UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

telecos BCN

Predicting emotion in speech: a Deep Learning approach using Attention mechanisms

A Degree Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Daniel Aromí Leaverton

In partial fulfilment

of the requirements for the degree in

TELECOMUNICATIONS TECHNOLOGIES AND SERVICES ENGINEERING, AUDIOVISUAL SYSTEMS SPECIALTY

Advisor: Dr. Francisco Javier Hernando Pericas

Barcelona, June 2021

Abstract

Speech Emotion Recognition (SER) has recently become a popular field of research because of its implications in human-computer interaction. In this study, the emotional state of the speaker is successfully predicted by using Deep Convolutional Neural Networks to automatically extract features from the spectrogram of a speech signal. Parting from a baseline model that uses a statistical approach to pooling, an alternative method is proposed by incorporating Attention mechanisms as a pooling strategy. Additionally, multi-task learning is explored as an improvement over the baseline model by assigning language recognition as an auxiliary task. The final results show a remarkable improvement in classification accuracy in respect to previous more conventional techniques, in particular Gaussian Mixture Models and i-vectors, as well as a notable improvement in performance of the proposed Attention mechanisms over statistical pooling.

Resum

En les últimes dècades, Speech Emotion Recognition (SER), o el Reconeixement d'Emocions per Veu, ha generat fort interès en l'àmbit del tractament de la parla per a les implicacions que presenta en la interacció humà-computador. En aquest treball, s'aconsegueix reconèixer l'estat emocional del parlant utilitzant xarxes neuronals profundes que extreuen de manera automàtica característiques contingudes en l'espectrograma del senyal de veu. Partint d'un model que utilitza analàlisi estadística per a pooling, es proposa una estratègia alternativa per a millorar el rendiment incorporant mecanismes d'Atenció. Com a millora afegida, s'explora el camp del mulit-task learning definint el reconeixement de l'idioma com a tasca auxiliar per al model. Els resultats finals obtinguts reflecten una millora substancial en la precisió comparat amb anteriors mètodes, concretament respecte Gaussian Mixture Models i i-vectors, i una milora notable en la precisió dels mecanismes d'Atenció respecte el pooling estadístic.

Resumen

En las últimas décadas, Speech Emotion Recognition (SER), o el reconocimiento de emociones por voz, ha generado un fuerte interés en el ámbito del tratamiento del habla por sus implicaciones en la interacción humano-computador. En este trabajo, se consigue reconocer el estado emocional del hablante mediante redes convolucionales profundas, capaces de extraer de manera automática características contenidas en el espectrograma de la señal de voz. Partiendo de un modelo que utiliza análisis estadístico para pooling, se propone una estrategia alternativa para mejorar el rendimiento incorporando mecanismos de Atención. Como mejora añadida, se explora el campo del multi-task learning definiendo el reconocimiento del idioma como tasca auxiliar para el modelo. Los resultados obtenidos reflejan una mejora substancial en la precisión comparado con anteriores técnicas más convencionales, concretamente Gaussian Mixture Models y i-vectors, y una mejora notable en la precisión de los mecanismos de Atención respecto al pooling estadístico.

Acknowledgements

Firstly, I would like to acknowledge the expert advice and support given by the advisor Javier Hernando. The insights, recommendations and references given by him provided me with the necessary tools and organizational skills to develop this project.

Secondly, this project could not have been possible without the help of Miquel Àngel India and the consistent support provided by him. His dedication to students and willingness to help were key to solve the most difficult challenges in this project. Thank you, Miquel.

Lastly, I would like to mention the work done at the beginning of this project regarding the survey done for SEAT was done jointly with Victor Emilio Hernández, another student and good friend, with whom I've shared this experience from the beginning.

Revision history and approval record

Revision	Date	Purpose
0	21/06/2021	Document creation
1	21/06/2021	Document revision

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Daniel Aromí Leaverton	daniel.aromi@estudiantat.upc.edu
Francisco Javier Hernando Pericas	javier.hernando@upc.edu

Written by:		Reviewed and approved by:	
Date	21/06/2021	Date	21/06/2021
Name	Daniel Aromí Leaverton	Name	Francisco Javier Hernando Pericas
Position	Project Author	Position	Project Supervisor

Table of contents

List of Figures

List of Tables:

1. Introduction

This introductory section is intended to give the reader context regarding this project; the motivation and reasoning behind the research done as well as the main goals it is intended to accomplish, to explain the methodology followed and lastly to describe the structure of this study.

1.1. Motivation and relevance of this project

The most amazing evolutionary advantage that humans have achieved is our capacity to act together towards a common goal. Civilizations have been created, cultures have been born and technological advancements have been achieved thanks to one special trait that we all have in common, and that is that we are extremely good communicators. We communicate in many different ways: through facial expressions, body language, touch, but by far the most effective and natural way in which we communicate is through speech. We are designed to convey information through sound.

Speech is such an important part of the human experience that we depend on it to communicate our thoughts, opinions and most important of all, our emotions. Emotions help us understand each other better. It is no coincidence that the use of emojis has become so common in text messages, as the lack of emotional content can lead to misunderstandings, and we have a need to express our emotional state as we would in speech. It only seems natural then that with the ever-growing automation and digitalization of our modern world we extend this ability to communicate emotion to machines. This fact is what motivated the development of this project, to dig deep into the great efforts that have been done to get computers to understand our emotional state, to explore and be a part of such a relevant field of study at this time, to join the effort to obtain *natural* interaction between human and machine.

Speech Emotion Recognition (SER) has really taken off with the explosion of Deep Learning techniques. Previous attempts at SER have provided acceptable if not remarkable results in some cases [1], but pale in comparison to what Deep Learning techniques have been delivering nowadays in SER [2]. This has led to a change of paradigm in which SER is no longer a niche field of study, but a full-blown area of research backed and sponsored by tech giants like Apple, Amazon and Huawei. This study aims to be the first of its kind in its institution, UPC, where state-of-the-art Deep Learning architectures are used in the task of SER, in order to propel forward the work done in this field and to contribute my part in the subject.

Results from this project, including plots, analysis and insights have been presented to SEAT as a way to introduce them to this field and potentially incorporate SER into their intelligent cars, proving that this field of study has commercial viability.

1.2. Objectives of this project

The main goal of this project is to implement and test a system capable of recognizing a closed set of 7 emotions, using specific features extracted from the speaker's voice using Deep Convolutional Neural Networks. Moreover, this project intends to enhance the system by introducing innovative attention mechanisms used in other fields related to speech processing, in particular for Speaker Verification purposes. Additionally, it explores the approach of multi-task learning, a sub-field of machine learning which consists of solving multiple tasks while exploiting commonalities and differences across tasks, in the hopes to improve the system's performance even further. The objectives and challenges to tackle in this project are, in a comprehensive list:

- To Implement a Deep Learning based system to predict emotion using features extracted from speech signals.

- To improve performance in respect to previous approaches that use more conventional methodologies, specifically GMM and I-vectors, to prove Deep Learning provides better results in SER than these techniques.

- To introduce pioneering attention mechanisms used in other Deep Learning applications related to speech processing and improve performance with these implementations.

- To implement multi-task learning into the model, obtaining simultaneous predictions from different tasks, with the intention of enhancing the performance of the main task of emotion recognition.

1.3. Methodology and context

The framework of this project falls under the research done by TALP (Tecnologies I Aplicacions del Llenguatge I la Parla), a research center affiliated to the UPC dedicated to speech and language applications and technologies. Previous work regarding SER has been done in TALP [3] as well as many other speech related studies, especially speaker recognition and language recognition.

Regarding structure, this project is approached as a typical classification problem; extraction of features, classification, test and results, following the schema from many works developed at TALP.

This project could not be possible without the recent work done by Miquel Àngel India and Javier Hernando [4] [5]. From these studies the idea of applying attention mechanisms was born, as they have proven to work remarkably well applied to the task of speaker verification and could potentially provide improvements to the emotion recognition task. From these studies I have been able to obtain the architecture used in this project, as well as the raw implementations of the proposed attention mechanisms; Self-Attention, Self-Multihead Attention and Double Mulithead Attention. The knowledge required to fully understand and apply these mechanisms was gained from the study of these works, as well as other materials provided by the authors.

The project was born as a potential application for an in-vehicle system, in which the car detects the driver's emotional state to initiate safety measures or intelligently adapt the cabin environment. The first experiments and briefings were presented to SEAT in a form of an explanatory survey. We gave an introductory explanation of a modern SER system as well as background information necessary to understand the task at hand, and have been in touch with the company during the entirety of the project, delivering results, plots and insights along the way.

The initial plan for this project was ambitious and included procedures and experiments that could not come to fruition due to time restrictions. The general idea was though out from the beginning, but work regarding feature fusion and experiments with different datasets unfortunately had to be put aside in order to focus on the main objectives. A Work Breakdown Structure with the corresponding work packages is shown in figure 1.1, detailing the internal tasks for each WP:

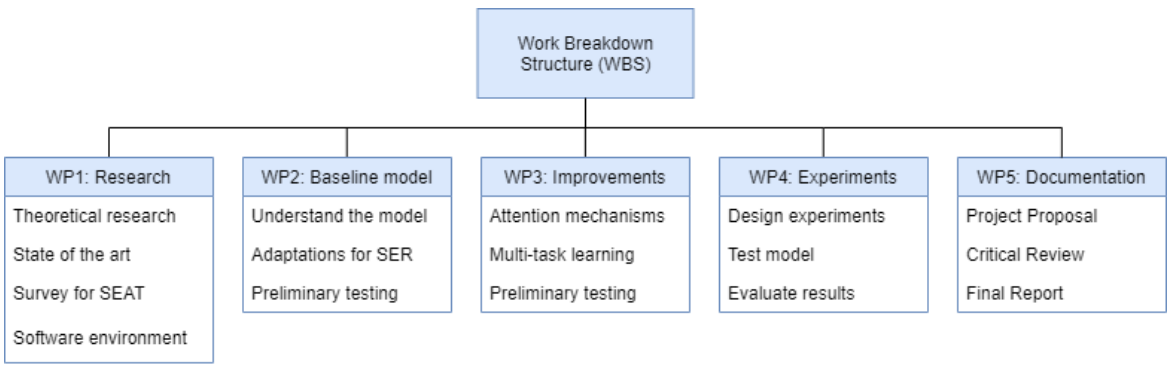

Figure 1.1. WBS of the project

Following is a Gantt diagram of the project timeline:

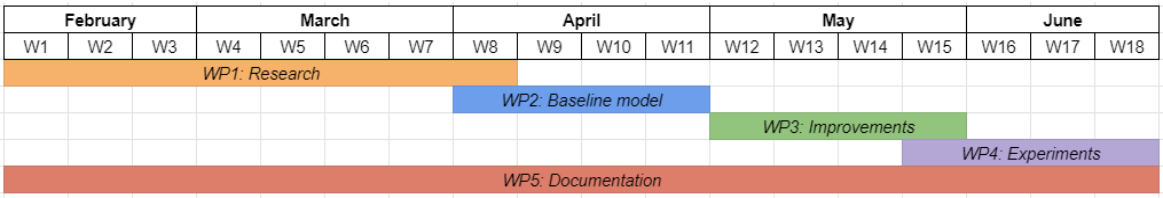

Figure 1.2. Gantt diagram

Some tasks were done in parallel with others, for example preliminary testing for the improvements was done in unison with the design of the final experiments, and training of the model was done throughout the project, as well as the documentation process and different deliverable documents.

1.4. Structure of this study

This study is divided into 5 sections in order to explain the work done: In section 2, an overview of the state of the art of SER is given, explaining several theoretical concepts and background information about the subject matter, while referencing recent work and providing an overview of current development. In section 3, the project methodology is explained, where the model and dataset used are described in detail, followed by an explanation of the procedures done in order to implement the different attention mechanisms and the multi-task learning approach. Next, in section 4, the experiments, tests and results are presented in an ordered fashion, accompanied by the data analysis performed and plots needed to explain key observations on these results. Following that is section 5, an analysis of the budget and costs of the project before concluding with section 6, were final conclusions of the project and closing statements are given, providing a general review of the project as well as personal thoughts and opinions.

2. State of the art

The goal of SER system is to identify the user's emotional state using only their speech. SER is a relatively new field of study inside of speech recognition. It has been around for about 20 years and has already proven to have extremely useful applications in human-computer interaction. Applications that require *natural* interaction between human and machine also have the intrinsic need to know in which emotional state the speaker is in, in order to respond accordingly and provide a realistic, more human or, in some cases, even safer response. Applications nowadays range from voice assistant interaction, call center user experience, as a diagnostic tool for therapists, automatic translation systems to In-car board systems, where information about the emotional state of the driver can be used to initiate his/her safety. An example is Affectiva's AUTO AI platform, which uses video and audio (speech) information from the driver to recognize their mood and adapt the cabin's environment accordingly.

The purpose of this section is to give a general overview of the research that is taking place nowadays in the task of SER, centering around the use of DL techniques and the different mechanisms that are applied in this project, referencing recent studies done on the subject. This includes effective architectures for the task, current common datasets and their properties, common speech features for SER, attention based pooling mechanisms and an introduction to multi-task learning and its recent implementations in the field of SER.

