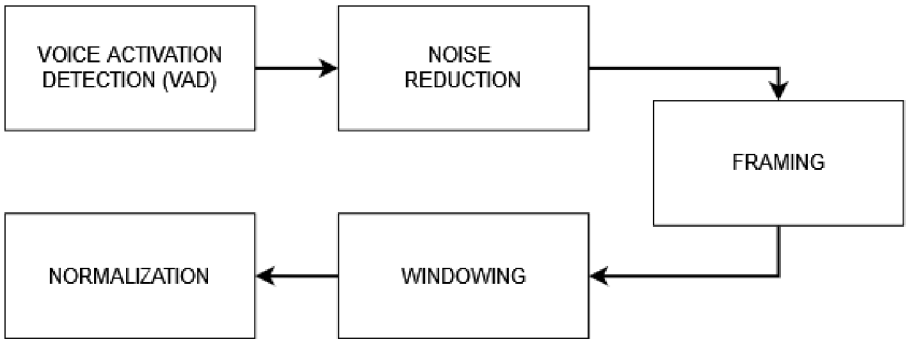


Figure 2.3: Preprocessing steps

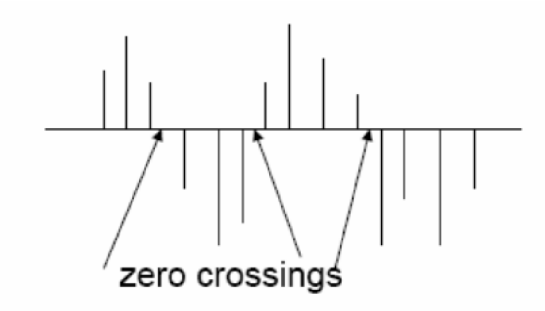


Figure 2.4: Definition of zero-crossing rate [7]

Windowing

After framing the signal, we are left with discontinuities in the beginning and the end of the frame. This can be solved by multiplying the signal with a window function. Window function (e.g. Hamming window or Hanning window) is used to smooth out values on both ends of a signal while leaving the middle part preserved. The smoothed edges can mean information loss, however, the relation and information between the frames can be retained by deliberately overlapping 30 to 50 percent of these segments [4].

Normalization

Feature normalization is done to reduce speaker and recording variability without losing the discriminative strength of the features. [4]. It can be performed on either the whole recording or applied to the framed signal. The most common normalization method is the z-normalization which can be calculated using equation 2.1 for signal x if mean μ and standard deviation σ are known.

$$z = \frac{x - \mu}{\sigma} \tag{2.1}$$

2.3.2 Features

Features represent different information sourced from audio signals. Most acoustic features used for SER can be separated as prosodic and spectral (vocal tract) [36]. Prosody focuses on the temporal and rhythmic aspects of speech that align with human perception of

emotion, spectral features describe frequency content and distribution of energy, providing a different perspective. The combination of more types of features often results in more robust and accurate models [12].

Prosodic Features

Prosody provides linguistic naturalness to speech through intonation, stress, and rhythm. These cues are conveyed using mainly three acoustic parameters: pitch, energy, and duration [25]. Prosodic features are applied by the speaker and can be detected in units like syllables, words, or sentences - units detectable in larger portions of recordings. Literature suggests prosody to be a high correlate of emotion. Features derived from statistical methods are important sources for discriminating emotions. Around 50% of average emotion recognition performance is reported using discriminant analysis [26].

Spectral Features

Spectral features in the context of speech emotion recognition involve analyzing the frequency content of the speech signal. These features provide information about the distribution of energy across different frequency bands. Generally, we have a 20 to 30 milliseconds long segment used to extract these features. Vocal tract characteristics are well reflected in frequency domain analysis of speech signals. The Fourier transform of each frame gives a short time spectrum where features like formants, bandwidths, spectral energy, and their slopes can be observed [19].

Mel-frequency Cepstral Coefficients (MFCCs)

Are a set of coefficients commonly used in speech and audio processing for representing the short-term power spectrum of the speech signal.

The MFCCs are widely used in SER due to their ability to mimic the workings of the human auditory system which does not follow a linear scale when it comes to perception of sound frequency contents for speech signal. Therefore, for each frequency f measured in Hz a subjective pitch is measured (Equation 2.2) on the Mel scale [15]

$$f_{met} = 2529 \log_{10}(1 + \frac{f}{700}) \tag{2.2}$$

To obtain MFCC, speech signals are divided into segments (described in 2.3.1) each segment is converted into the frequency domain using a short-time discrete Fourier transform. Several sub-band energies are calculated using a Mel filter bank and the logarithm of those sub-bands is calculated. The inverse Fourier transform is applied to obtain MFCC [4].

Linear Prediction Cepstral Coefficients (LPCCs)

Another set of coefficients used in speech and audio processing are the LPCCs. They are derived from Linear Prediction Coefficients (LPC), based on the speech production model where the characteristic of the vocal tract can be modeled by an all-pole filter. LPCCs same as LPCs have the disadvantage of approximating speech linearly on all frequencies, which is inconsistent with how human hearing works [35].

Table 2.4: Spectral features - overview

	LPCC	**MFCC**
Extraction Process	Linear prediction analysis followed by cepstral analysis	Mel-frequency filter-bank analysis followed by cepstral analysis.
Frequency Domain	Primarily captures vocal tract characteristics	Captures spectral characteristics, emphasizing perceptual loudness
Sensitivity to Noise	More sensitive to noise due to detailed vocal tract modeling	Relatively robust to noise due to perceptual loudness focus
Biological Inspiration	Inspired by linear prediction modeling of the vocal tract	Inspired by human auditory perception of frequency
Dimensionality	Typically has a lower dimensionality	Generally has a higher dimensionality
Common Usage	Less commonly used in general speech processing tasks	Standard and widely used in speech and audio processing

2.4 Classifiers

After extracting all valuable information from data SER systems use classifiers to learn and detect patterns that they attribute to various emotions. A classification algorithm takes an input X, typically in the form of labeled data, and maps it onto an output Y. The mapping function is approximated aiding in predicting the class of the next input. The learning algorithm utilizes the labeled data to identify samples and their relevant classes. Data is used to train the classifier and to further test and validate its performance. For SER data used are in the form of feature vectors obtained in the feature extraction process. There is no preferred classification approach to SER. The choice can be based on past references or experimental evaluation. The performance is then greatly affected by the combination of feature extraction and classification method [33] and several algorithms can be combined to improve predictions [4].

2.4.1 Traditional Machine Learning Classifiers

A traditional learning classifier, refers to a type of machine learning algorithm that are trained using supervised learning, where the algorithm learns to predict the correct class label for input data based on labeled examples provided during training. These classifiers are typically based on well-established algorithms and techniques.

Many traditional classifiers offer interpretable models, meaning that the decision-making process can be understood and explained based on the learned parameters or decision rules.

Support Vector Machine (SVM)

Support Vector Machine is the supervised linear algorithm transforming the original input set to a high dimensional feature space by using a kernel function, in which input space is converted into high dimensional feature space making the input data become linearly separable. The main advantage of SVM is that it has limited training data and hence has very good classification performance [27].

Hidden Markov Models (HMM)

Hidden Markov Model is a supervised algorithm used in speech recognition and successfully extended for use in SER. HMM is a sequential model relying on the continuity of states in time. The current state of a system is at time t and only depends on the previous state in $t-1$. The term *hidden* implies the inability to observe the generation of state-generating logic and we can use probability to predict the next state only by observing the current state [4].

Gaussian Mixture Models (GMM)

GMM is a probabilistic method that models the data as a mixture of several components with their parametric form. Each data point then belongs to one of the components. We can view GMM as a special continuous case of HMM with just one state [4].

