# Contribution to the realization of an Emotion Recognition System : multilingual case

Anwer Slimi

THÈSE EN COTUTELLE PRÉSENTÉE

POUR OBTENIR LE GRADE DE

# DOCTEUR DE

# L'UNIVERSITÉ DE BORDEAUX

# ET DE LA FACULTÉ DES SCIENCES DE MONASTIR

ÉCOLE DOCTORALE UBX

ÉCOLE DOCTORALE DES SCIENCES ET TECHNOLOGIES DE L'INFORMATION (EDSTI)

SPÉCIALITÉ Informatique

Par Anwer SLIMI

# Contribution to the realization of an Emotion Recognition System: multilingual case

Sous la direction de Mounir ZRIGUI
et de Henri NICOLAS

Soutenue le 18/08/2022

Membres du jury :

| | | | | |
|---|---|---|---|---|
| Mme. | BENOIS-PINEAU | Jenny | Professeur (Univ. Bordeaux) | Présidente |
| M. | MAHJOUB | Mohamed Ali | Professeur (Univ. Sousse) | Rapporteur |
| M. | HADDAR | Kais | Professeur (Univ. Sfax) | Rapporteur |
| M. | BIMBOT | Frédéric | Directeur de recherche (CNRS, Rennes) | Examinateur |
| M. | NICOLAS | Henri | Professeur (Univ. Bordeaux) | Directeur de thèse |
| M. | ZRIGUI | Mounir | Professeur (Univ. Monastir) | Directeur de thèse |

## Titre : Contribution à la réalisation d'un système de reconnaissance des émotions : cas multilingue

**Résumé:** L'émotion est un phénomène complexe qui apporte une contribution significative à la communication humaine. Les émotions donnent du sens à la conversation entre les individus et nous permettent de mieux nous comprendre. En outre, l'interaction homme-machine a produit des changements importants ces dernières années pour satisfaire les exigences et les responsabilités des clients. La communication entre les individus et les machines est devenue possible grâce aux multiples avancées technologiques. Sous cet angle, il serait parfait que les machines reconnaissent automatiquement les émotions humaines afin d'améliorer la communication et l'interaction entre les deux parties. Dans cette thèse, nous proposons un certain nombre d'innovations pour améliorer l'efficacité des systèmes actuels. Nous proposons également le premier modèle de détection du changement de catégories d'émotions dans les discours.

**Mots clés:** Reconnaissance des émotions, Analyse de la parole, Apprentissage en profondeur

---

## Title: Contribution to the realization of an Emotion Recognition System: multilingual case

**Abstract:** Emotion is a complicated phenomenon that makes a significant contribution to human communication. Emotions add meaning to the conversation between individuals and allow us to better understand each other. Besides, Human-Computer interaction has produced significant changes in recent years to satisfy the requirements and responsibilities of clients. Communication between individuals and machines has become feasible through multiple advances in technology. From this angle, it would be perfect for machines to automatically recognize human feelings in order to improve communication and interaction between both parts. In this thesis, we propose a number of innovations to enhance the efficiency of current systems. We propose as well the first model for the detection of emotion categories' change in Speeches.

**Keywords:** Emotion Recognition, Speech analysis, Deep Learning

---

# Unité de recherche

LaBRI (Laboratoire Bordelais de Recherche en Informatique) UMR 5800 - 351 Cours de la Libération, 33405 Talence

RLANTIS (Research Laboratory in Algebra, Numbers theory and Intelligent Systems), LR 18ES15, Monastir, Tunisia

# Aknowledgements

*As a preamble to this thesis, I would like to express my gratitude to the people who helped me and who contributed to the development of this work.*

*First of all, I thank my supervisors Pr. Mounir ZRIGUI, professor at the Faculty of Sciences of Monastir (Tunisia) and Pr. Henri NICOLAS, professor at the University of Bordeaux (France), for all their contributions, their encouragement and their quite useful and fruitful advises that helped me to the realization of this work.*

*I owe a dept of gratitude to to my dear friend Marwa THABET for her trust, encouragements, and her endless support during this thesis.*

*I want also to convey my appreciation to my family and my friends, whom I consider family, for sticking by my side and believing in me. They made me realize that the family is sacred. They were a real source of inspiration for me and were always by my side during difficult times.*

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

**BiLSTM** Bidirectional Long Short-term Memory

**CNN** Convolutional Neural Network

**CTC** Connectionist Temporal Classification

**eGeMAPS** The Geneva Minimalistic Acoustic Parameter Set

**LSTM** Long Short-term Memory

**MFCC** Mel Frequency Cepstral Coefficient

**RNN** Recurrent Neural Network

**SER** Speech Emotion Recognition

**t-SNE** t-Distributed Stochastic Neighbor Embedding

**ViT** Vision Transformer

# List of publications

- Anwer Slimi, Mohamed Hamroun, Mounir Zrigui and Henri Nicolas: **Emotion Recognition from Speech using Spectrograms and Shallow Neural Networks**. The 18th International Conference on Advances in Mobile Computing Multimedia **(MoMM2020)**.