2.1. Emotion classification

Although it has many applications, emotion detection is a challenging task, because emotions are subjective. The interpretation of emotions, or how to classify them, is not trivial, and requires assessment depending on the task one wishes to develop. There is no objective or scientific way of classifying human emotion, which emotional states exist, and how distinct they are from each other. It is an intrinsically biased and subjective task.

A common interpretation is one of six archetypal emotions, or basic emotions, from which all emotional states can be derived [6]. The archetypal emotions are Anger, Fear, Disgust, Surprise, Joy, Sadness and often an added Neutral state.

Other researchers define emotional states as having continuous values in independent dimensions. The definition of these dimensions and the number of them is an area of research, but a widely used representation is the VAD scale, or the Arousal, Valence, Dominance scale. It defines an emotional state as having a specific value for each of these dimensions. Arousal references the intensity of emotion provoked by a stimulus. Valence, on the other hand, characterizes the pleasantness of a stimulus, positive vs negative. Finally, dominance intends to characterize the degree of control exerted by a stimulus, an active role vs a passive role. The archetypal emotions then have a representation in this 3-dimensional space, as shown in figure 1.

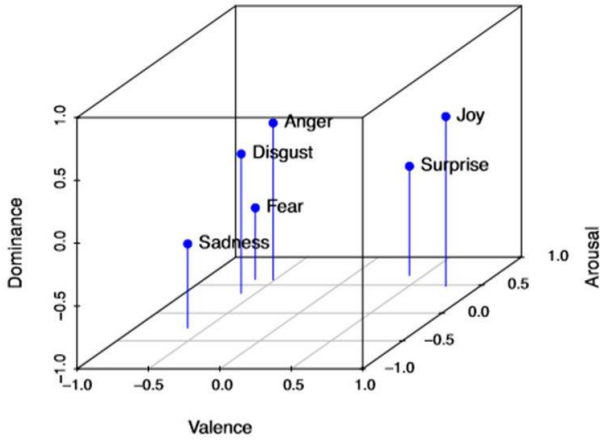

Figure 2.1. Archetypal emotions in the VAD scale

Studies suggest these dimensions are in fact not orthogonal and dependencies between them can be exploited for SER [7].

2.2. Datasets

There are many factors that determine the selection of the correct database for SER. One aspect to consider is the labelling of the dataset (the emotions being classified). Depending on the classification task, one will prefer to use a database that is labelled into the archetypal emotions (anger, joy, sadness, disgust, fear, surprise), or one that classifies into varying levels of stress, like the *SUSAS* dataset [8].

It is also important to consider the nature of the recorded utterances, that is, whether the emotions recorded are from real-life situations or if they are simulated by actors. A spontaneous (or natural) emotional speech source will offer the most realistic representation. The *MSP-Podcast Corpus* is an example of a naturalistic dataset: the recordings are taken from a live podcast environment and are completely non-acted. It is widely used in SER for this reason. Unfortunately, a naturalistic source is hard to come by, and many databases are constructed by hiring professional actors to elicit emotional sentences in sound laboratories to overcome this issue. This is the case for *INTERFACE*, the dataset used in this project.

Furthermore, distribution of the utterances over the labelled emotions can be uniform or can reflect real world cases, in which neutral emotions are much more present. This is then reflected in the database. An example of this is *INTERFACE*, as is explained further in this document, in section 3.1

Lastly, size, language and access to these databases (private, public access, access with a license fee…) are also important factors to determine the database that is going to be used for the specific SER task. A compiled list of modern databases used in SER is given as an appendix in page 50.

2.3. Feature extraction

The feature extraction process is an important part of a modern SER system. The objective is to find an appropriate set of features, extracted from the voice signal, that will successfully characterize the different emotions. A proper selection of features will significantly improve the classification accuracy and is an ongoing field of research. The trend nowadays [9] is to consider features belonging to 4 categories: spectral features, prosodic features, voice quality features and TEO(Teager Energy Operator)-based features [10].

Spectral features are the most widely used speech features because they convey critical information about the vocal tract and the power spectrum of the speech signal. Several studies show that the distribution of energy in the spectrum of a signal is affected by the emotional content of the utterance. For example, Banse and Scherer [11] reported that utterances with happy emotional content show higher energy in the high frequency range of the spectrum, while sad emotional content shows lower energy in that range. The most used spectral features in SER are those related with the Mel scale, i.e the MFCCs (Mel-Frequency Cepstral Coefficients) or the Log-Mel Spectrogram [12].

Log-Mel Spectrogram

For the purpose of this project, we consider the Log-Mel Spectrogram as the input of our model. To understand the Log-Mel Spectrogram we must introduce the Mel scale. The Mel scale is a non-linear transformation of the frequency scale. It is constructed in such a way that sounds at equal distance from each other in the Mel scale also sound to humans at the same distance to one another. The difference in perception between 500 and 1000Hz to humans is not the same as from 8000 to 8500Hz, for example, and the Mel scale reflects this. It can be understood as the scale in which humans hear. Graphically, it is represented as follows:

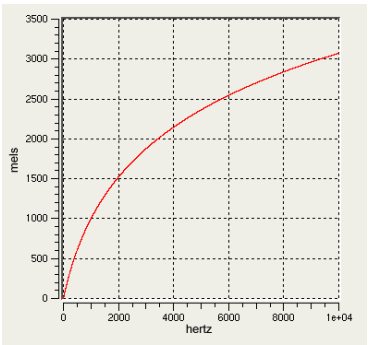

Figure 2.2. Mel scale

The regular spectrogram consists in dividing the signal into frames and taking the Fourier transform of each frame. The difference between the spectrogram and a Mel-spectrogram is that the Mel-spectrogram is the result of applying a Mel filter bank to the frequency response of each frame. If we use N mel bands, that is N filters in the mel scale, we will obtain a spectrogram of size N in the vertical axis. We can now obtain the log-Mel spectrogram by applying the logarithm to this spectrogram.

2.4. Deep Learning classifiers

The majority of recent work in SER is centered around DL techniques and experiments with different architectures. The conventional approach is to model each class by a probability distribution based on the available training data. These classifiers, also known as *statistical classifiers*, have been used in many speech recognition applications; various types have been used for the task of SER, such as GMM [13], HMM [3] and i-vectors [1]. For the interest of this study, we consider GMM and two variations of the i-vector approach: i-vectors combined with Cosine Distance and with Probabilistic Linear Discriminant Analysis (PLDA), used in a 2017 degree thesis [14].

From the success of Deep Learning in various fields, multiple architectures have emerged as effective models across different tasks. Recurrent architectures such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) RNNs have been effective in natural language processing [15] [16], but feed-forward architectures such as Convolutional Neural Networks (CNNs or ConvNets) have been particularly useful in image and video processing and can be successfully used for speech emotion recognition [12].

Convolutional Neural Networks

Convolutional Neural Networks are the state-of-the-art architecture in image and video processing since the input for these networks are grid-like matrices, i.e., images. The processing stages provide the architecture the ability to capture temporal and spatial dependencies from an input source. The inputs are reduced into a form without loss of features so that computational complexity decreases, and the success rate of the algorithm is increased. A basic CNN consists of 3 layers: a convolution layer, a pooling layer and a classification or fully connected layer. The architecture of a classic CNN is shown in figure 2.3.

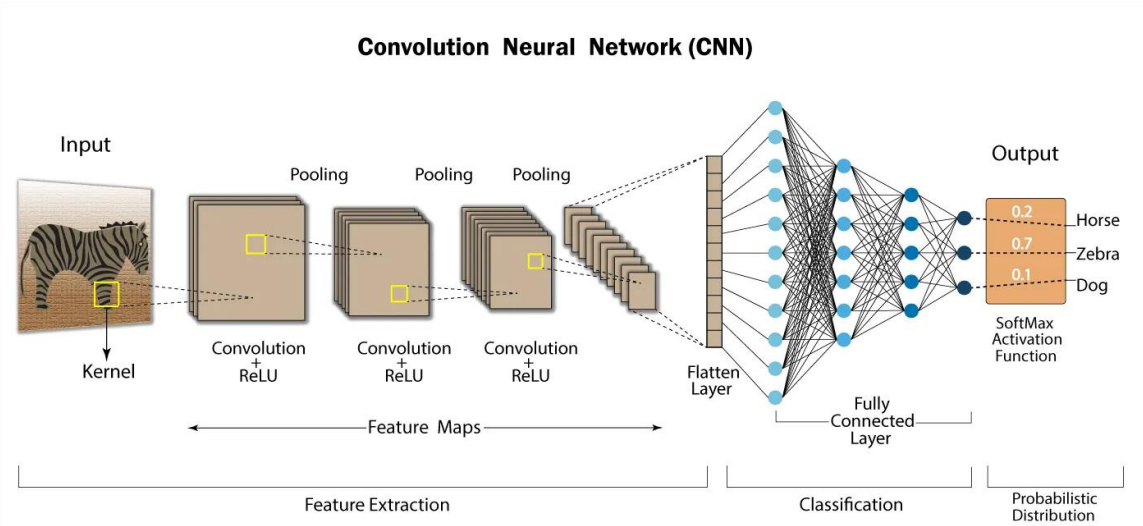

Figure 2.3. Architecture of a classic CNN

The way the convolution layer works is by passing the input through several convolution stages with filters whose weights are trainable parameters. The size of the filters is called the *kernel size.* After a convolution step, the output is composed of K different feature maps or channels, with K being the number of filters used. After that a MaxPooling operation of a certain size can be performed to each channel, in which the size of the image is reduced by taking the maximum value in a region. Lastly a ReLu activation function is applied at the output before passing to the next convolution stage.

Several pooling operations are done throughout the network (MaxPoolings), but a general pooling layer is commonly added at the output of the convolution layer and before classification. This pooling layer works by taking the output of the convolutions and performing statistical analysis or other operations to reduce the dimensionality of the features and flattening the output before classification. Pooling methods are explained in section 2.5 and it are one of the central themes of this project.

The final classification stage is a Fully connected layer composed of several dense layers, with each node connected to all of the nodes from the following layer. The output of this stage is passed through a SoftMax function to obtain the final probabilities for each class. A special variation of the SoftMax function is used for this project, called the Angular Margin SoftMax (AMSoftMax) [17], which provides better classification results by limiting the decision region of the SoftMax. This variation has recently gained traction in the field of speech recognition for its improvement over the original SoftMax and is used successfully in the task of speaker verification in [4] [5].

2.5. Attention-based pooling mechanisms

Pooling of features is the process where the dimensionality of the feature space is reduced for the purpose of compression, simplification and generalization of the data. Several methods exist, but regarding this project we consider 2 general types, statistical pooling and attention-based pooling mechanisms. In particular, for the attention-based mechanisms we consider Self-Attention, Self-Multihead Attention and Double Multihead Attention. These are newly proposed mechanisms that have an underlying functionality derived from attention mechanisms first used in Neural Machine Translation (NMT).

In the past, seq2seq (sequence-to-sequence) models were the state-of-the-art approach to translating text, but they had a limitation, and that is that the context vector, or alignment between input and output sequence, was fixed-length and taken from the output of the encoder's last hidden state. This representation intended to be a good summary of the meaning of the *whole* source sequence but was incapable of remembering long sentences. Rather than building a single context vector out of the encoder's last hidden state, Attention [18] assigns a weight to each input sequence through a SoftMax layer, giving a score of how well query and target state match. The resulting context vector is then constructed as a weighted sum of the input states and takes into consideration all the input sequences, in other words, it is capable of searching for the most relevant encoded states in the input sequence in order to predict the output sequence.

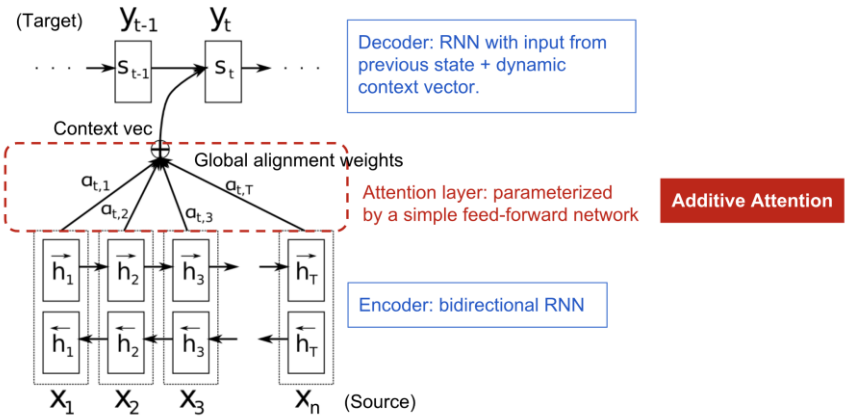

Figure 2.4. Attention mechanism proposed in [19]

Self-Attention pooling

Self-attention, or self-attentive pooling, was initially proposed in 2018 in the task of text-independent speaker verification [19]. The purpose was to create a trainable and more adaptable pooling mechanism than averaging the frames over time.