Table 2.5: Traditional machine learning classifiers - overview

	SVM	**HMM**	**GMM**
Output	Direct classification	Sequence of hidden states representing emotions	Probabilistic representation of emotions
Feature representation	High-dimensional feature spaces	Often requires carefully selected features	Flexible feature types
Training data requirements	Labeled data for each class	Labeled sequences of emotional states	Labeled data for each component representing an emotion
Benefits	Effective in high-dimensional feature spaces	Can capture sequences of emotions	Probabilistic representation allows uncertainty

2.4.2 Deep Learning Based Classifiers

Currently most models in SER pivoted to using deep learning, which has been outperforming traditional machine learning approaches [13]. The main argument for the utilization of deep learning lies in its ability to automatically extract features from raw audio.

While traditional machine learning approaches (described in 2.4) rely on handcrafted features provided in structures appropriate as their input, deep learning algorithms process data and extract features during the computation. The drawback is a requirement for larger datasets in the learning process. To achieve the best results deep learning models can be combined with handcrafted features [13].

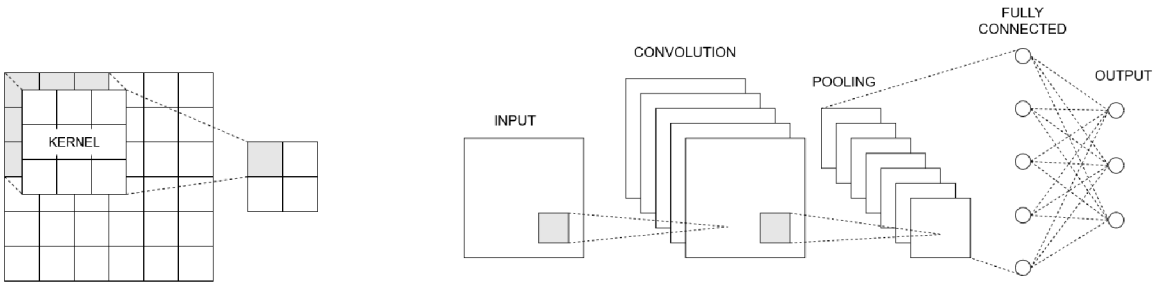
Recurrent Neural Network(RNN)

The recurrent neural network is a discriminative supervised model built on **recurrent architecture**. By the usage of internal memory they can process sequential data, remember received input and from their interdependencies predict the future state of input. RNNs use a recurrent loop where an output of previous cycle is used in the next one. Each step processes one part of the input which allows for time-dependency modelling.

Convolutional Neural Network (CNN)

The convolutional neural network is a very successful network in the field of pattern recognition. CNNs are adept at extracting abstract features as data progresses through deeper layers. For instance, in image classification, early layers might detect edges, followed by simpler shapes, and ultimately higher-level features like faces in subsequent layers [5]. The networks consist of multiple layers as shown in Figure including convolutional layer, non-linearity layer, pooling layer and fully-connected layer for making predictions.

Convolution in CNNs serves the purpose of parameter reduction. Instead of connecting every part of the input we connect only local regions as shown in Figure 2.5a whose weights remain fixed. We can further utilize this as a method of applying filters and after adding more layers extracting different features from the input regardless of their position in the input [5].



(a) Convolution of one local region

(b) A basic CNN structure

Figure 2.5: Principles of CNN

Non-linear layers manage the output by saturating or limiting it [5]. We manage this by adjusting or cutting off the generated output. The Rectified Linear Unit (ReLU) achieves this by propagating any positive values and setting all negative values to 0 as shown in Equation 2.3.

$$ReLU(x) = max(0, x) \tag{2.3}$$

In order to reduce the complexity, we send to the next layer we have to down-sample. The down-sampling is done by a **pooling layer**. The purpose of this is to achieve spacial invariance [30] meaning the existence of a feature is left but it does not matter in which region the feature was.

Chapter 3

Proposed Methodology

This chapter utilizes information gathered in the previous chapter and through understanding of principles of core components of an SER describes the proposed approach to designing such system. It focuses on the building blocks of the system and their functions within the system.

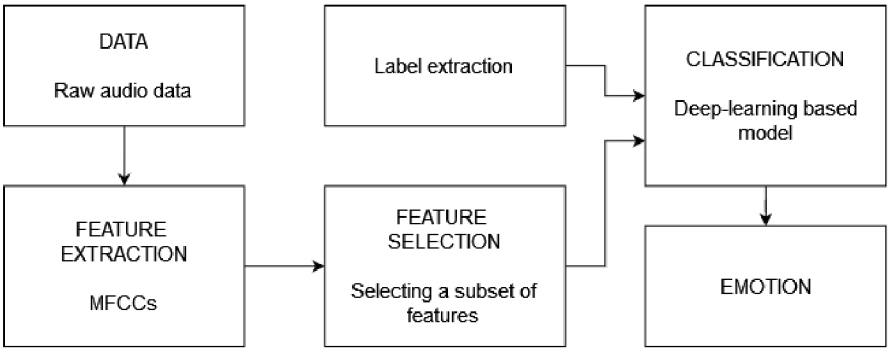


Figure 3.1: Blocks of the proposed SER system

3.1 Feature Extraction

When working with data in any form it is necessary to establish how to use the information contained in them in the most efficient way possible. Datasets often contain real-life data that have high dimensionality [32]. Reducing dimensions using feature extraction helps in data compression therefore can greatly reduce storage space and computation time [2]. Feature extraction is the process of removing all ineffective features while extracting the important and relevant data, aiding in increase of learning speed and generalization in machine learning process [2].

Feature extraction is a computation of feature vectors providing a representation of a speech signal. It is done in three stages. First, a spectra-temporal analysis is performed. It produces features describing the envelope of the power spectrum. The second stage produces a feature vector of static and dynamic features and in the third stage makes these vectors more robust and compact [11].

Mel-frequency Cepstral Coefficients 2.3.2 are often used in speech related classification problems due to their ability to extract rich amount of information from speech signal [2] and performing extraction similarly how a human ear processes sound [11] without

capturing noise in the signal. Spectral features are in general shown to perform well in n-way classification problems [12]. Despite deep-learning algorithms being able to perform deep feature extraction directly from raw data [2], the step serves the purpose of compressing the data and removing personal characteristics of speakers from the learning process.

3.1.1 Feature Selection

Feature selection involves selecting a subset of features from the original set without altering them and assessing their relevance to the analysis objective [18]. This process can be accomplished using various methods depending on the goal, available resources, and desired level of optimization.

It typically follows a process of generation, evaluation, definition of stopping criterion and final validation [18]. However, the first few MFCCs often capture essential spectral characteristics of the audio signal, such as information related to formants, spectral envelope, and fundamental frequency making them a viable subset. By selecting only the first 40 MFCCs [3][8], the feature representation focuses on the most relevant and discriminative information.

3.2 Deep Learning Model

Deep learning models are composed of layers of neurons connected by weighted edges. Structure and arrangement of these layers and connections between neurons, define the architecture of the model. By having multiple layers, deep learning models can learn hierarchical representations of the input data, with each layer capturing increasingly more abstract and complex features, enabling deep learning models to effectively learn and generalize.

This section describes parts of a deep learning model proposed by Aftab et al [3] shown in Figure 3.2.

3.2.1 Receptive Field

A receptive field of a neural network is defined as the size of the region in input that produces the feature [6]. As described in 2.5a each neuron in a CNN is connected to a small localized region of the input data. The size of a receptive field affects the ability of the network to capture finer or more global features.

To ensure that a unit in a Convolutional Neural Network captures all relevant information from the input, it's crucial to carefully control its receptive field. This ensures that the unit encompasses the entire region of the input that contains pertinent features. Otherwise, any information outside the receptive field of a unit would have no impact on its value. Thus, by managing the receptive field size effectively, the CNN can accurately extract meaningful features [24].

We can increase the receptive field linearly by adding more layers and making the network deeper. Each layer increases the receptive field by its kernel size [24]. This however creates more parameters of the network and causes over-fitting of the model.

The model shown in Figure 3.2 proposes a multi-receptive field (part Body I), working with the fact that feature extraction provides a multi-dimensional input: spectral, temporal and spectra-temporal features. Each dimension of the input has its own convolution unit containing a convolutional layer, batch normalization a non-linear layer and a pooling layer.