- Anwer Slimi, Henri Nicolas and Mounir Zrigui: **Detection of Emotion Categories' Change in Speeches.** The 14th International Conference on Agents and Artificial intelligence **(ICAART 2022).**

- Anwer Slimi, Mounir Zrigui and Henri Nicolas: **MuLER: Multiplet Loss for Emotion Recognition.** ACM International Conference on Multimedia Retrieval **(ICMR 2022)**

- Anwer Slimi, Nafaa Haffar, Mounir Zrigui and Henri Nicolas: **Multiple Models Fusion for Multi-label Classification in Speech Emotion Recognition Systems.** The 26th International Conference on Knowledge-Based and Intelligent Information Engineering Systems **(KES 2022).**

- Anwer Slimi, Henri Nicolas and Mounir Zrigui: **Hybrid Time Distributed CNN-Transformer for Speech Emotion Recognition.** The 17th International Conference on Software Technologies **(ICSOFT 2022).**

# Résumé substantiel

L'une des choses qui unit les humains est la communication, qui peut être définie comme le cœur de notre existence quotidienne. Ce processus peut être établi de plusieurs façons telles que la communication verbale (parole), la communication écrite (lettres), ou même avec la langue des signes. Les émotions sont parmi les informations les plus importantes véhiculées dans la parole puisqu'elles jouent un rôle fondamental dans tous les phénomènes sociaux.

Avec l'essor des interactions homme-machine, il est devenu nécessaire pour les machines de mieux comprendre les humains afin de réagir de manière appropriée. Par conséquent, afin d'augmenter la communication et l'interaction, il serait idéal que les machines détectent automatiquement les émotions humaines. La reconnaissance des émotions de la parole a fait l'objet de nombreuses études ces dernières années. Cependant, les modèles peuvent être considérés comme peu précis et doivent être améliorés. En règle générale, le pipeline du système de reconnaissance des émotions de la parole consiste à prétraiter un signal audio, à extraire un ensemble de caractéristiques et enfin à alimenter l'ensemble de caractéristiques à un algorithme de classification pour déterminer l'émotion exprimée dans ce signal. Dans le domaine de la reconnaissance des émotions de la parole, nous sommes confrontés à certaines limites et défis. Le premier concerne la partie extraction de caractéristiques puisqu'elle joue un rôle majeur dans le succès de chaque modèle de machine learning. Ces dernières années, de nombreux algorithmes d'extraction de caractéristiques ont été utilisés à des fins de classification des émotions. Ainsi, le meilleur ensemble de fonctionnalités reste un défi majeur. La deuxième limitation est représentée dans l'algorithme de classification puisque la principale raison de la précision la plus faible ou la plus élevée est le choix du modèle et la configuration du modèle lui-même. Avec

l'essor et le succès de l'apprentissage en profondeur dans différents domaines, plusieurs recherches ont tendance à utiliser les réseaux de neurones profonds. Ce qui en fait un défi, c'est le manque des données, car la plupart des bases de données actuelles sont petites. Ceci est traité comme un problème car les réseaux de neurones sont la plupart du temps plus efficaces que les algorithmes d'apprentissage automatique traditionnels, mais ils nécessitent une énorme quantité de données pour l'apprentissage afin d'atteindre des précisions plus élevées. La troisième limite concerne les êtres humains qui ont parfois des difficultés à reconnaître les émotions et à titre d'exemple les fichiers audio du jeu de données RAVDESS ont été validés par 247 évaluateurs et la précision était d'environ 60% , ce qui pose un défi majeur quant à comment concevoir un modèle qui est plus précis que les gens eux-mêmes.

A travers cette thèse, nous avons pu présenter plusieurs contributions significatives et de nouvelles perspectives :

- Les Transformers ont pris d'assaut le monde de la TALN (Traitement Automatique de Langage Naturel) ces années récentes. Ils sont basés sur une architecture qui utilise le principe d'Attention pour améliorer considérablement les performances des modèles de traduction automatique.Cependant, bien que les Transformers aient été conçus à l'origine pour fonctionner avec des données textuelles, cela ne les a pas empêchés d'être utilisés dans une variété de tâches de traitement de la parole et de vision par ordinateur, y compris le traitement d'images, avec des résultats comparables au réseau de neurones convolutif. Malgré leur efficacité, les Transformers fonctionnent mal sur des petits ensembles de données, ce qui est préoccupant compte tenu du fait que la majorité des ensembles de données dans le domaine de la reconnaissance des émotions de la parole sont assez petits. De plus, de nombreuses études ont démontré que les Transformers en général sont moins puissants lorsqu'ils traitent des données de vision localement invariantes, et que les ViT en particulier sont vulnérables aux correctifs contradictoires. Pour éviter ces soucis, nous proposons de combiner le transformateur standard avec un réseau de neurones convolutif. Nous avons en fait utilisé un réseau de neurones convolutif distribué dans le temps (Time Distributed Convolutinal Neural Network) au lieu d'un réseau de neurones convolutif

traditionnel, ce qui nous permet d'appliquer une couche à chaque tranche temporelle d'une entrée. Cette nouvelle architecture a permis d'améliorer la précision d'une base de données sur deux.