Inheriting from the attention mechanism explained above, the difference between self-attention and attention is that self-attention is self-referential, meaning that target and query state both refer to the input hidden states from the encoder. By assigning a trainable target state u of the same size as the input states, we can obtain the weights for each state by a SoftMax layer that will give us the level of similarity or importance of each frame. The result is again a weighted sum of all the hidden states, giving us a final representation that takes into consideration all the frames, therefore acting as a pooling layer.

Mathematically, if we consider a sequence of hidden states $h_t = [h_1, h_2, \dots, h_N]$ with $h_t \in \mathbb{R}^D$, and a trainable $u \in \mathbb{R}^D$, we are able to define an attention weight w_t for each element of the sequence through a SoftMax layer:

$$w_t = \frac{\exp\left(h_t^T u\right)}{\sum_{l=1}^{N} \exp\left(h_l^T u\right)}$$

Now we can compute the final pooled representation or *context vector* c as the weighted sum the hidden states.

$$c = \sum_{i=1}^{N} h_t^T * w_t$$

Self-Multihead Attention pooling

Created in 2017 by Google, Self-Multihead Attention [20], or simply referred to as MHA, works by splitting the encoded representation into sub-vectors called *heads* and performs single self-attention to each head. This way each head obtains a set of weights with which to perform the weighted sum to obtain their respective context vector. The final pooled representation is the concatenation of the head context vectors:

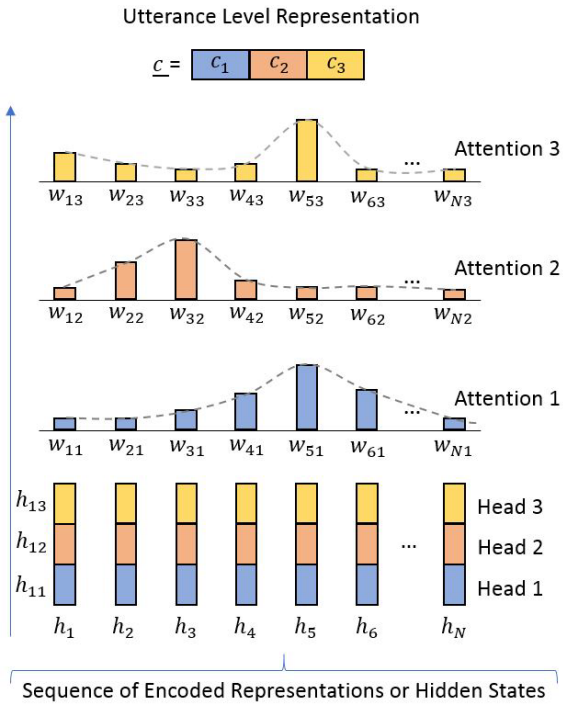

Figure 2.5. Example of MHA pooling with 3 heads [5]

In the reference study [4], MHA with 64 heads is used for the task of Speaker Verification and compared to previous methods used for the same task. They obtained an EER (Equal Error Rate) improvement of 58% in respect to i-vectors and 18% in respect to vanilla statistical pooling.

Double Multihead Attention pooling

Double Multihead Attention (DMHA) was proposed recently by Miquel Àngel India [5] as an improvement over MHA in the task of Speaker Verification. The operation can be described as MHA followed by self-attention. In DMHA, self-attention is applied over the head context vectors obtained from MHA, instead of concatenating them, obtaining the final context vector as a weighted sum of head context vectors.

With this method, the final context vector is created scaling the information of the most/least relevant heads. In the study, DMHA was compared to MHA and single self-attention in the task of speaker verification and obtained a generalised improvement, most notably DMHA 32 heads showed a EER improvement of 13,83% in respect to MHA 32 heads.

2.6. Multi-task learning

In Machine Learning (ML), the general objective is to optimize for a particular metric. While this has proven to achieve remarkable results, by being focused on a single task we ignore information that might help us do even better on the metric we care about. This can be achieved by simultaneously training different but related tasks and forcing the system to learn the underlying information that relates them, with the goal of improving the original task performance. This approach is called Multi-Task Learning (MTL).

Multitasking offers advantages like data efficiency, reduction of overfitting, and less learning time by leveraging auxiliary information. In SER, multi-task learning works by designating emotion recognition as a primary task and several other tasks such as gender, spontaneity and naturalness classification are selected as auxiliary tasks. Jaebok Kim, et al. proposed a system that uses MTL and assigns gender and naturalness as auxiliary tasks, obtaining an improved performance compared with single-task learning [21]. In this thesis, language recognition is considered as an auxiliary task in order to aid in the main task of emotion recognition.

Having defined the theoretical motivations for MTL, two common approaches are taken to perform multi-task learning in deep neural networks, *hard* and *soft* parameter sharing. For the interest of this project, we consider hard parameter sharing.

2.6.1. Hard parameter sharing

Hard parameter sharing is the concept of sharing the feature extraction layers of the model (convolutional layers in the case of CNN) between tasks, while having separate classification stages for each task (see figure 2.6). The classification stages can be fully connected layers or separate SoftMax functions, and are in charge of defining the class probabilities for each task. Hard parameter sharing greatly reduces the risk of overfitting [22]. This makes sense intuitively: If we are learning tasks simultaneously, the model has to find a representation that serves all the tasks, and consequently reduces the chance of overfitting on our original task.

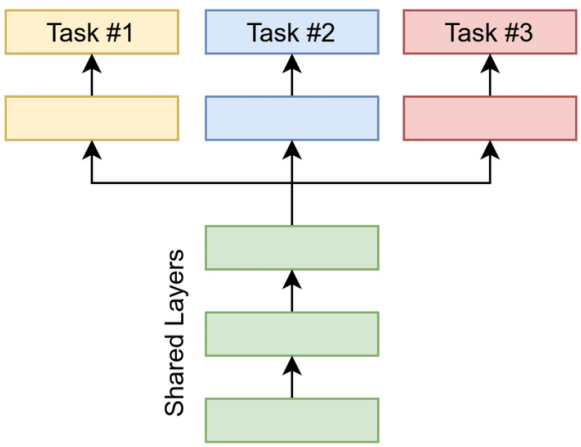

Figure 2.6. Layer distribution in hard parameter sharing

2.6.2. Loss functions

When we talk about MTL, we are considering an optimization problem involving more than one loss function, one for each task. But in order to perform backpropagation throughout the layers, we need to define a general loss function that reflects all the tasks. There exist several ways to do this, but the most common approach is to define the general loss function as a linear combination of the losses from each task:

$$L = \sum_{i=1}^{M} L_i * \alpha_i$$

Where L_i is the loss associated to task i from a total of M tasks, and α_i is a weight parameter set experimentally. Li et al. [23] used MTL and self-attention for the main task of emotion recognition and considered gender recognition as an auxiliary task. Different experiments were performed with different weight values to obtain the best results. The results were a 7,7% absolute improvement in classification accuracy in respect to previous methods without multi-task learning.

3. Project development and methodology

This next section is dedicated to describing the methodology used for the development of this project. Firstly, the dataset is described to contextualize the data we're working with. Next, description of the model used, including the architecture and distinct layers within, and the proposed modifications to this model. What follows is a brief description of the software used, for repeatability and context.

3.1. INTERFACE dataset

As mentioned in section 2.2, the dataset used for this project is *INTERFACE*. *INTERFACE* is the result of a joint effort between the EU and a large number of European universities, including UPC, as a part of the IST project Interface. It was developed in coordination with TALP between 2000-2002.

The database is designed for general study of emotional speech as well as automatic emotion classification purposes. The dataset contains recordings in 4 different languages: Slovenian, English, Spanish and French, distributed uniformly as shown in figure 3.1. For each language, there are 170-190 sentences spoken by two different actors, one male and one female, except for Spanish where two male actors and one female are used. Each sentence is spoken in seven different styles, for a total number of 24197 recordings. The styles (emotion labels) are: Anger, Sadness, Joy, Fear, Disgust, Surprise and a Neutral style with different variations depending on the language. For Spanish, for example, there are 5 variations: neutral-normal, neutral-soft, neutral-loud, neutral-slow and neutral-fast. For Slovenian and English, 2 variations: neutral-soft-slow and neutral-loud-fast. Finally for French, 3 variations: neutral-normal, neutral-soft-slow and neutral-loud-fast. This over-presence of a neutral style is to give a reference to emotional speech and reflects the emotionally neutral style of the collection of texts from which the sentences were extracted [24]. For the purpose of classification, all these styles are considered as one general "Neutral" style. The distribution of emotions is shown in figure 3.1. Apart from the neutral style, the rest of the emotions are distributed uniformly throughout.

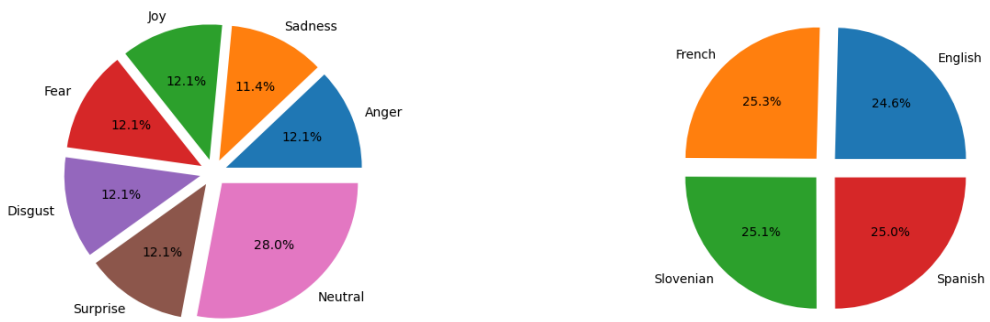

Figure 3.1. Distribution of the dataset: emotions (left) and language (right)

The sentences uttered range from digits and numbers, isolated words, sentences of different length to paragraphs texts, and the mean length of the audio files is 4,85 seconds. All the files are in WAV format. A histogram of the file lengths is shown in figure 3.2.

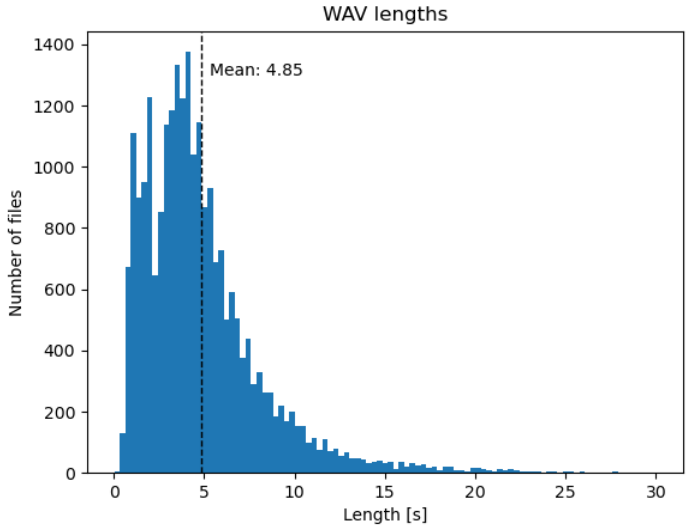

Figure 3.2. Histogram of file lengths in the dataset

3.2. Model

The model used in this project is a modified version of a model inherited from a previous study [5] in which a CNN architecture was used for the purpose of Speaker Verification. It was generously provided by the author of the aforementioned study and the tutor of this thesis. The first step was to adapt the model for a SER task. What follows is a description of the architecture from the Speaker Verification model and the subsequent modifications done in order to achieve the baseline used for emotion recognition.

.

3.2.1. Baseline architecture

The baseline model used for this project consists of 3 main blocks: Front end, a Pooling Layer and a Fully Connected block, shown in figure 3.3. The front-end block is in charge of extracting the relevant information from the input spectrogram, which then is passed through a pooling layer that flattens the channels and reduces the dimensionality of the feature space before classification through the fully connected block. Finally, a SoftMax function transforms the embedding at the output of the FC block into probabilities for each class:

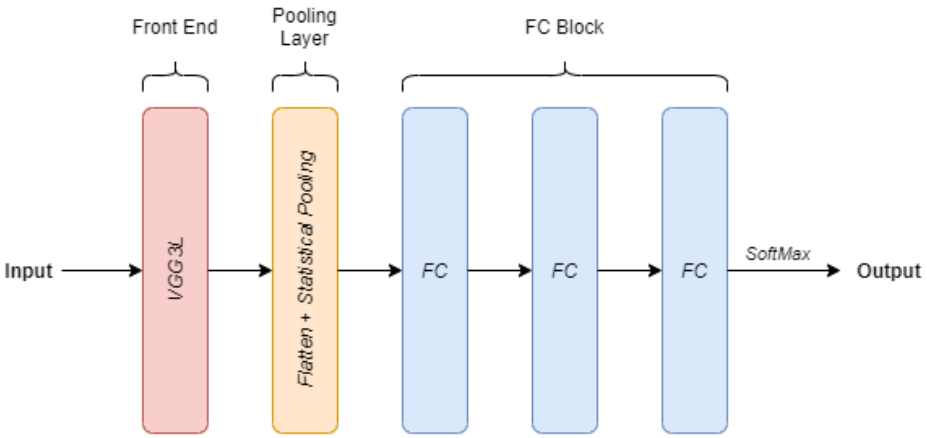

Figure 3.3. High-level diagram of the model

Front End

The Front End is an implementation derived from the vgg16 architecture proposed by Karen Simonyan and Andrew Zisserman in 2014 [25], a model that has recently gained traction and is now staple in the field of image processing as it significantly outperforms previous models and has low implementation complexity.