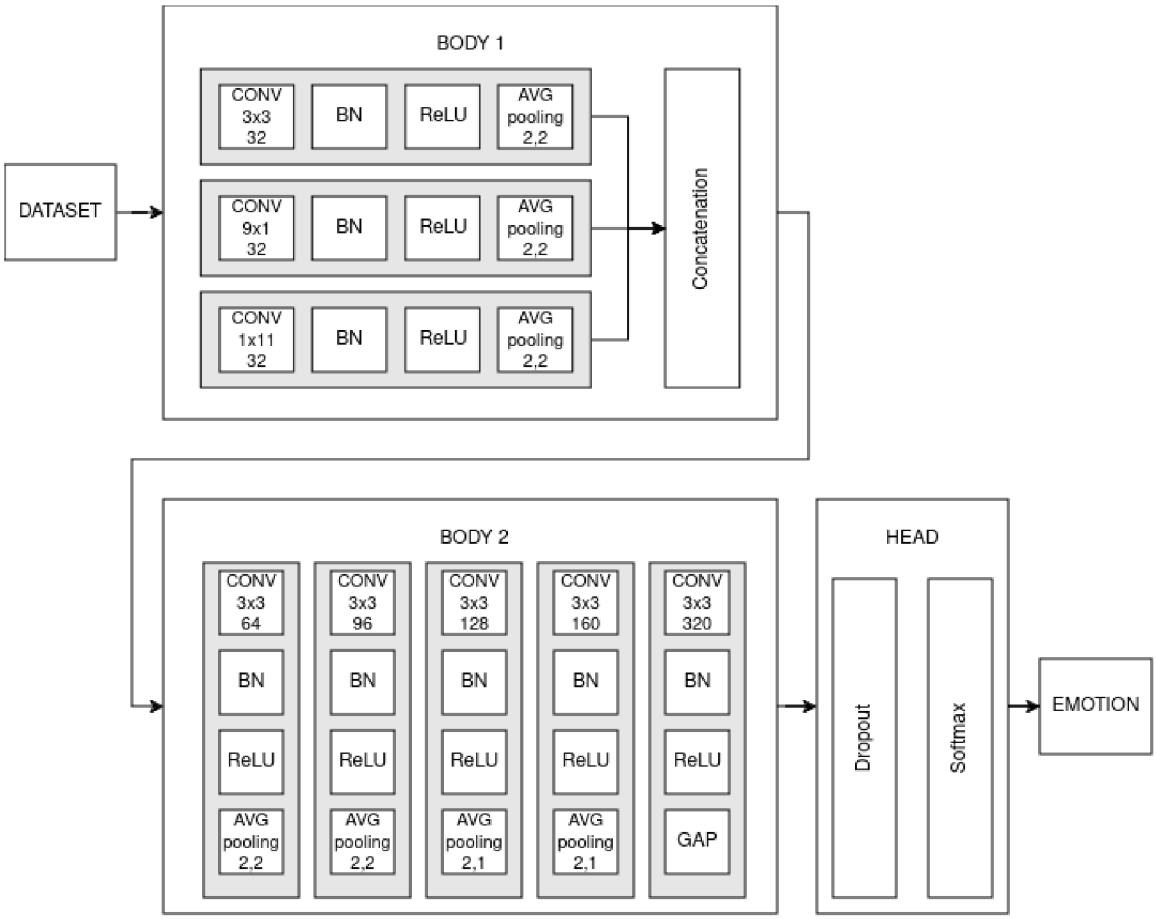


Figure 3.2: Architecture of the model

The outputs of these units are aggregated at the end. This approach manages to increase the receptive field of the network while preventing large parameter count [3].

3.2.2 Feed Forward Propagation and Backwards Propagation

Feed forward neural networks are a type of neural networks without any output feedback (like in a recurrent neural network 2.4.2). In feed forward neural layers the input signals are propagated to the output by weights and neuron biases [38]. The input data performs the required computations, affected by the network's parameters and produces an output prediction. Once the forward pass is complete, a loss function 3.2.3 is used to assess how much the predicted value differs from the true target.

To give the network the ability to learn, the back propagation updates the weights of connections based on the error rate of the forward run. The backward pass computes the gradients of the loss function 3.2.3 layer by layer starting at the output respecting the parameters of all layers. The weights can then be adjusted in a way that minimizes the loss function, thus achieving better performance. The magnitude of parameter step is controlled by the learning rate.

The process of forward and back propagation is repeated iteratively over multiple epochs until the model reaches a satisfactory solution or a predefined stopping condition.

3.2.3 Loss Function

The loss function estimates the degree of difference between the prediction and the true value. It's typically a function that yields a non-negative real value, expressed as Equation 3.1. Reducing the loss function typically indicates an improvement in the model's robustness [38]. This metric quantifies how well the model's predictions align with the actual data. Lower loss values suggest that the model is making predictions closer to the ground truth, indicating enhanced performance and robustness. Therefore, minimizing the loss function is a key objective in training neural networks, as it reflects the model's ability to generalize well to unseen data and effectively capture underlying patterns in the training dataset.

$$L(Y, f(x)) = |Y - f(x)| \tag{3.1}$$

Cross-Entropy Loss Function

Cross-entropy loss is a probability-based loss function. It quantifies the difference between predicted probability distribution of classes and the true distribution provided by labels. It applies a softmax normalization 3.2.5 which ensures that the scores provided by neural network can be interpreted as probabilities.

When we are dealing with a classification problem with N classes we expect the neural network to have an N-dimensional score representation space described in Equation 3.2, where $\mathcal{X}_L$ denotes the set of samples labeled by L. The softmax function takes this space and normalizes the scores. Cross-entropy loss is then computed as a negative logarithm of the probability of the true class label. Then the gradient of the loss is computed as a difference between the vector of the softmax scores and a vector representing the true labels (with 1 for the true class and 0 for other classes) [22].

$$\mathcal{F}(x \in \mathcal{X}_L) = [s_1, s_2, \ldots, s_N]^\top \tag{3.2}$$

3.2.4 Overfitting and Dropout Layer

Overfitting occurs when a model fails to generalize from observed to previously unseen data, causing perfect predictions on training set and poor performance on testing set. Many factors can be the cause of overfitting. Generally, there are three kinds of situations [39]: (1) **noise learning on the training set**: when the training data is small, unrepresentative, or contains excessive noise, the model may inadvertently learn these noise patterns alongside genuine relationships. This can lead to over-reliance on irrelevant details during prediction; (2) **hypothesis complexity**: In statistical and machine learning contexts, the complexity of hypotheses involves a trade-off between variance and bias. When models become overly complex, often by incorporating too many inputs or hypotheses, they may achieve high accuracy on average but exhibit low consistency across different datasets; (3) **multiple comparisons procedures**: induction algorithms, including those used in artificial intelligence, involve comparing multiple items based on evaluation scores to select the most promising candidate. However, this process introduces the risk of selecting items that do not genuinely improve classification accuracy or may even decrease it.

Dropout is a regularization technique commonly used in neural networks, particularly in deep learning models, to prevent the unwanted interdependencies among neurons on training sets [14] and reducing the risk of overfitting. In each training iteration, for every neuron in the dropout layer, a random binary decision is made whether to retain the neuron

or omit it („drop it out“). This produces the same result as averaging the predictions of a large number of networks, but in a reasonable time [14].

3.2.5 Activation Function

By default, a neural network without an activation function is a simple linear function unable to recognize complex mapping of the data [31]. Activation functions introduce non-linearity, enabling the network to extract complex information and intricate patterns from data and represent non-linear mappings between inputs and outputs. A crucial aspect of activation functions is their differentiability, which enables the implementation of backpropagation 3.2.2.

Softmax Activation Function

The softmax activation function provides a vector of probability distribution from a vector of real numbers. It is particularly useful in classification problems with multi-class models as the returned vector represents the probability of each class (Equation 3.3 [31]), with the target class having the highest value.

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for} \quad j = 1, \ldots, K \tag{3.3}$$

3.3 Cross-Validation

Trained models can show a satisfactory performance on the training sets while not being able to generalize well and fail when shown previously unseen data. Cross-validation is a technique used to evaluate models' ability to generalize to new data. We achieve this by splitting the data into a training and testing set. One part is used for training while the other is withheld and used after the model is trained. The ratio of splitting the data is known as validation size [37].

The value is usually represented in the form of a fraction denoting a percentage of the dataset (a validation size of 0.2 indicates that 20% of dataset will be reserved for a certain task). The choice of this parameter involves a trade-off between the amount of data used for training resulting in over or under fitting of the model and the reliability of the validation.

3.3.1 Hold–out Cross–Validation

Hold-out validation is a validation method solving overfitting problems of validation set being a subset of training data. The data is divided into two disjunct parts: one for training and one for validation the model. The validation samples are introduced after the model has been trained.

These two subsets usually have a different validation size, we use a 0.8 to 0.2 training to validation ratio, but any other combination is possible. When splitting the data we have to keep in mind that the distribution of information is the key to successfully validating the model and preventing overfitting [37].

3.3.2 K-Fold Cross-Validation

Similarly, as in the hold-out validation the data is split into two parts used for training and validation. The k-fold method first splits the data into K equal parts called **folds**. The data is usually stratified in order to provide equal distribution of classes through the folds. One of the folds is then used for validation and the other folds are used for training. This process then iterates K times, every iteration using a different fold as the validation set. The model accuracy is then expressed as the average of the iterations.

Choosing the value of K the size of the dataset and the computation time has to be considered. K-fold validation can outperform hold-out validation [37], however the number of iterations can result in unacceptable computation time.

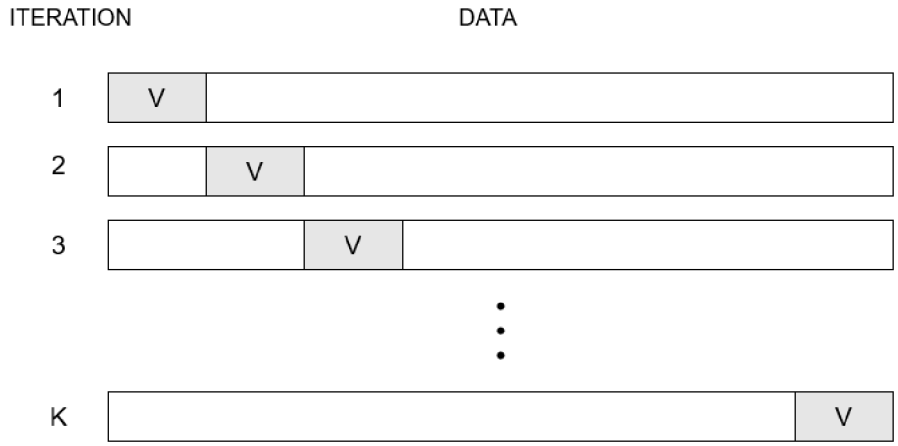


Figure 3.3: K-fold validation process

Chapter 4

Implementation

The proposed implementation takes root in the model described in 3.2. This chapter details the implementation part of handling data, constructing the model, training process and final evaluation.

The implementation utilizes Python as the main programming language. The model is implemented in the PyTorch[1]. Significant libraries used are `Torchvision`, `Torchaudio`, `torch.nn`, `torch.optim`, `NumPy` [2], `Matplotlib`[3], `librosa`[4] and `scikit-learn`[5].

All necessary dependencies are provided in the `requirements.txt` file and more detailed information on the usage of scripts is provided in the `README.md` file.

4.1 Data

This system utilizes two databases: The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) and Berlin Emotion Database (EmoDB). These datasets share many similarities: they are recorded by actors, have discrete labeling and share majority of emotion classes. Both are described in Table 2.3.

Emotion classes of these databases do not perfectly correspond. Some classes had to be omitted in order to have a unified set. The unification is depicted in Figure 4.1. The majority of classes stayed the same and the overall number of samples was reduced by approximately 24% in case of RAVDESS and 15% for EmoDB.

4.1.1 Labels

The Berlin Emotion Database (EmoDB) is labeled based on the discrete model 2.1.2. Each label provides information in form of 7 characters representing the speaker identification, code for transcription, emotion and the version of the recording. The transcription and speaker flags were not used for any purpose as the model does not utilize transcriptions in any way and feature processing is speaker indifferent. The RAVDESS database utilizes discrete models as well. The label provides information about the media type, transcription, intensity of emotion, speaker and emotion class.

[1] https://pytorch.org/
[2] https://numpy.org/
[3] https://matplotlib.org/
[4] https://librosa.org/doc/latest/index.html
[5] https://scikit-learn.org/stable/

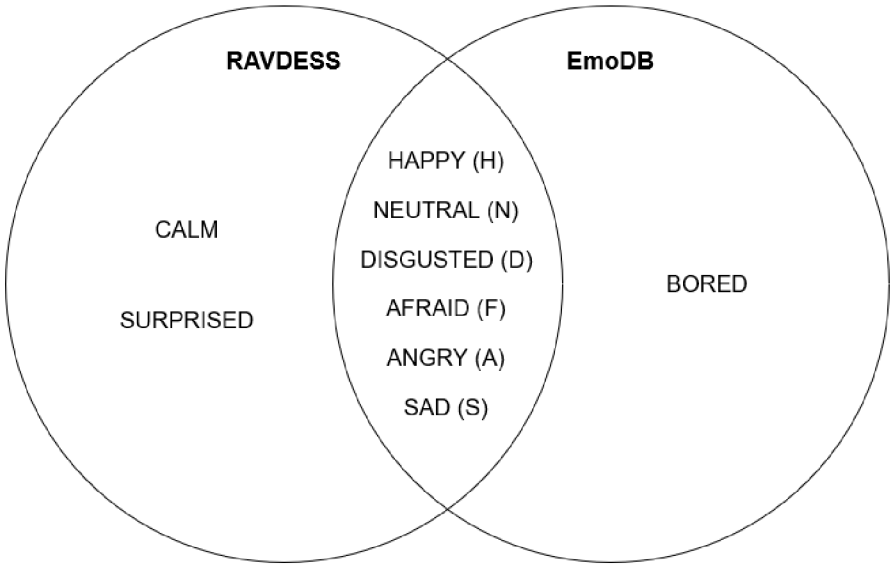


Figure 4.1: Class reduction in used datasets

The labels are then reconstructed to better fit the implementation steps. First label contains a single character representing the emotion class (emotion abbreviations are listed in Figure 4.1). This label is later used to stratify the classes in dataset partitioning. The same information is then used to create a one-hot-encoding-based vector used as true value during training. One-hot encoding-based vector is a binary vector that sets one value at index corresponding to the representing class to 1 and sets others as 0. Dataset is then constructed as in Figure 4.2.

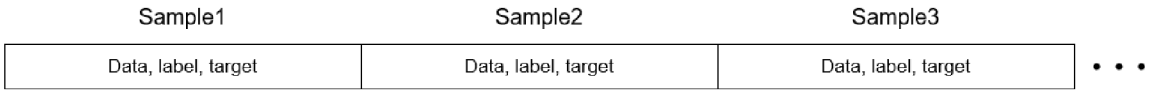


Figure 4.2: Dataset composition

4.1.2 Dataset partitions

Before training the dataset is partitioned into three parts: the training set, the final validation set and a testing set used to assess the performance after every epoch. The partitions divide the dataset in 0.7 to 0.3 ratio (training to testing) and the testing set is then halved for validation.

The dataset splitting is done using `sklearn.model_selection.train_test_split` to ensure stratification of the classes. The resulting subsets then have the same class distribution as the original dataset.

Each subset is then divided into batches. The number of batches set for training control the rate of network parameter updates. The forward and backward propagation 3.2.2 occur for every batch. If these updates occur too often it can lead to learning some local minimum or saddle point. On the other hand, if these updates occur too infrequently it can make the learning process to slow down and require more iteration in order to reach satisfactory performance. To prevent both scenarios and taking the smaller dataset size into account the sets are divided into 32 batches.