VGG3L is the name of the variation used for this project. It consists of 2 hidden convolution layers with a small 3x3 filter of stride 1, combined with a MaxPooling operation over a 2x2 pixel window, with stride 2. This pooling operation reduces the image size by 2 in both dimensions. All hidden layers are equipped with the rectification non-linearity (ReLu). This whole stack is then repeated 3 times, hence the name *VGG3L* (3 layers). The number of channels is multiplied by 2 at each outer layer, giving 128 channel depth at the first layer, 256 at the second and 512 at the third. figure 3.4 shows a diagram of the described architecture.

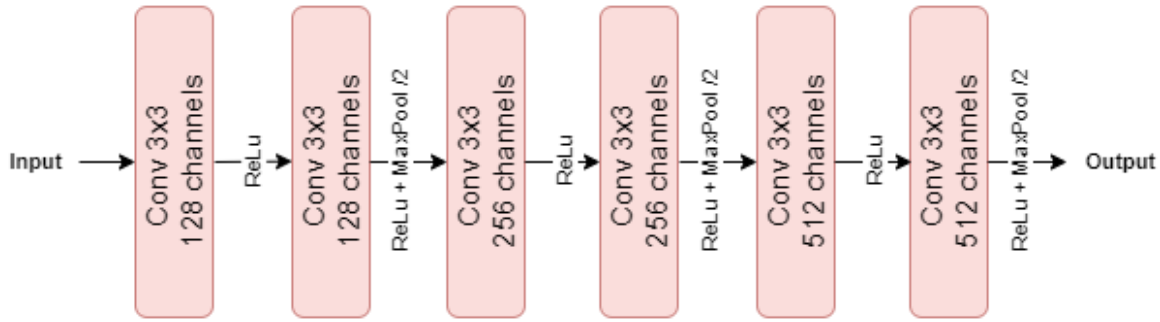

Figure 3.4. Diagram of the VGG3L Front End architecture

The idea behind using this model for Speaker Verification (and later Emotion Recognition) is to feed it a spectrogram as the input image, in the case of [5] as well as our case, we use the log Mel Spectrogram as the input image, with 25ms length Hamming windows and 10ms window shift. The model accepts variable length input of size N (in frames) and variable the number of Mel bands M used in the filter bank. The input resulting image therefore is of size NxM. For our project we have fixed the number of Mel bands to M=80, so our input images are of size Nx80.

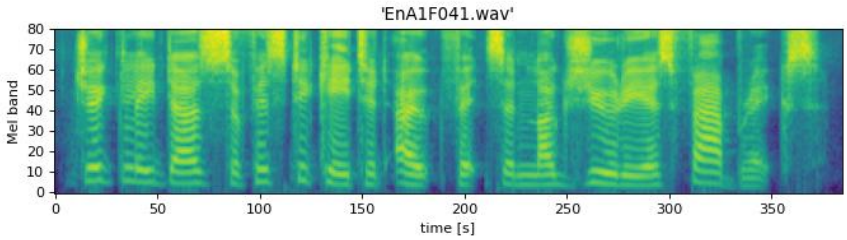

Figure 3.5. Example of an input image, of size 385x80

With this in mind, the result at the output of the front-end block is a matrix of size Cx(N/8)xM/8, C being the number of channels, N the frames and M the Mel filters. The next step is a flattening and pooling, as the FC block requires a 1-dimensional vector as input.

Statistical Pooling Layer

Before Pooling and the FC block, a flattening stage concatenates all the channels at the output of the front-end block to create a matrix of size (N/8)x(M*C). This allows the next pooling stage to perform statistical pooling over the frames (over the N dimension). Statistical pooling averages the frames and calculates their standard deviation to obtain a final representation that is the concatenation of both results. In other terms, it reduces the (N/8)x(M*C) matrix to two vectors of size 1x(M*C), containing the mean and standard deviation of the frames respectively, and then concatenate these two vectors obtaining a single vector of size 1x(M*C*2).

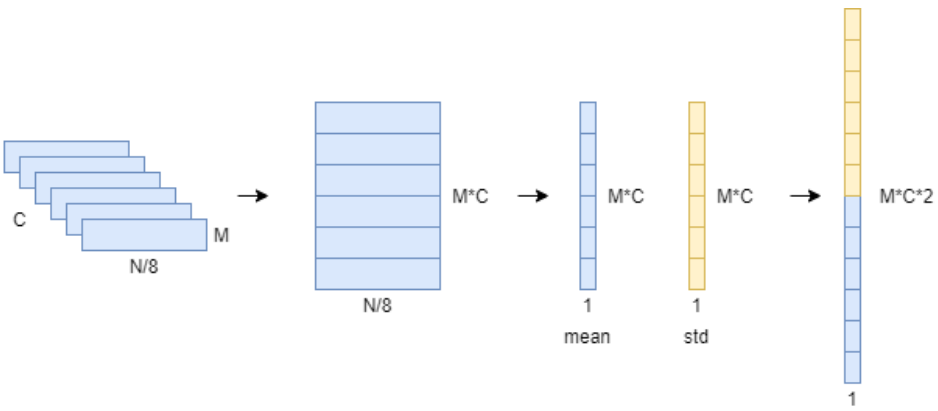

Figure 3.6. Flattening + statistical pooling

Fully-Connected block

The last stage of the model is the classification stage, in which a prediction is obtained from the output of the pooling layer. This is done through a fully connected block, consisting of three fully-connected layers: one input layer of size 1xM*C*2, and two hidden layers of size 400. The first two layers consist of a ReLu activation and a batch normalization function. Finally, the AMSoftmax function is applied over the output of the last hidden layer to obtain a probability for each speaker and to obtain a speaker classification.

For the task of speaker verification, the embedding of the second hidden layer is extracted, but this task does not affect the project as we are interested only in the classification task, as is explained below.

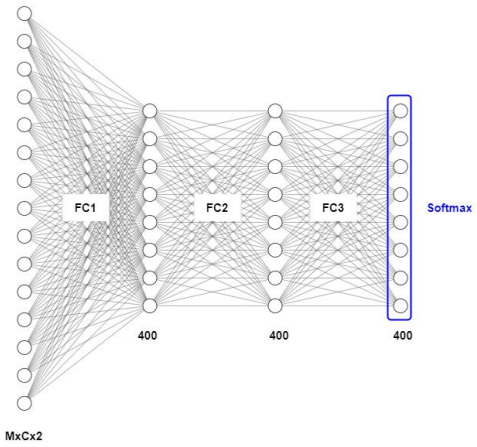

Figure 3.7. FC layers in the model

3.2.2. Adaptations for Emotion Recognition

The described model can be abstracted as a general classifier. For the task of emotion recognition our classes are anger, sadness, joy, fear, disgust, surprise and neutral. Now, the task of the model is to classify within these labels in the same way as it would classify within a set of speakers. In order to do this, we have to change the classification layer or FC block to output the prediction probabilities of 7 classes. This is done by changing our SoftMax function at the last layer to output 7 probabilities instead of Nspeaker probabilities.

The dimensions at the output of each layer of our complete model are shown in table 1. Now our model will classify in 7 classes (emotions) instead of N speaker classes, and constitutes the baseline from which to perform modifications for the different experiments.

Layer	Out dim
Input	1xNx80
Conv + ReLu	128xNx80
Conv + ReLu	128xNx80
Max Pooling 2D	128x(N/2)x40
Conv + ReLu	256x(N/2)x40
Conv + ReLu	256x(N/2)x40
Max Pooling 2D	256x(N/4)x20
Conv + ReLu	512x(N/4)x20
Conv + ReLu	512x(N/4)x20
Max Pooling 2D	512x(N/8)x10
Flatten	(N/8)x5120
Pooling	1x10240
FC + ReLu	1x400
FC + ReLu	1x400
FC	1x400
AMSoftmax	1x7

Table 1. Output dimensions at each layer of the final model

3.3. Enhancements with attention-based pooling mechanisms

Once the baseline model has been defined, one of the main goals of this project is to improve the accuracy of the classifier by modifying the pooling layer. The main problem with statistical pooling is that it assumes that all the elements of the sequence must contribute equally in obtaining the final representation. As proposed in [4] [5], we can replace vanilla statistical pooling with attention mechanisms that will act as a pooling layer, in particular Self-Attention, MHA and DMHA. The raw implementations were generously provided by the author of these studies. The following section describes the methodology followed to implement these three attention mechanisms.

Self-Attention

To implement attention-based poolings we need to remove the statistical pooling step and replace it with our attention mechanism. The flattening stage before pooling remains unchanged as it is needed as well.

Given the set of encoded representations from our CNN feature extractor, the flattened vector of size (N/8)xD the output of the front-end, our self-attention mechanism performs a weighted average over the frames to obtain an utterance level representation. The key feature of self-attention is that these weights are derived from trainable parameters.

The first step is to define a trainable vector u of size 1xD containing the trainable parameters, with D being the size of the encoded representations. With this vector and our encoded sequence, we obtain the normalized scalar weight w_t of each frame as the output of a SoftMax function, as explained in section 2.5. Finally, the output context vector c of our self-attention mechanism is the weighted sum of each encoded frame, an utterance level representation of size 1xD.

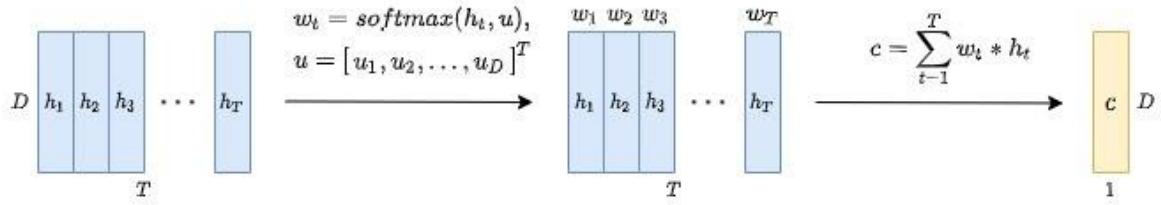

Figure 3.8. Implemented self-attention mechanism. T=N/8, D=5120

With this implementation we have substituted vanilla statistical pooling, which outputs a vector of size 1x10240 with a self-attention mechanism that outputs a context vector of size 1x5120. Because of this it is necessary to perform an extra modification to the FC block, defining the first layer to have the same input size as the context vector.

Self-Multihead Attention

The limitation with Self-Attention is that the attention weights are calculated considering the whole information of the embedding, therefore assuming that all of the important information of the signal must come from the same encoded representations. With MHA we overcome this and are more selective about which information of the embedding is more important.

We again take the output of the front end, the flattened vector of size (N/8)xD and split it into *heads*. If we fix the number of heads to a given number K, we will have to define K different trainable u vectors, one for each head, containing D/K parameters. For example, given that our input is of size (N/8)x5120, if we define 4 heads, we will have 4 different u vectors of 5120/4 = 1280 trainable parameters each. Note that the number of total trainable parameters added to the network (1280*4=5120) is the same as in self-attention and does not depend on the number of heads. In the actual implementation the number of heads is set as a variable, as it is a key feature to experiment with and it affects the classification performance. Several experiments have been done with different head numbers, explained in section 4.3.

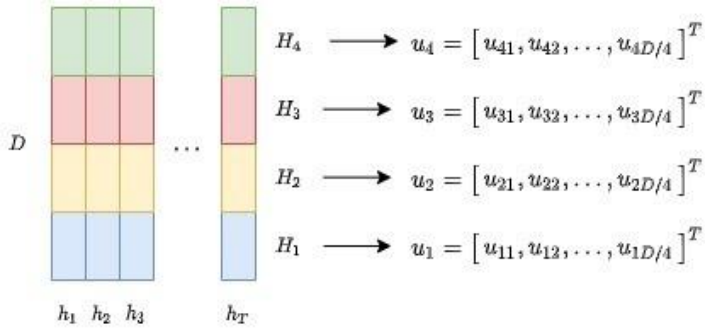

Figure 3.9. Example of the trainable vectors with 4 heads. T=N/8, D=5120

Now that we have a specific u vector for each head, we can calculate the attention scores for each head over the entire sequence. This is done by taking the output of a SoftMax function, in the same manner as in self-attention. The last step is to calculate the utterance level representations, or contexts vector c_i, for each head in the same way as single self-attention, by a weighted sum. The output of the MHA mechanism is the concatenation of these context vectors from all the heads. For example, if we have 4 heads we will calculate 4 different sets of attention weights, each set containing N/8 weights. We perform the weighted sum and obtain 4 different context vectors of size 1x1280 each, and with a concatenated result of size 1x5120. This method allows the network to extract different kinds of information over different regions of the embedding.