4.1.3 Feature Extraction and Selection

The preprocessing 2.3.1, feature extraction and feature selection 3.1 is all done in the `MFCC_computing.py` script.

The raw audio used was sourced from an acted databases and did not contain any abnormal levels of noise therefore no noise reduction was performed. The signal was loaded and normalized by `torchaudio.load()` function. The audio then undergoes voice activity detection. The RAVDESS samples have a significant gap of silence at the beginning of recordings. The voice activity detection is implemented using the zero-crossing rate method 2.3.1 using the `librosa.feature.zero_crossing_rate`.

The waveform was then used to compute a set of Mel-Frequency Cepstral Coefficients 2.3.2. Using `torchaudio.transforms.MFCC()` class a set of first 40 coefficients were extracted by using a *1024*-point fast Fourier transform and a Mel filter bank with *40* filters. The window length is set to *32ms* frames with *16ms* hop size. The output is two dimensional: one dimension representing the frequencies and the other representing time. This allows capturing both temporal and spectral features during convolution. Figure 4.3 serves as a visualization of the output tensor.

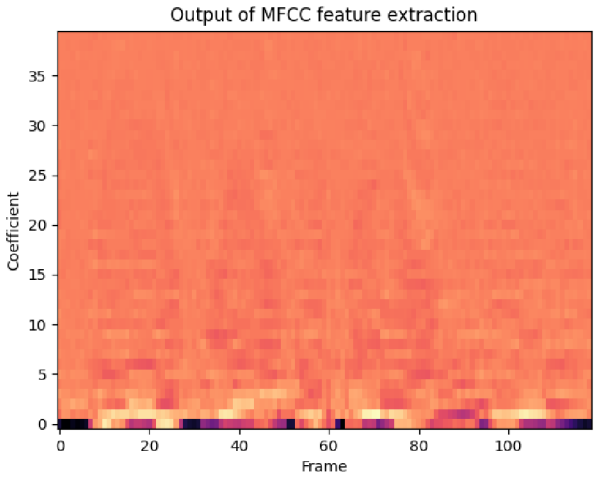


Figure 4.3: Feature extraction output for a sample in EmoDB dataset

4.2 Model

The definition of model architecture in `models.py` script follows the proposed architecture 3.2. Model is implemented using `torch.nn.Module`. The layers are separated into classes following the same logic parts as the proposed model: `Body1`, `Body2` and the final class adding the head `ModelCNN`.

The softmax layer depicted in Figure 3.2 is not added when defining the `ModelCNN` class layers. Instead, only a single fully-connected layer is in its place and the softmax activation is performed by cross-entropy loss function.

4.2.1 Training

The training loop is performed for a predefined number of epochs. Each epoch produces a separate model parameters. The training follows three main steps described in Figure 4.4: the forward propagation of input producing the prediction, computation of loss assessing the difference between the prediction and the true label and final back propagation. These steps are performed in an inner loop for every batch in dataset. The loss of each batch is then used to compute the average train loss for a given epoch.

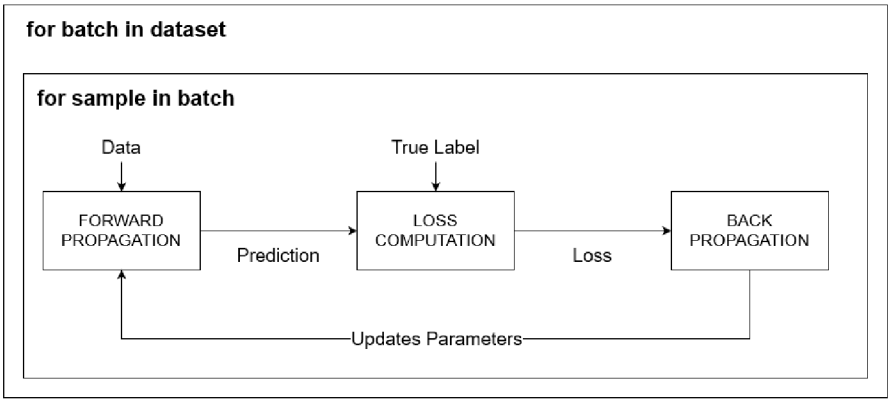


Figure 4.4: Model training loop

After every training loop, the model is tested on a different set than given to the training loop. The testing loop omits the back propagation step and therefore does not update the model parameters at all. Same as the training loop the average loss is computed as well as the number of correct model predictions. Based on these numbers the model evaluation is performed and the model parameters are saved. Once all the epochs have finished the best model is chosen. The choice is based on the test evaluation.

During the training the model is in a „training mode“ (`model.train()`). This activates the batch normalization and dropout layers. In the testing stage the model is set to „evaluation mode“ (`model.eval()`) causing the dropout layer to be inactive in order to produce deterministic results and the batch normalization layers to use statistics obtained during training. This encourages more consistent behavior of the model.

Optimizer and Loss Function

The loss is computed by the `torch.nn.CrossEntropyLoss` as this is a multi-class classification problem. The cross-entropy loss function is described in 3.2.3.

Optimizer adjusts the network parameters in a way that minimizes the loss. The Adaptive Moment Estimation or `torch.optim.Adam` algorithm was used. This algorithm adjusts the learning rates based on the gradients as well as based on information stored in moving averages. This approach helps the model to converge faster.

When initializing the optimizer weight decay and learning rate parameters were set to regulate the performance. The weight decay parameter is set to prevent overfitting by penalizing overly complex patterns with large weights. The learning rate defines the rate at which parameters are updated.

4.3 Validation and Evaluation

The model is validated using the hold-out cross-validation method 3.3.1. To produce an objective performance and generalization estimate the validation utilizes a different data source then during training.

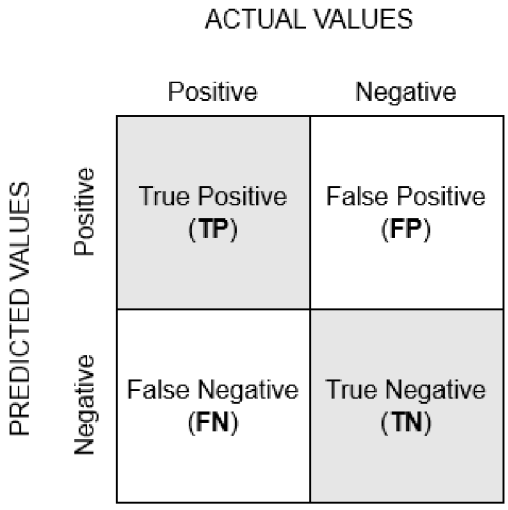
During training, each epoch produces a version of the model that can be tested for accuracy. This is done by the training loop on the training dataset. For the final validation the best model is chosen from all the epochs based on the accuracy it achieved. Early stopping of the learning is done to avoid deterioration of accuracy once the model starts to overfit.

4.3.1 Evaluation Metrics

To assess how well the model is performing we can quantify the performance using evaluation metrics. The ones used to evaluate the proposed model are: accuracy, precision, recall and F1-score. The implementation utilizes `sklearn.metrics.classification_report` for the calculations and `sklearn.metrics.confusion_matrix` to visualize them in confusion matrix.

Confusion matrix helps evaluate models' performance by comparing and visualizing the predictions compared to actual values. The matrix defines 4 scenarios: true positive, true negative, false positive and false negative shown in Figure 4.5a. The true positive and the true negative are the correct scenarios. In true positive the model correctly predicted the true value and in true negative the model did not predict a false value. The false positive and negative represent faulty predictions.

With this understanding we can then construct the matrix for n classes as shown in Figure 4.5b. We define only true positive and false positive cases. In the context of more than one class we can reduce the false scenarios to just one as it simply represents a wrong prediction. True negatives for a class are all true positives of all other classes. Information from confusion matrix can then be used to derive other performance metrics.



(a) Confusion matrix for one class

	Class 1	Class 2	Class 3	Class 4
Class 1	**TP** 1	**FP** 1→2	**FP** 1→3	**FP** 1→4
Class 2	**FP** 2→1	**TP** 2	**FP** 2→3	**FP** 2→4
Class 3	**FP** 3→1	**FP** 3→2	**TP** 3	**FP** 3→4
Class 4	**FP** 4→1	**FP** 4→2	**FP** 4→3	**TP** 4

(b) Confusion matrix for $n = 4$ classes

Figure 4.5: Confusion matrix

Accuracy defines the proportion of correct predictions to the number of all samples in dataset. It gives an overall measure of how correct the predictions are. From confusion matrix data we can derive accuracy as Equation 4.1.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

Precision defines accuracy of only positive predictions. It does not take account of the true negative. It is calculated as Equation 4.2

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

Recall measures the proportion of true positive predictions to all positive instances in the dataset. It defines the models' ability to identify positive instances of actual values. It is calculated as Equation 4.3

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

To balance the precision and recall values we compute the **F1-score**. It is the harmonic mean of precision and recall values as shown in Equation 4.4. this value can be especially useful in cases of unbalanced classes. If the classes are unbalanced accuracy remains high as it may be biased towards the majority class.

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4.4}$$

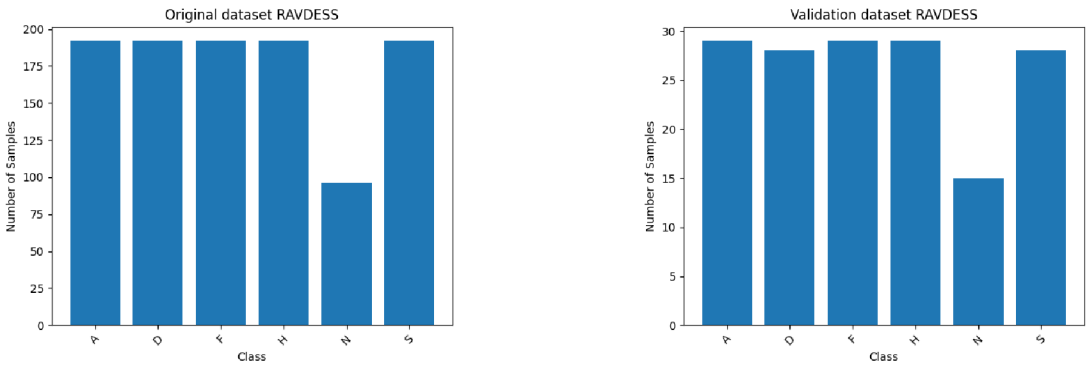
Chapter 5

Results

The model described in the previous Chapter 4 was tested on two datasets described in 4.1. The purpose of these experiments was to evaluate the success of the model implementation and its performance. Experiments include training and validation of the model on both datasets separately. All results shown in this chapter were obtained using parameters provided in `hyperparameters.py`. Evaluation methods and significance of the metrics is described in 4.3.

5.1 RAVDESS

The RAVDESS dataset was selected in order to compare the model performance when provided with more samples and utterances recorded in a different language corpus. The class distribution in this dataset is shown in Figure 5.1a. Compared to EmoDB the emotion classes are uniform in sample count with the exception of the *Neutral* (N) class having half the samples.

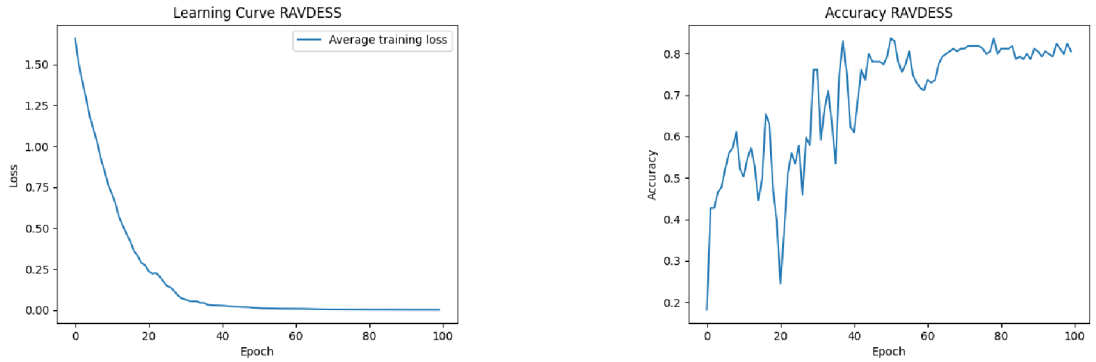


(a) In the original dataset

(b) In the validation dataset

Figure 5.1: Class distribution in RAVDESS

The experiments shown the model reaches an accuracy of **84.2%** over *100* epochs, lower compared to EmoDB. The trend of training loss and training accuracy across epochs shown in Figures 5.2a and 5.2b shows major spikes in accuracy. The maximum accuracy was reached at epoch *80*. It showed a stable result that stayed at a level even after training over more epochs. The accuracy suggests that the model began converging around epoch *60*.



(a) Training loss across all epochs

(b) Accuracy across all epochs

Figure 5.2: Training loss and accuracy with RAVDESS

The confusion matrix for RAVDESS shown in Figure 5.3 suggests that the model done quite well with only occasional errors. The *Neutral* emotion has shown the worst results mainly in *Sadness-Neutral* case. This result is consistent with the evaluation metrics shown in Table 5.1. The best result was reached for *Anger*.

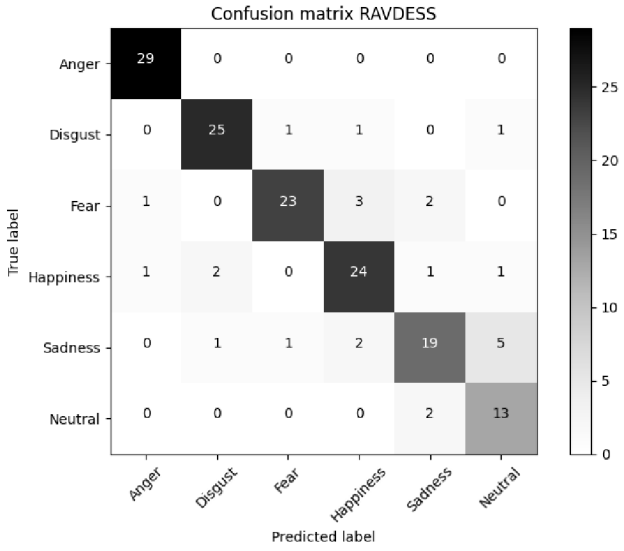


Figure 5.3: Confusion matrix for RAVDESS

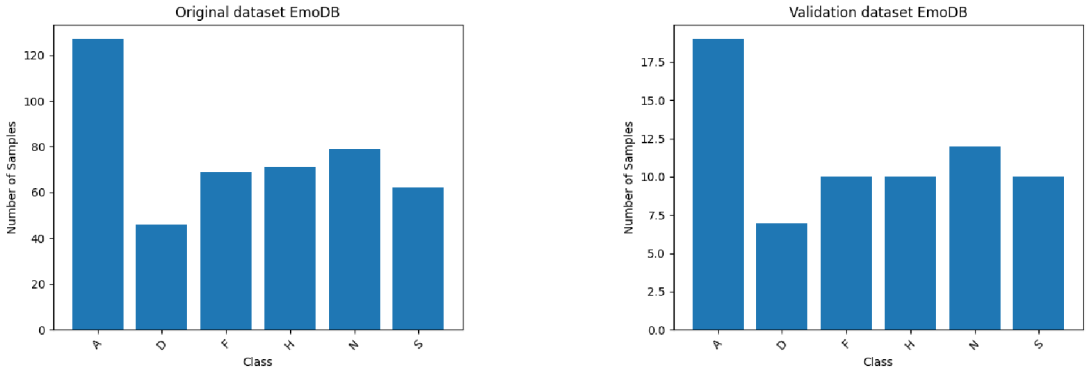
5.2 EmoDB

The EmoDB dataset is one of the smaller in size datasets available and used for training automated speech emotion recognition systems compared to the rest in Table 2.3. The class distribution across this dataset shown in Figure 5.4a shows that *Anger* (A) is a majority class and *Disgust* (D) a minority class. Due to no data augmentation step the classes

Table 5.1: RAVDESS class evaluation metrics

Emotion	**Precision**	**Recall**	**F1-score**
Happy	0.80	0.83	0.81
Neutral	0.65	0.87	0.74
Disgusted	0.89	0.89	0.89
Afraid	0.92	0.79	0.85
Angry	0.94	1.00	0.97
Sad	0.79	0.68	0.73

remained distributed this way in all subsets 4.1.2. The distribution shown in Figure 5.4b shows the same trend in class sample counts in the validation subset.