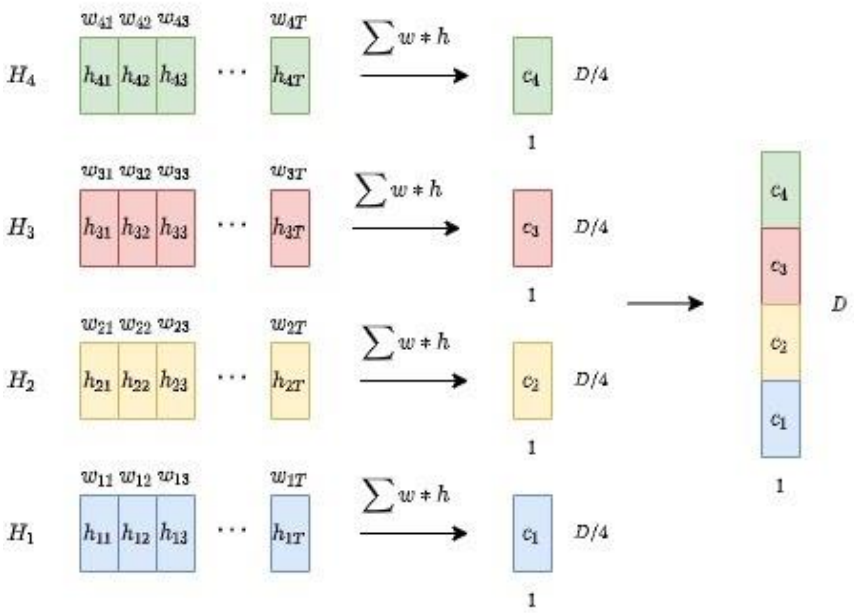

Figure 3.10. Process of obtaining the output of the MHA mechanism. T=N/8, D=5120

The size of the output remains the same as in single self-attention, so the same modifications to the FC block have to be done, changing the first FC layer to have input size 5120 instead of 10240 in vanilla statistical pooling.

Double Multihead Attention

The problem with MHA is that by concatenating each head representation it assumes uniform head relevance. For DMHA, we concatenate a MHA pooling stage with a single self-attention pooling stage. Given our context vectors c_i, we assign new weights to each one of them with the self-attention mechanism described previously. In the example case of 4 heads, we define our new vector u' of 5120/4 = 1280 trainable parameters, so the total number of trainable parameters is increased by 1280 in respect to MHA. Now apply the self-attention mechanism to obtain our 4 weights w_i', one for each head. Finally, the new context vector c is the weighted sum of context vectors c_i and has a resulting size of 1x1280 (5120/4).

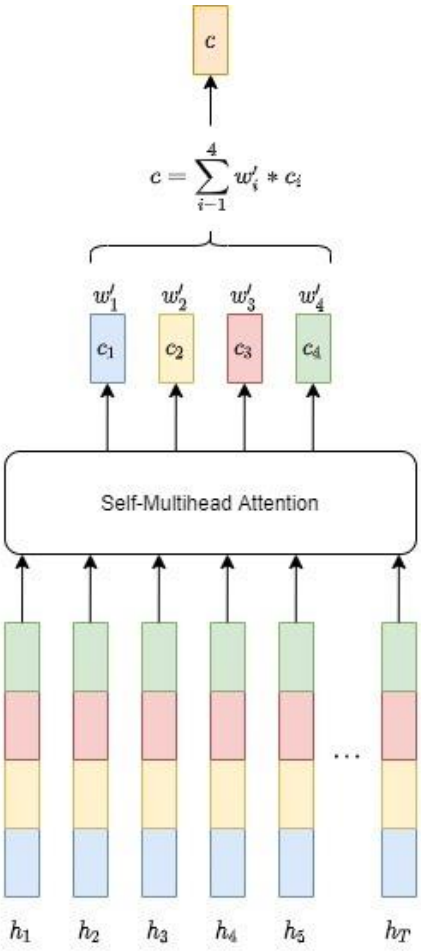

Figure 3.11. Diagram of the DMHA implementation with 4 heads. T=N/8

As it shows, the number of heads defines the dimension of the output context vector, so in the same way as in the previous implementations we need to modify the first layer of the FC block to have the correct input size. In this case, the size will be $5120/K$, where K is the number of heads used. The size of the heads is a key feature in DMHA, much like in MHA.

3.4. Enhancements with multi-task learning

In this stage of development, we consider using language recognition as an auxiliary task in order to improve the main task of emotion recognition. Emotions are expressed with different intonations, speed and inflections depending on the language, so it makes intuitive sense to consider language information in a SER task. This section is dedicated to explaining the modifications done to the baseline model in order to perform this MTL approach.

For the auxiliary task of language recognition, our classes will be the four languages present in the dataset: Slovenian, Spanish, English and French. The goal is to get our system to simultaneously predict emotion and language and share learning parameters doing so, but this introduces the question, how deeply should these tasks share a representation in the model? When should they diverge?

For hard parameter sharing, as explained in section 2.6.1, the approach is to have the tasks share the hidden layers, while having task-specific output layers. In our case the tasks will share the entire model up to the last hidden layer of the FC block. In this stage we divide into 2 distinct layers, one for each task, each one consisting of a AMSoftmax function that takes as input the last embedding of size 400 and outputs a probability for each class.

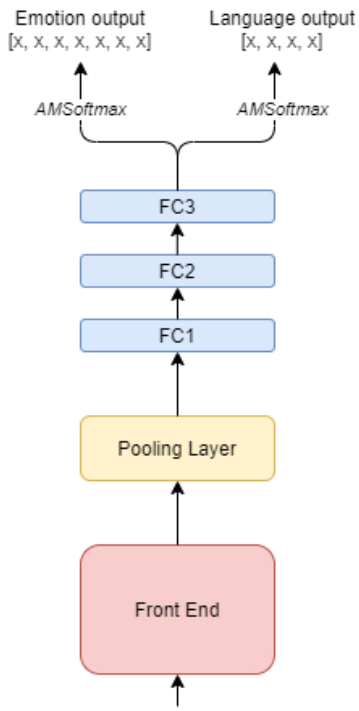

Figure 3.12. Overview of the proposed architecture for MTL

Dealing with multiple tasks entails dealing with multiple loss functions, so the next step was defining a general loss function for backpropagation that takes into account both tasks proposed. The approach taken was to consider the general loss to be a weighted sum of

each individual task loss, as explained in section 2.6.2. The general loss function defined is:

$$L = \alpha_1 * L_{em} + \alpha_2 * L_{lang}$$

Where L is the general loss, L_{em} the loss associated to the emotion recognition task, L_{lang} to the language recognition task, α_1, α_2 are the weights for each loss. These weights are fine tuned in a dedicated experiment, explained in section 4.4, to obtain the best classification results.

3.5. Software

Having explained the project methodology, it is appropriate to give context about the software used in this project. The process followed was to use *git* for version control of the project. A repository for the Speaker Verification was provided, used in a previous final thesis, which contained a branch with the baseline implementation used in [4]. This repository was forked to create a new repository called 'CNNEmotionRecognition' from which three branches were created: A 'v1' branch for the baseline model, with the baseline modifications, a 'Poolings' branch for the attention mechanisms and a 'MultiTaskLearning' branch for the MTL modifications. figure 3.13 shows a diagram of the git flow.

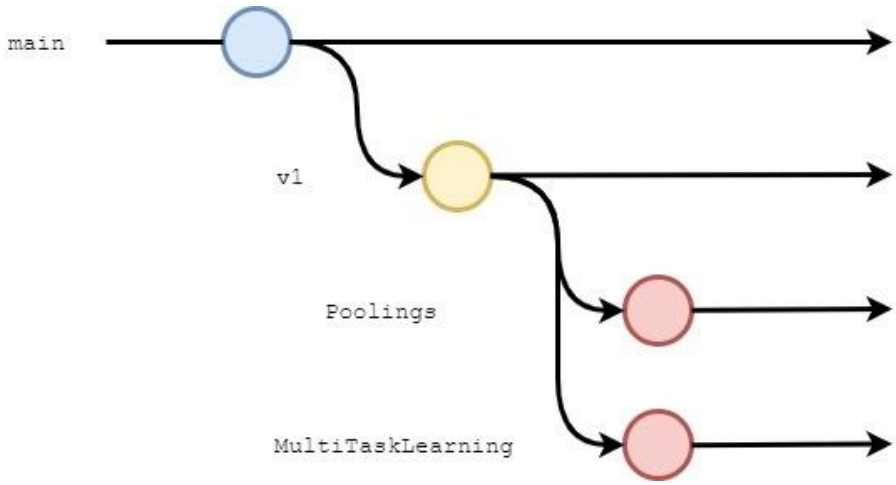

Figure 3.13. Git diagram of the project

Once it was created the repository was cloned into the CALCULA server, from which all the experiments were done. This way all the changes introduced locally were also reflected in the server. For further context, a comprehensive list of all the software elements is given below with an explanation of their usage.

Development:

- *Windows 10:* Operating System used locally.
- *Python 3*: Programming language used for the development of this project
- *Pytorch*: Open-Source Python library for machine learning and deep learning used for the development of the model.
- *Git:* Version control software used to manage the different branches in this project and to push local changes into the remote repository.
- *Visual Studio Code:* IDE used locally to develop the project.

Remote connection:

- *OpenVPN:* Tool used to connect to the TSC VPN network.
- *ssh:* Protocol used to connect to the CALCULA TSC server.
- *byobu:* Terminal multiplexing tool used to perform experiments simultaneously in the same session.
- *slurm:* Workload manager used by the CALCULA server to run scripts, manage devices and queue jobs.

Tables, plots and figures:

- *NN SVG:* Online tool to create Neural Network schematics, used in this project to visualize the fully connected layers.
- *draw.io:* Online tool to create flowcharts and diagrams. Used in this project to visualise architectures, models and processes.
- *matplotlib:* Python library used to create MATLAB-style plots and figures. Used in this project to create plots and visualizations of data and experiments.
- *OverLeaf (LaTex):* Online LaTex editor used to create all the tables in this document.

4. Experiments and results

This section is dedicated to explain in detail the setup of our experiments, the results obtained from these experiments, and insights on these results. The results are given in terms of the test accuracy, which is the ratio of correctly predicted utterances out of all utterances in the test set. In addition, some results are explained using the confusion matrix as a way to obtain a per-class perspective. To compare results, the Relative Error (RE) improvement is used. Dataset partitions, hyperparameters, as well as specific parameters tuned for each experiment are also defined. Before describing the experimental setup, it is appropriate to review some preliminary experiments.

4.1. Preliminary experiments

This section is dedicated to explaining some preliminary experiments performed beforehand as well as experiments that were discarded for different reasons, in order to contextualize and justify the experimental setup used.

The very first tests and results obtained for this project were done using different partitions. At that point in time, language recognition was had not been considered yet as an auxiliary task for MTL and were interested in obtaining results only for the Spanish language. For this reason, the Spanish partition of the dataset was used. An 80/20 split was done containing 4833 utterances for train and 1208 for test and using a 1s temporal window for both. The first test results indicated high performance for statistical (92,3%) and Self-Attention (97,43%) but it was shortly discovered that some mistakes had been made in the procedure. The problem was that the test partition had been used for validation instead of testing. The testing process was not appropriate, these results were obtained from reused validation data that by design rendered the best accuracies, therefore making these results invalid. It was then decided that the correct split that was going to be used from that point on was 70% train, 10% validation and 20% test, making sure that the test partition contained unseen data and the procedure was in fact valid. With the intention of maximizing the amount of data used, the splits were to be done with the entirety of the dataset, including all the languages, and the temporal window was set to the whole length of the file for validation and testing.

Another battery of tests was designed in which a 3s temporal window was used for training instead of 1s. While an increase in accuracy was obtained for MHA (all head numbers) and DMHA (8 and 16 heads) in respect to statistical pooling, there was a generalized drop in accuracy compared to using a 1s window. This was a puzzling result but could be explained by the padding used in the data. To elaborate: in this project, if the file length is shorter than the temporal window, padding is used by repeating the fragment until covering the whole window. If a 3s window is selected, that means that all the audio files with an inferior length will be padded by repetition. Considering that 3s corresponds to the 30,71% percentile of the dataset lengths, 30,71% of files will contain repetition. A 1s window corresponds to 4,22% of files containing repetition, or the 4,22% percentile. Both values are visually represented in the histogram of the file lengths in figure 4.1:

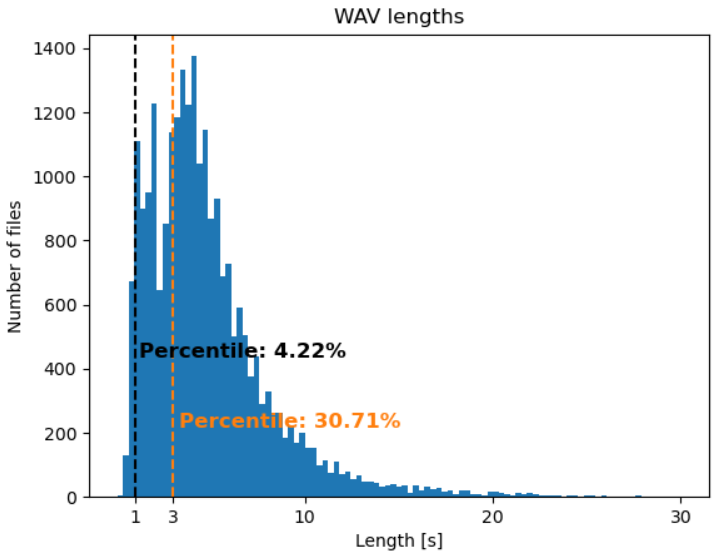

Figure 4.1. Percentile values for 1s and 3s, represented in the histogram

Therefore, if a 3s window is used, we will have a large quantity of files will have repeating frames and consequently the attention weights will be trained using repeated data. This explains the generalized decrease in accuracy for the attention mechanisms. For these reasons, it was decided to exclude these results from evaluation and to train the model with a 1s temporal window.