(a) In the original dataset

(b) In the validation dataset

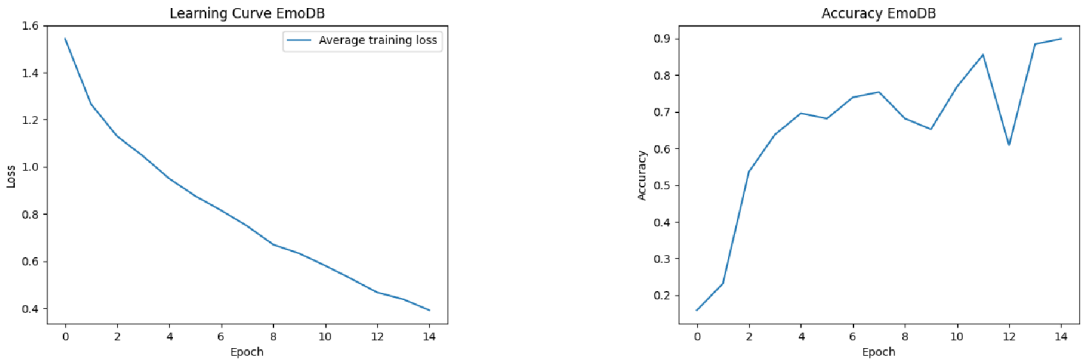
Figure 5.4: Class distribution in EmoDB

The training and testing loops were iterated over *15* epochs. Figure 5.5a depicts the average training loss over the whole training. The average per epoch is calculated from training losses of batches. Figure 5.5b shows the training accuracy. We can notice rapid spikes in the accuracy values compared to the training losses. The low number of epochs was chosen due to the model becoming less stable and failing to provide any meaningful results. At epoch *15*[1] the model reached its peak accuracy at **92.6%**.

Figure 5.6 shows the confusion matrix for the validation subset. The matrix shows an overall good performance with low number of false positives and false negatives. The only remarkable occurrence of a false negative is the *Happiness-Anger* showing consistently across all experiments.

Evaluation metrics for each class are concluded in Table 5.2. The table showed a possible overfitting problem based on the presence of value *1.00* in the *Afraid* emotion row. The false negative problem observed in the confusion matrix is visible in the table in the **recall** value of *Happiness* reaching only *0.70* and **precision** of *Anger 0.83*

[1]The graphs showed are indexed from 0, whereas the text describes the number indexed form 1 for better readability



(a) Training loss across all epochs

(b) Accuracy across all epochs

Figure 5.5: Training loss and accuracy with EmoDB

Table 5.2: EmoDB class evaluation metrics

Emotion	**Precision**	**Recall**	**F1-score**
Happy	1.00	0.70	0.82
Neutral	0.92	1.00	0.96
Disgusted	1.00	0.86	0.92
Afraid	1.00	1.00	1.00
Angry	0.83	1.00	0.90
Sad	1.00	0.90	0.95

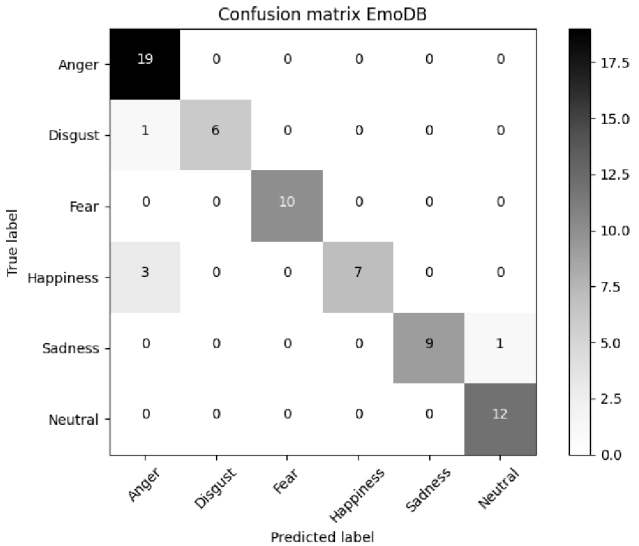


Figure 5.6: Confusion matrix for EmoDB

5.3 Discussion and Future Work

The proposed model was inspired by Aftab et al [3]. The system is lightweight and does not require big computation power nor time. The implementation presented in this thesis was done with adjustments to the paper. These alterations were done based on the literature reviewed in first chapters of this work as well as empirical experiments with the implemented system.

The overall result achieved showed lesser accuracy, however the model managed to converge early on in the training process due to added audio preprocessing stage. The generalization of the model degrades when presented with a cross-corpus validation. The choice of the discrete model as a labeling method reduces the applicability of the model to a limited predefined set of emotions and has no way of detecting or interpreting mixed emotions from the results.

The experiments shown *Anger* as a strong and detectable emotion. Despite having a uniform number of samples in the RAVDESS experiment it achieved consistently good results. The size and stratification of classes in subsets proved to be a necessity. It is the belief of the author that the instability of the EmoDB experiment could be solved by data augmentation and unifying the sample count of each class.

Overall, the system showed the ability to learn and recognise emotion. The work presented meaningful insight into methodology of designing and utilizing such systems as well as described various challenges in the field from theoretical and methodological standpoint.

Proposed Improvements

The system would benefit from future alterations. The main proposal is to change the validation method to the K-fold approach in order to achieve better data utilization. Other possibilities are further training with different input parameters, data augmentation and testing on different datasets like IEMOCAP or MSP-Podcast providing not acted but elicited and natural speech recordings. These alterations have the possibility of making the system more robust.

Bibliography

[1] Darwin, C. *The Expression of the Emotions in Man and Animals*. London: John Murray, 1872.

[2] Abdul, Z. K. and Al Talabani, A. K. Mel Frequency Cepstral Coefficient and its Applications: A Review. *IEEE Access*. 2022, vol. 10, p. 122136–122158. DOI: 10.1109/ACCESS.2022.3223444.

[3] Aftab, A., Morsali, A., Ghaemmaghami, S. and Champagne, B. LIGHT-SERNET: A lightweight fully convolutional neural network for speech emotion recognition. In: IEEE. *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2022, p. 6912–6916.

[4] Akçay, M. B. and Oğuz, K. Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers. *Speech Communication*. 2020, vol. 116, p. 56–76. DOI: https://doi.org/10.1016/j.specom.2019.12.001. ISSN 0167-6393. Available at: https://www.sciencedirect.com/science/article/pii/S0167639319302262.

[5] Albawi, S., Mohammed, T. A. and Al Zawi, S. Understanding of a convolutional neural network. In: *2017 International Conference on Engineering and Technology (ICET)*. 2017, p. 1–6. DOI: 10.1109/ICEngTechnol.2017.8308186.

[6] Araujo, A., Norris, W. and Sim, J. Computing receptive fields of convolutional neural networks. *Distill*. 2019, vol. 4, no. 11, p. e21.

[7] Bachu, R., Kopparthi, S., Adapa, B. and Barkana, B. Separation of voiced and unvoiced using zero crossing rate and energy of the speech signal. In: American Society for Engineering Education. *American Society for Engineering Education (ASEE) zone conference proceedings*. 2008, p. 1–7.

[8] Bansal, V., Pahwa, G. and Kannan, N. Cough Classification for COVID-19 based on audio mfcc features using Convolutional Neural Networks. In: *2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON)*. 2020, p. 604–608. DOI: 10.1109/GUCON48875.2020.9231094.

[9] Cannon, W. B. The James-Lange Theory of Emotions: A Critical Examination and an Alternative Theory. *The American Journal of Psychology*. University of Illinois Press. 1927, vol. 39, 1/4, p. 106–124. ISSN 00029556. Available at: http://www.jstor.org/stable/1415404.

[10] Coles, N. A., Larsen, J. T. and Lench, H. C. A meta-analysis of the facial feedback literature: Effects of facial feedback on emotional experience are small and

variable. *Psychological bulletin*. American Psychological Association. 2019, vol. 145, no. 6, p. 610.

[11] Dave, N. Feature extraction methods LPC, PLP and MFCC in speech recognition. *International journal for advance research in engineering and technology*. 2013, vol. 1, no. 6, p. 1–4.

[12] El Ayadi, M., Kamel, M. S. and Karray, F. Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*. 2011, vol. 44, no. 3, p. 572–587. DOI: https://doi.org/10.1016/j.patcog.2010.09.020. ISSN 0031-3203. Available at: https://www.sciencedirect.com/science/article/pii/S0031320310004619.

[13] Hashem, A., Arif, M. and Alghamdi, M. Speech emotion recognition approaches: A systematic review. *Speech Communication*. 2023, vol. 154, p. 102974. DOI: https://doi.org/10.1016/j.specom.2023.102974. ISSN 0167-6393. Available at: https://www.sciencedirect.com/science/article/pii/S0167639323001085.

[14] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. R. Improving neural networks by preventing co-adaptation of feature detectors. *ArXiv preprint arXiv:1207.0580*. 2012.

[15] Hossan, M. A., Memon, S. and Gregory, M. A. A novel approach for MFCC feature extraction. In: *2010 4th International Conference on Signal Processing and Communication Systems*. 2010, p. 1–5. DOI: 10.1109/ICSPCS.2010.5709752.

[16] Ibrahim, Y. A., Odiketa, J. C. and Ibiyemi, T. S. Preprocessing technique in automatic speech recognition for human computer interaction: an overview. *Ann Comput Sci Ser*. 2017, vol. 15, no. 1, p. 186–191.

[17] Johnstone, T., Reekum, C. M. van, Hird, K., Kirsner, K. and Scherer, K. R. Affective speech elicited with a computer game. *Emotion*. American Psychological Association. 2005, vol. 5, no. 4, p. 513.

[18] Jović, A., Brkić, K. and Bogunović, N. A review of feature selection methods with applications. In: *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. 2015, p. 1200–1205. DOI: 10.1109/MIPRO.2015.7160458.

[19] Koolagudi, S. G. and Rao, K. S. Emotion recognition from speech: a review. *International journal of speech technology*. Springer. 2012, vol. 15, p. 99–117.

[20] Labied, M., Belangour, A., Banane, M. and Erraissi, A. An overview of Automatic Speech Recognition Preprocessing Techniques. In: *2022 International Conference on Decision Aid Sciences and Applications (DASA)*. 2022, p. 804–809. DOI: 10.1109/DASA54658.2022.9765043.

[21] Lazarus, R. S., Averill, J. R. and Opton, E. M. Towards a cognitive theory of emotion. *Feelings and emotions*. 1970, p. 207–232.

[22] Li, L., Doroslovački, M. and Loew, M. H. Approximating the Gradient of Cross-Entropy Loss Function. *IEEE Access*. 2020, vol. 8, p. 111626–111635. DOI: 10.1109/ACCESS.2020.3001531.

[23] Lu, X., Yang, H. and Zhou, A. Applying PAD three dimensional emotion model to convert prosody of emotional speech. In: *2014 International Conference on Orange Technologies*. 2014, p. 89–92. DOI: 10.1109/ICOT.2014.6956606.

[24] Luo, W., Li, Y., Urtasun, R. and Zemel, R. Understanding the effective receptive field in deep convolutional neural networks. *Advances in neural information processing systems*. 2016, vol. 29.

[25] Mary, L. and Yegnanarayana, B. Extraction and representation of prosodic features for language and speaker recognition. *Speech Communication*. 2008, vol. 50, no. 10, p. 782–796. DOI: https://doi.org/10.1016/j.specom.2008.04.010. ISSN 0167-6393. Available at: https://www.sciencedirect.com/science/article/pii/S0167639308000587.

[26] McGilloway, S., Cowie, R., Douglas Cowie, E., Gielen, S., Westerdijk, M. et al. Approaching automatic recognition of emotion from voice: A rough benchmark. In: *ISCA Tutorial and Research Workshop (ITRW) on Speech and Emotion*. 2000.

[27] Milton, A., Roy, S. S. and Selvi, S. T. SVM scheme for speech emotion recognition using MFCC feature. *International Journal of Computer Applications*. Foundation of Computer Science. 2013, vol. 69, no. 9.

[28] PS, S. and Mahalakshmi, G. Emotion models: a review. *International Journal of Control Theory and Applications*. 2017, vol. 10, no. 8, p. 651–657.

[29] Russell, J. A. A Circumplex Model of Affect. *Journal of Personality and Social Psychology*. American Psychological Association. 1980, vol. 39, no. 6, p. 1161–1178.

[30] Scherer, D., Müller, A. and Behnke, S. Evaluation of pooling operations in convolutional architectures for object recognition. In: Springer. *International conference on artificial neural networks*. 2010, p. 92–101.

[31] Sharma, S., Sharma, S. and Athaiya, A. Activation functions in neural networks. *Towards Data Sci.* 2017, vol. 6, no. 12, p. 310–316.

[32] Van Der Maaten, L., Postma, E. O., Herik, H. J. van den et al. Dimensionality reduction: A comparative review. *Journal of Machine Learning Research*. 2009, vol. 10, 66-71, p. 13.

[33] Wang, Y. and Guan, L. An investigation of speech-based human emotion recognition. In: IEEE. *IEEE 6th Workshop on Multimedia Signal Processing, 2004.* 2004, p. 15–18.

[34] Wani, T. M., Gunawan, T. S., Qadri, S. A. A., Kartiwi, M. and Ambikairajah, E. A Comprehensive Review of Speech Emotion Recognition Systems. *IEEE Access*. 2021, vol. 9, p. 47795–47814. DOI: 10.1109/ACCESS.2021.3068045.

[35] Wong, E. and Sridharan, S. Comparison of linear prediction cepstrum coefficients and mel-frequency cepstrum coefficients for language identification. In: *Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing. ISIMP 2001 (IEEE Cat. No.01EX489)*. 2001, p. 95–98. DOI: 10.1109/ISIMP.2001.925340.

[36] Wu, S., Falk, T. H. and Chan, W.-Y. Automatic speech emotion recognition using modulation spectral features. *Speech Communication*. 2011, vol. 53, no. 5, p. 768–785. DOI: https://doi.org/10.1016/j.specom.2010.08.013. ISSN 0167-6393. Perceptual and Statistical Audition. Available at: https://www.sciencedirect.com/science/article/pii/S0167639310001470.

[37] Yadav, S. and Shukla, S. Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification. In: *2016 IEEE 6th International Conference on Advanced Computing (IACC)*. 2016, p. 78–83. DOI: 10.1109/IACC.2016.25.

[38] Yang, J. and Li, J. Application of deep convolution neural network. In: *2017 14th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*. 2017, p. 229–232. DOI: 10.1109/ICCWAMTIP.2017.8301485.

[39] Ying, X. An overview of overfitting and its solutions. In: IOP Publishing. *Journal of physics: Conference series*. 2019, vol. 1168, p. 022022.

Appendix A

SD content

The attached media contains following items:

- `thesis-latex/` the LaTeX source of this thesis
- `program/` implementation source files
  - `saves/` saved model versions used in experiments
  - `data_processing.py`
  - `hyperparameters.py`
  - `MFCC_computation`
  - `plots.py`
  - `train.py`
  - `README.md` description of scripts and usage
  - `requirements.txt` required dependencies
- `data/` the data used for experiments
  - `EmoDB/`
  - `RAVDESS/`
- `thesis.pdf` thesis pdf file