Lastly, in relation to the multi-task learning experiments, some preliminary tests were done with the task of gender recognition instead of language recognition, considering that the file names also contained the gender of the speaker and could extract the appropriate labels. This proved to be unsuccessful as the results were not indicative enough to extract any conclusions. It was decided to use language recognition instead, as it seemed like a more interesting and potentially more useful approach.

4.2. Experimental setup

To train our model, we split the entirety of the dataset 3 ways: 70% train, 10% validation and 20% test. The splits were done randomly, and the same sets were used for all of the experiments. The final sizes for the split were 16938 utterances for train, 2420 for validation and 4839 for testing.

The way the training works is by validating every epoch and saving the best result. With an early stopping parameter of 5, if the model does not improve the validation accuracy within 5 epochs it stops training and saves the parameters that achieved the best result. This yielded different outcomes for experiments that trained for more epochs than others, but considering that some experiments have a higher computational cost and take more time to converge, it is a better approach than to fix a number of epochs.

For training, batch size is 64 and the number of seconds or temporal window taken from the audio files in training is fixed to 1s. If the audio file is longer than the temporal window, the fragment is taken at random. If it is smaller, repeat padding is performed as explained in section 4.1.

On the other hand, we used a batch size of 1 for validation so we would not have to fix a temporal window and could use the entirety of the audio file. The same was done for testing. As explained previously in section 3.2.1, the number of Mel bands used for the spectrogram was 80. The average training time for each experiment was roughly 1h, depending on the capacity of the server. Finally other hyperparameters include:

- *optimizer:* Optimizing algorithm used for training. 'Adam' was used.
- *learning_rate:* The amount that the weights are updated during training. This parameter fixes the speed with which the model learns. If it is too low, it can cause the training to take too long to converge or to get stuck at a local minimum. If it is too high, it can overshoot and cause unstable training. LR was 0.0001.
- *weight_decay:* Parameter used to prevent overfitting of the data. WD was 0.01.

4.3. Experiment: Attention mechanisms

For this experiment the model with the 3 different attention mechanisms was trained and tested to see if it improved classification performance in respect to our trained model with statistical pooling. The first part of this section describes 11 specific tests: 1 for self-attention, 5 for MHA testing different head numbers, and 5 for DMHA also testing different head numbers. These results of these tests will serve to select the candidate for the final comparative experiment, where the baseline statistical pooling model and the best resulting model from the previous tests are compared to GMM and i-vectors.

4.3.1. Self-Attention

The purpose of this experiment was to test whether the self-attention mechanism proposed in section 3.3 delivers better classification results than vanilla statistical pooling. The intuition was that considering previous success with this methodology in other fields, it could deliver good results. The results of this test are shown in table 2:

Test accuracy	
Statistical	89.98%
Self-attention	89.32%

Table 2. Test results for the self-attention experiment

Self-attention performs close to statistical pooling, but slightly worse. The reason for the underwhelming performance of self-attention could be the short 1s temporal window used, as attention mechanisms benefit from a larger input size. The reasoning behind this is that if the input contains more frames, more attention weights can be assigned. The context vector is obtained with more information, thus the attention mechanism becomes more effective at obtaining a pooled representation.

4.3.2. Self-Multihead Attention

In this next experiment we tested our MHA mechanism with a head number 4, 8, 16, 32 and 64. There were 3 main objectives for this experiment: to test the influence the number of heads has in the classification accuracy, to obtain the optimal number of heads and to see if any given number of heads obtained a better accuracy than statistical pooling. The results are shown in table 3.

Test accuracy	
Statistical	89.98%
MHA 4 heads	89.01%
MHA 8 heads	89.42%
MHA 16 heads	89.32%
MHA 32 heads	**91.09%**
MHA 64 heads	**90.31%**

Table 3. Test results for the MHA experiment

As it shows the best results were obtained with 32 heads, with 91,09% accuracy, an absolute increase of 1,11% in respect to statistical pooling. Notably 64 heads also obtained better results than with statistical pooling, with 90,31% accuracy. As the results show, the number of heads has an influence in the classification accuracy and an optimal value can be obtained. The number of heads in MHA is a trade-off: the context vector sizes are inversely proportional to the number of heads, meaning that more heads will result in a smaller head size and can have a direct impact in the performance of the model. The results of this experiment were the most successful; MHA has effectively obtained a better pooled representation than statistical pooling by being more selective about the information contained in the embedding. This experiment ultimately provided the candidate (MHA 32 heads) to perform the comparative experiments described in section 4.3.4.

4.3.3. Double Multihead Attention

Following the same procedure as in MHA, we tested DMHA with a head number 4, 8, 16, 32 and 64. Additionally we fixed a head drop probability of 0.3 for 16, 32 and 64 heads, meaning that any given head had a probability of 30% to have its weight set to 0. The drop probability for 4 and 8 heads was set to 0.01 because the number of heads is too low for a 30% drop and caused instabilities in training. The test results are shown in table 4.

Test accuracy	
Statistical	89.98%
DMHA 4 heads	89.05%
DMHA 8 heads	86.48%
DMHA 16 heads	87.58%
DMHA 32 heads	89.87%
DMHA 64 heads	82.15%

Table 4. Test results for the DMHA experiment

For this experiment the best result was obtained again with 32 heads. Note that the performance decreases substantially for 64 heads. As stated in section 3.3, in DMHA the higher the number of heads, the smaller our final context vector will be. That means we will obtain a more compressed utterance representation, and this might lead to a worse classification performance. On the other hand, if the number of heads is too low, we obtain a less specific representation which could also affect performance. As you can see, much like in MHA, the number of heads is a trade-off. In this case 32 heads is the optimal number for DMHA. Having said that, this experiment provided worse results than expected, as the best result of 89,87% is inferior to the baseline result with statistical pooling of 89,98%.

While for speaker verification DMHA provides an increase in performance, the compression factor that comes with the DMHA is affecting the model's ability to predict emotion, compared to MHA which does not compress the final context vector. This points to the conclusion that it is not the optimal pooling method for emotion recognition.

Concluding observations

We will now proceed to analyse the confusion matrices of the result with MHA 32 heads and with statistical pooling. A confusion matrix is a useful tool used to visualize the number of predictions of each class versus the actual number of instances of that class, and can be used to extract the per-class accuracy of our system. The i-th row and j-th column entry indicates the number of samples with true label being i-th class and predicted label being j-th class. Therefore the diagonal across the matrix is the number of correct predictions for each emotion. If we normalize this matrix by dividing each cell value by the total number of true instances of that class, we obtain the normalized confusion matrix, with the diagonal being the per-class accuracy. The test accuracy can be retrieved by the sum of per-class accuracies weighted accordingly:

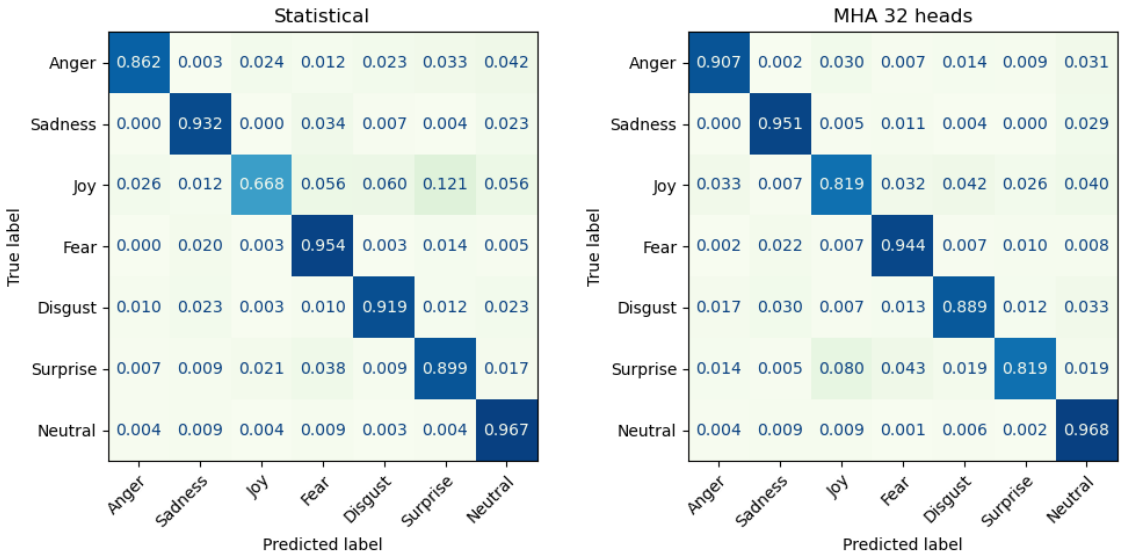

Figure 4.2. Normalized confusion matrices for statistical pooling and MHA 32 heads

As it shows, our MHA mechanism with 32 heads has increased the classification accuracy of 4 out of 7 emotion classes. Most notably the accuracy of 'Joy' has increased from 66.78% to 81,90%, a remarkable result. 'Fear', 'Disgust' and 'Surprise' have suffered a decrease in accuracy with the worst case being 'Surprise', specifically from 89,93% to 81,94%. Below (figure 4.3), a bar graph shows another visualization of the accuracy by emotion for both pooling methods, with the rightmost column being the test accuracy, denoted as 'Average':

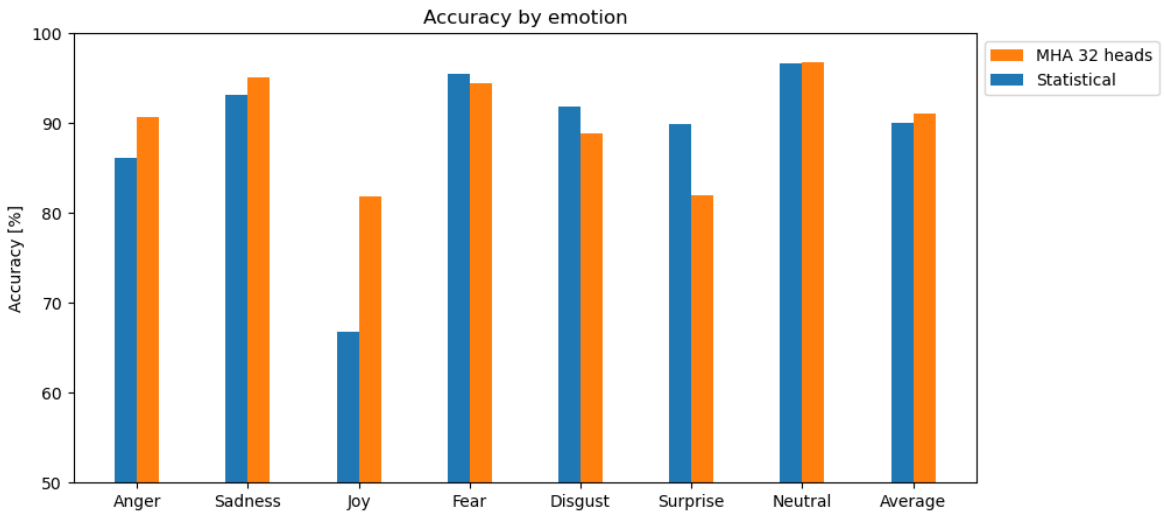

Figure 4.3. Accuracy by emotion, statistical pooling vs MHA 32 heads

In conclusion, these experiments have shown that the best pooling method for our emotion recognition task, out of all of our mechanisms, is MHA with 32 heads, with a test accuracy

of 91,09% in respect to 89,98% with statistical pooling. If we consider how much the relative error has improved:

$$RE_{improvement} = \left(1 - \frac{TestError}{BaselineError}\right) * 100$$

it represents a notable RE improvement of 11,08%. A good result was also obtained with MHA 64 heads, with a RE improvement of 3,29%. The rest of variations did not show a RE improvement in respect to statistical pooling.

RE improvement	
Self-attention	-6.59%
MHA 4 heads	-9.68%
MHA 8 heads	-5.59%
MHA 16 heads	-6.59%
MHA 32 heads	**11.08%**
MHA 64 heads	**3.29%**
DMHA 4 heads	-9.28%
DMHA 8 heads	-34.93%
DMHA 16 heads	-23.95%
DMHA 32 heads	-1.10%
DMHA 64 heads	-78.14%

Table 5. RE improvement for all pooling methods in respect to statistical pooling

4.3.4. Comparative experiments

For the closing experiment, the statistical and MHA 32 heads models are faced up against previous methods, in particular GMM, i-vectors with Cosine Distance and i-vectors with PLDA by reproducing the test performed in [14], to test if the implementations developed in this project provide better performance and if the research has proven successful. Although the reference study used the same dataset, the results are not directly comparable, considering that the models been trained and tested with different partitions. Specifically, the reference study used a partition containing 555 test utterances, while our partition contained 4839. Moreover, they were trained using only the Spanish language partition, therefore making the results incomparable. Luckily, we managed to obtain the specific partitions used for the study and could proceed to test and train our models.

The original test and train split from the study was 80/20, but because of the way our models were trained we needed to split in 70/10/20 like the previous experiments. This way we obtained a split containing 1950 utterances for train, 278 for validation and 555 for testing, with the same utterances for testing as the reference study. The results for this test are shown in a bar plot below (figure 4.4) as the rightmost column, as well as the accuracies per emotion obtained from the normalized confusion matrices:

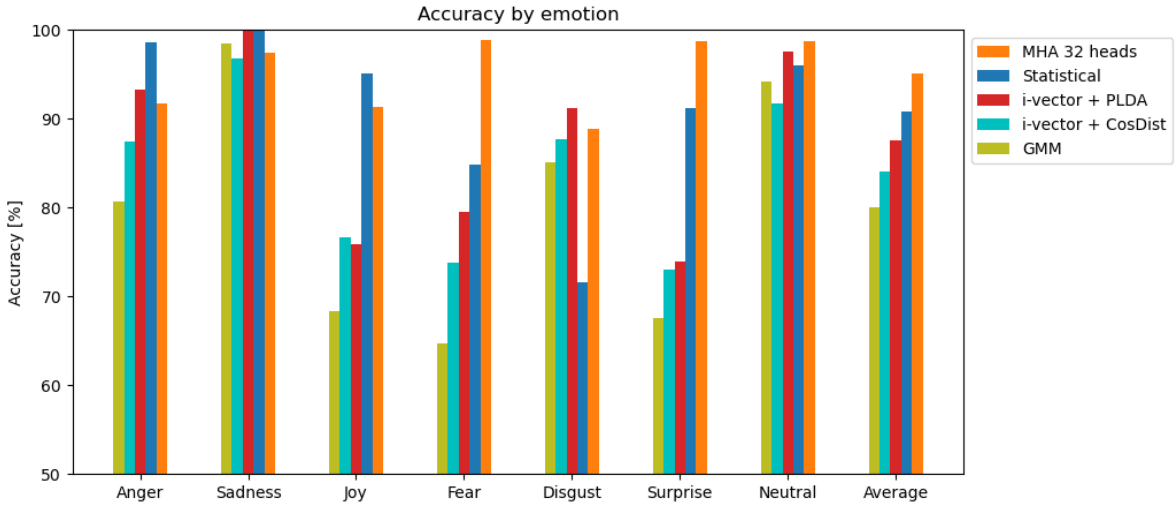

Figure 4.4. Accuracy by emotion, all considered methods

If we analyse the accuracy per emotion, a remarkable performance for MHA 32 heads has been achieved for 'Fear' (98,84%), 'Surprise' (98,73%) and 'Neutral' (98,68%), outperforming all other methods. Statistical pooling has achieved the best performance out of all the methods for 'Anger' (98,61%) and 'Joy' (95,06%), and obtained 100% accuracy for 'Sadness', equalling the result for i-vectors + PLDA for this class. I-vectors + PLDA did achieve better results for 'Disgust' (91,23%), which shows why it is considered a powerful method. A table with the final results is given below (table 6), in ascending order, accompanied by the RE improvement in respect to GMM, i-vectors + PLDA and in respect to the previous best method:

Method	Test accuracy	RE improvement		
		GMM	i-vectors + PLDA	previous
GMM	80.09%	-	-59.54%	-
i-vectors + CosDist	84.03%	19.79%	-27.96%	19.79%
i-vectors + PLDA	87.52%	37.32%	-	21.85%
DL + Statistical pooling	90.81%	53.84%	26.36%	26.36%
DL + MHA 32 heads	95.14%	75.59%	61.06%	47.12%

Table 6. Test results with RE improvement for all methods

We can observe that both statistical and MHA 32 heads provide a solid increase in general performance in this test in respect to previous methods: GMM, i-vectors + Cosine Distance and the most effective method used in the reference study, i-vectors + PLDA. Statistical pooling offers 26,36% RE improvement over i-vectors + PLDA and 53,84% over GMM. MHA 32 heads has provided the best results with a remarkable 61,06% RE improvement over i-vectors + PLDA and 75,59% over GMM. These results indicate that we have successfully managed to top the best methods used in the reference study with both approaches.

4.4. Experiment: Multi-task learning with language recognition

In this next experiment we considered language recognition as an auxiliary task, and tested our model with the modifications proposed in section 3.4. We considered our baseline model with statistical pooling, no attention mechanisms were used. This was done to test whether our MTL approach provided an increase in performance, independently from any other modifications. In this experiment different weight values for the loss functions were tested with the restriction that $\alpha_1 = 1 - \alpha_2$. Our selected values for α_1 were 0,5, 0,7, 0,9, 0,95, with α_1 the loss associated with the emotion recognition task. With these values we gradually give more importance to the emotion recognition and less importance to language recognition. The validation accuracy and test accuracy are in reference to the emotion, as it is the metric we are interested in. The results obtained were:

Test accuracy	
Baseline (No MTL)	89.98%
$\alpha_1 = 0.5, \alpha_2 = 0.5$	89.30%
$\alpha_1 = 0.7, \alpha_2 = 0.3$	87.42%
$\alpha_1 = 0.9, \alpha_2 = 0.1$	89.50%
$\alpha_1 = 0.95, \alpha_2 = 0.05$	89.23%

Table 7. Test results for different weight values

The best result obtained was with $\alpha_1 = 0{,}9, \alpha_2 = 0{,}1$, with an accuracy of 89,50%. As it shows, our MTL approach has not improved the performance of our model for any given value of the weights. It is worth noting that the accuracy has not decreased in a substantial way, the performance is close to our baseline model.

Additionally, we performed a test to observe how well the model with trained with $\alpha_1 = 0{,}9$, $\alpha_2 = 0{,}1$ scored in our auxiliary task of language recognition. The results were a test accuracy of 99,52%. Given our regular (non-normalized) confusion matrix for the task of language recognition:

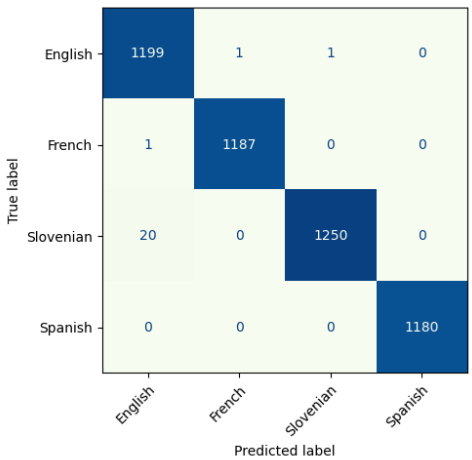

Figure 4.5. Confusion matrix (language) for $\alpha_1 = 0{,}9$, $\alpha_2 = 0{,}1$

We can see that our model only failed in classifying 23 out of 4839 utterances. The excellent performance hints that the auxiliary task is not challenging for the model. This can be observed further if we analyse the loss function of the auxiliary task throughout the training process, which converges very quickly to a minimal value. Figure 4.6 shows the value throughout the training epochs of both task losses and the general L loss, which is the weighted sum:

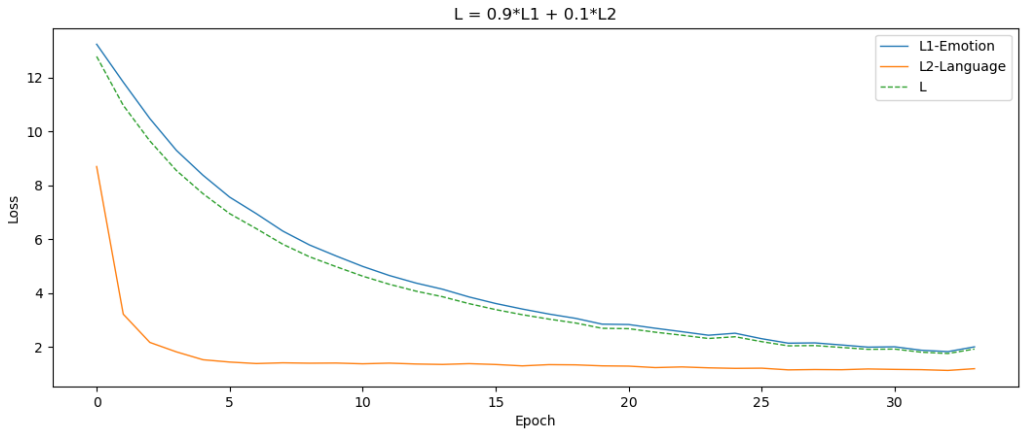

Figure 4.6. Value of the different multitask losses throughout training

As it shows, the loss for language recognition drops very quickly, roughly at the 3rd epoch to a converging value of 1,4. From the 3rd epoch on, there is no discernible decrease, meaning that it has reached the minimum value. The loss associated to emotion meanwhile presents a more usual behaviour of convergence, reaching the minimum at around the 30th epoch. This indicates that the model does not find language recognition as a difficult task to learn. What is happening is the opposite of the expected behaviour; the emotion recognition task is helping the model recognize language and not the other way around. It is worth noting that there is a chance that the model is not recognizing language but some other metric. This can happen if the task is not discriminative enough and contains some degree of overlap with other tasks. For example, considering that the different languages were recorded with two actors for each language, it could very well be that the model is recognizing the *speaker* instead of language. Additionally, the fact that the recording environment is consistent within each language partition could indicate that the model is classifying the *channel* instead. In practice though, for this dataset these tasks are equivalent and there is no concrete way to tell them apart, as they all fall within the same label.

These results, although interesting, show that for this specific model and dataset, language recognition as an auxiliary task does not help to improve the performance of the main task of emotion recognition. It is possible that other tasks could help, but it escapes the framework of this study.

5. Budget

To assess the budget of this project, I am going to consider this work as if I were hired as a junior engineer at SEAT, explaining the different costs and expenses accordingly.

Part of the expenses of this project are the work hours dedicated to development. This final thesis is equivalent to 18 ECTS, and according to European regulation 1 ECTS=30h. Considering this, I assume that the total number of hours is 18*30=540. Considering that the average hourly wage of a junior engineer is 14€/h, that sums up to:

$$C_1 = 540\ hours * 14€/h = 7560€$$

The second expense for this project is the computing power needed to train the models. There are several platforms that provide cloud computing services and the option to run on several GPUs for an hourly rate. A common one is Google Cloud Services, which has an hourly rate for the NVIDA A100 model of 2.47€/h. We only have the need to use 1 GPU. If we consider 35 experiments, including preliminary and final experiments, at an average of 1h per experiment we obtain a cost of:

$$C_2 = 35\ exp * 1h/exp * 2{,}47€/h = 86{,}45€$$

Giving us a final cost of the project of:

$$C = C_1 + C_2 = 7560€ + 86{,}45€ = 7646{,}45€$$

6. Conclusions

The research, development and experiments performed in this project have served the purpose of creating a powerful system that with Deep Learning is capable of recognizing a closed set of 7 emotions: 'Anger', 'Sadness', 'Joy', 'Fear', 'Disgust', 'Surprise' and a 'Neutral' emotional state, by automatically extracting the necessary features contained in the spectrogram of the speech signal.

The baseline model with statistical pooling has achieved the goal of improving the classification accuracy in respect to previous methods, specifically GMM, i-vectors with Cosine Distance and i-vectors with PLDA. Not only that, but the proposed model equipped with MHA pooling has delivered even better results, outperforming all other methods considered in this study by a considerable margin, including the statistical pooling approach and previous methods. This has also proven that attention mechanisms can be successfully implemented in order to pool the features in a more efficient way than by statistical methods, improving the classification performance of our model even further.

Self-attention and DMHA unfortunately have not resulted in a perceptible improvement over statistical pooling, but have served to obtain a more specific understanding about attention mechanisms and their implementations, as well as to gain insight into why the MHA mechanism performs better for SER.

The *INTERFACE* dataset serves its purpose and has been a great resource for the work done in this project, but in order to maximize the statistical consistency of the results, the model could have benefitted from training and testing with a larger amount of data. It is also possible that the self-attention mechanism could have presented better results if trained with longer audio lengths. In general, the differences between these attention mechanisms and statistical pooling could have been more evident if we had used a larger dataset. Complications arose when trying to access different data sources, so it was decided to proceed with *INTERFACE*.

Regarding the multi-task learning approach, it was unsure if it would provide an increase in performance, but I had the motivation to try it as it is an interesting challenge to tackle. Unfortunately, it did not achieve the desired results, but it did provide insight into this field and was a rewarding experience overall. If another dataset had been used different approaches could have been taken, maybe defining different auxiliary tasks, or even multiple secondary tasks. Further modifications to the output layers are also possible, there is much room for experimenting with multi-task learning.

All in all, it is safe to say that our DL approach to SER has exceeded expectations delivering remarkable results in the final comparative experiment, managing to accomplish the goals set for this project. On a more personal level, I am proud of the work done and consider it a success. I also consider that the knowledge obtained through this endeavour is valuable and very much in line with the state-of-the-art, considering the direction that Audio-visual Engineering is taking nowadays and the 'Deep Learning revolution' in the fields of speech and image processing. During the past few months developing this project, I have learned to structure my work, summarize information, compare results with different metrics and to explain these results in a comprehensive way. It has been tough at times, but overcoming the various challenges and complications that have appeared and arriving to a concluded project has been a rewarding final experience as a graduate student.

Future work

Experiments with feature fusion have not been performed in this project and could be an interesting approach to explore. TEO-based features, as well as prosodic features and voice quality features like HNR (Harmonics-to-Noise Ratio) and jitter can be studied and incorporated into the model in order to enhance the performance, as they convey critical information.

A promising approach could be to use what is known as Deep Speaker Conditioning [26]. In this study, an auxiliary network provides speaker embeddings, which conditions multiple layers of the primary classification network on a single neutral speech sample of the target speaker. This way it is possible to incorporate speaker information into the primary task of SER.

Other possible future developments include work regarding data augmentation, loss functions, other datasets and exploring different secondary tasks for MTL.

7. Bibliography

[1] R. Xia y Y. Liu, «Using i-Vector Space Model for Emotion Recognition,» *INTERSPEECH 2012,* pp. 2230-2233, 2012.

[2] K. Han, D. Yu y I. Tashev, «Speech emotion recognition using deep neural network and extreme learning machine,» *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH,* nº September, pp. 223-227, 2014.

[3] A. Nogueiras, A. Moreno, A. Bonafonte y J. B. Mariño, «Speech emotion recognition using hidden Markov models,» *EUROSPEECH 2001 - SCANDINAVIA - 7th European Conference on Speech Communication and Technology,* pp. 2679-2682, 2001.

[4] M. India, P. Safari y J. Hernando, «Self multi-head attention for speaker recognition,» *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH,* pp. 4305-4309, 2019.

[5] M. India, P. Safari y J. Hernando, «Double Multi-Head Attention for Speaker Verification,» 2021.

[6] R. Cowie y R. R. Cornelius, «Describing the emotional states that are expressed in speech,» *Speech Communication,* vol. 40, nº 1, pp. 5-32, 2003.

[7] S. Parthasarathy y C. Busso, «Jointly predicting arousal, valence and dominance with multi-Task learning,» *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH,* Vols. %1 de %22017-August, pp. 1103-1107, 2017.

[8] J. H. L. Hansen y S. E. Bou-Ghazale, «Getting started with SUSAS: a speech under simulated and actual stress database,» *EUROSPEECH-1997,* pp. 1743-1746, 1997.

[9] M. B. Akçay y K. Oğuz, «Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers,» *Speech Communication,* vol. 116, pp. 56-76, 2020.

[10] H. M. Teager y S. M. Teager, «Evidence for Nonlinear Sound Production Mechanisms in the Vocal Tract,» *Speech Production and Speech Modelling. NATO ASI Series (Series D: Behavioural and Social Sciences),* vol. 55, pp. 241-261, 1990.

[11] R. Banse y K. R. Scherer, «Acoustic Profiles in Vocal Emotion Expression,» *Journal of Personality and Social Psychology,* vol. 70, nº 3, pp. 614-636, 1996.

[12] H. Meng, T. Yan, F. Yuan y H. Wei, «Speech Emotion Recognition from 3D Log-Mel Spectrograms with Deep Learning Network,» *IEEE Access,* vol. 7, pp. 125868-125881, 2019.

[13] M. M. El Ayadi, M. S. Kamel y F. Karray, «Speech emotion recognition using Gaussian mixture vector autoregressive models,» *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings,* vol. 4, nº May, 2007.

[14] F. Perez, «Speech Emotion Recognition : Un Sistema de Reconocimiento de Emociones por Voz basado en Ivectors,» *ETSETB, UPC,* 2017.

[15] Y. Lecun, Y. Bengio y G. Hinton, «Deep learning,» *Nature,* vol. 521, nº 7553, pp. 436-444, 2015.

[16] J. Schmidhuber, «Deep Learning in neural networks: An overview,» *Neural Networks,* vol. 61, pp. 85-117, 2015.

[17] F. Wang, J. Cheng, W. Liu y H. Liu, «Additive Margin Softmax for Face Verification,» *IEEE Signal Processing Letters,* vol. 25, nº 7, pp. 926-930, 2018.

[18] D. Bahdanau, K. H. Cho y Y. Bengio, «Neural machine translation by jointly learning to align and translate,» *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings,* pp. 1-15, 2015.

[19] W. Cai, J. Chen y M. Li, «Exploring the Encoding Layer and Loss Function in End-to-End Speaker and Language Recognition System,» *Odyssey 2018,* pp. 74-81, 2018.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser y I. Polosukhin, «Attention is all you need,» *Advances in Neural Information Processing Systems,* Vols. %1 de %22017-Decem, pp. 5999-6009, 2017.

[21] J. Kim, G. Englebienne, K. P. Truong y V. Evers, «Towards speech emotion recognition "in the wild" using aggregated corpora and deep multi-task learning,» *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH,* Vols. %1 de %22017-August, pp. 1113-1117, 2017.

[22] J. Baxter, «A Bayesian/Information Theoretic Model of Learning to Learn via Multiple Task Sampling,» *Machine Learning,* vol. 28, nº 1, pp. 7-39, 1997.

[23] Y. Li, T. Zhao y T. Kawahara, «Improved end-to-end speech emotion recognition using self attention mechanism and multitask learning,» *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH,* Vols. %1 de %22019-Septe, pp. 2803-2807, 2019.

[24] V. Hozjan, Z. Kacic, A. Moreno, A. Bonafonte y A. Nogueiras, «Interface databases: Design and collection of a multilingual emotional speech database,» *Proceedings of the 3rd International Conference on Language Resources and Evaluation, LREC 2002,* pp. 2024-2028, 2002.

[25] K. Simonyan y A. Zisserman, «Very deep convolutional networks for large-scale image recognition,» *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings,* pp. 1-14, 2015.

[26] A. Triantafyllopoulos, S. Liu y B. W. Schuller, «Deep speaker conditioning for speech emotion recognition,» *2021 IEEE International Conference on Multimedia and Expo (ICME),* pp. 1-6, 2021.

Corpus	Access	Language	Size	Source	Emotions
Amir et al.	Private	Hebrew	60 Hebrew and 1 Russian actors	Nonprofessional actors	Anger, disgust, fear, joy, neutral, sadness
BabyEars	Private	English	509 utterances, 12 actors (6 males + 6 females), 3 emotions	Mothers and fathers	Approval, attention, prohibition
Beihang University	Private	Mandarin	7 actors x 5 emotions x 20 utterances	Nonprofessional actors	Anger, joy, sadness, disgust, surprise
Berlin emotional database	Public and free	German	800 utterances (10 actors x 7 emotions x 10 utterances + some second version) = 800 utterances	Professional actors	Anger, joy, sadness, fear, disgust, boredom, neutral
CLDC	Private	Chinese	1200 utterances, 4 actors	Nonprofessional actors	Joy, anger, surprise, fear, neutral, sadness
CMU-MOSEAS-WE1	Public and free	French, Spanish, Portuguese and German	1250 annotated videos for each language (640 GB)	Natural speech from videos	Happiness, sadness, anger, fear, disgust, surprise + sentiment (7 likert scale)
CMU-MOSEI	Public and free	English	65 hours of annotated video from more than 1000 speakers and 250 topics	Natural speech from videos	Happiness, sadness, anger, fear, disgust, surprise + sentiment (7 likert scale)
CREMA-D	Public and free	English	7442 clip of 12 sentences spoken by 91 actors (48 males and 43 females)	Professional actors	Angry, disgusted, fearful, happy, neutral, and sad
Danish emotional database	Public with license fee	Danish	4 actors x 5 emotions (2 words + 9 sentences + 2 passages)	Nonprofessional actors	Anger, joy, sadness, surprise, neutral
Emov-DB	Public and free	English	6893 recordings by 4 actors	Professional actors	Neutral, sleepiness, anger, disgust and amused
ESMBS	Private	Mandarin	720 utterances, 12 speakers, 6 emotions	Nonprofessional actors	Anger, joy, sadness, disgust, fear, surprise
FERMUS III	Public with license fee	German, English	2829 utterances, 7 emotions, 13 actors	Automotive environments	Anger, disgust, joy, neutral, sadness, surprise

Corpus	Access	Language	Size	Source	Emotions
Hao Hu et al.	Private	Chinese	8 actors x 5 emotions x 40 utterances	Nonprofessional actors	Anger, fear, joy, sadness, neutral
INTERFACE	Commercially available	English, Slovenian, Spanish, French	24197 recordings	Actors	Anger, disgust, fear, joy, surprise, sadness, slow neutral, fast neutral
JL-Corpus	Public and free	English	2400 recording of 240 sentences by 4 actors (2 males and 2 females)	Professional actors	Primary emotions: angry, sad, neutral, happy, excited + Secondary emotions: anxious, apologetic, pensive, worried, enthusiastic
KES	Private	Korean	5400 utt., 10 actors	Nonprofessional	Neutral, joy, sadness, anger
KISMET	Private	American English	1002 utterances, 3 female speakers, 5 emotions	Nonprofessional actors	Approval, attention, prohibition, soothing, neutral
LDC Emotional Prosody Speech and Transcripts	Commercially available	English	7 actors x 15 emotions x 10 utterances	Professional actors	Neutral, panic, anxiety, hot anger, cold anger, despair, sadness, elation, joy, interest, boredom, shame, pride, contempt
LEGO Corpus	Public and free	English	347 dialogs with 9,083 system-user exchanges	System-user exchanges	Garbage, non-angry, slightly angry and very angry
MELD	Public and free	English	1400 dialogues and 14000 utterances from multiple speakers	Friends TV show	Anger, disgust, sadness, joy, neutral, surprise and fear + sentiment (positive, negative, neutral)
MPEG-4	Private	English	2440 utterances, 35 speakers	US American movies	Joy, anger, disgust, fear, sadness, surprise, neutral
MSP-Podcast Corpus	Public and free	English	100 hours by over 100 speakers	Podcast speakers	Activation, dominance and valence (7 likert scale) + Primary emotion (anger, sadness, happiness, surprise, fear, disgust, contempt and neutral state) + secondary emotion (amused, frustrated, depressed, concerned, disappointed, excited, confused, and annoyed)

Corpus	*Access*	*Language*	*Size*	*Source*	*Emotions*
Natural	Private	Mandarin	388 utterances, 11 speakers, 2 emotions	Call centers	Anger, neutral
Pereira	Private	English	2 actors x 5 emotions x 8 utterances	Nonprofessional actors	Hot anger, cold anger, joy, neutral, sadness
RAVDESS	Public and free	English	7356 recordings by 24 actors.	Professional actors	Calm, happy, sad, angry, fearful, surprise, and disgust
SUSAS	Public with license fee	English	16,000 utterances, 32 actors (13 females + 19 males)	Speech under simulated and actual stress	Four stress styles: Simulated Stress, Calibrated Workload Tracking Task, Acquisition and Compensatory Tracking Task, Amusement Park Roller-Coaster, Helicopter Cockpit Recordings
TESS	Public and free	English	2800 recordings by 2 actresses	Professional actors	Anger, disgust, fear, happiness, pleasant surprise, sadness, and neutral

Glossary

AMSoftmax: Angular Margin SoftMax

CNN: Convolutional Neural Network

DL: Deep Learning

DMHA: Double MulitHead Attention

ECTS: European Credit Transfer and Accumulation System

EER: Equal Error Rate

FC: Fully Connected

GMM: Gaussian Mixture Models

GPU: Graphical Processing Unit

HMM: Hidden Markov Models

HNR: Harmonics-to-Noise Ratio

IDE: Integrated Development Environment

IST: Information Society Technologies

LSTM: Long Short-Term Memory

MFCC: Mel-Frequency Cepstral Coefficients

MHA: MultiHead Attention

ML: Machine Learning

MTL. Multi-Task Learning

NMT: Neural Machine Translation

PLDA: Probabilistic Linear Discriminant Analysis

RE: Relative Error

ReLu: Rectified Linear Unit

RNN: Recurrent Neural Networks

seq2seq: sequence-to-sequence

SER: Speech Emotion Recognition

SUSAS: Speech Under Simulated and Actual Stress

TALP: Centre de Tecnologies i Aplicacions del Llenguatge i la Parla

TEO: Teager-Energy Operator

VAD: Arousal-Valence-Dominance

vgg16: Visual Geometry Group 16

VGG3L: Visual Geometry Group 3 Layers

WAV: Waveform audio format