- L'échec du modèle proposé sur la deuxième base de données nous a incités à déplacer notre attention de la classification de la parole vers l'encodage de la parole. Nous avons suggéré un modèle pour le codage de la parole, qui prend un fichier audio en entrée et génère un vecteur avec 128 valeurs. Le concept derrière cette contribution est d'encoder les signaux audio de telle manière qu'ils exposent les émotions plutôt que de les classer, c'est-à-dire que les fichiers audio avec la même étiquette auront des encodages comparables (la similarité intraclasse est maximale et la similarité interclasse est minimale). Pour acquérir le codage parfait, nous avons besoin d'une fonction de perte (Loss function). Pour commencer, nous avons utilisé la fonction Triplet-loss, qui a considérablement amélioré les résultats par rapport aux études précédentes, puis nous avons proposé notre propre fonction, une extension de la Triplet-loss que nous avons nommée Multiplet-loss. Notre modèle prend de nombreux fichiers simultanés, encode chacun d'eux, puis utilise la nouvelle fonction de perte pour diminuer la distance entre l'ancre et les discours avec la même étiquette tout en maximisant la distance entre l'ancre et les autres discours. En termes de précision et de temps d'exécution, la Multiplet-loss a dépassé la Triplet-loss.

- Avec le nouveau modèle, nous apportons une nouvelle perspective dans le domaine de la reconnaissance des émotions de la parole. Donner un discours à une personne induirait la reconnaissance d'une émotion. Donner le même discours à plusieurs personnes les amènera à reconnaître une ou plusieurs émotions. De ce point de vue, on peut affirmer qu'une machine qui ne peut identifier qu'une seule émotion à la fois est une perception du cerveau d'un seul individu. Un ordinateur intelligent est un ordinateur qui surpasse la grande majorité, sinon la totalité, des humains et est capable de réagir et d'envisager des scénarios alternatifs basés sur des possibilités alternatives. En conséquence, nous avons envisagé de développer un classificateur multi-étiquettes. Cette nouvelle vision permet à une machine d'évaluer toutes les catégories d'émotions possibles que les humains peuvent identifier. Un tel modèle

nécessite un ensemble de données dédié dans lequel chaque fichier doit inclure une ou plusieurs étiquettes simultanément. A notre connaissance, toutes les bases de données existantes sont cataloguées et annotées à l'aide d'une seule étiquette. En conséquence, nous avons étiqueté à la main l'ensemble de données RML avec l'aide de 50 volontaires.

- Les émotions sont dynamiques par nature, et elles changent avec le temps. Ainsi, un système intelligent devrait être capable de détecter les changements d'état émotionnel lorsqu'ils se produisent lors d'une interaction homme-ordinateur dans laquelle les émotions des locuteurs sont déterminées par des indices comportementaux. De cette façon, il peut réagir de manière appropriée. Les recherches existantes se sont principalement concentrées sur la détection de l'instant du changement d'émotion, c'est-à-dire la détermination précise du moment où un changement d'émotion s'est produit, ou sur la prévision du changement de valence (positive ou négative) et d'excitation (faible ou élevée). À notre connaissance, il s'agit de la première étude à introduire une méthode pour détecter les changements des catégories d'émotions, c'est-à-dire déterminer si un changement s'est produit d'une catégorie (en colère, triste, neutre, etc.) à une autre. En d'autres termes, si une personne parlait avec une émotion particulière puis changeait brusquement d'attitude, le système serait capable de détecter ce changement. Dans cette contribution, nous avons présenté le premier système de ce type qui détecte les changements dans les catégories d'émotions. Le modèle a été entraîné à l'aide de la fonction CTC loss (Connectionist Temporal Classification) et était basé sur l'architecture CNN-LSTM.

# General introduction

One of the things that binds human people to each other is communication, which may be defined as the core of our daily existence. This process can be established in several ways such as verbal communication (Speeches), written communication (Letters), or even with sign language. Emotions are among the most important information conveyed in speeches since play a basic role in all social phenomena [1].

With the rise of human-machine interactions, it has become necessary for machines to better understand humans in order to respond appropriately. Hence, in order to increase communication and interaction, it would be ideal for machines to automatically detect human emotions. Speech Emotion Recognition (SER) has been a focus of a lot of studies in the past few years. However, they can be considered poor in accuracy and must be improved.

Usually, the SER system's pipeline consists of pre-processing an audio signal, extracting a set of features, and finally feeding the set of features to a classification algorithm to determine the emotion expressed in that signal. In the SER domain, we are facing some limitations and challenges. The first one concerns the feature extraction part since it plays a major role in every machine learning model's success. In recent years, a lot of feature extraction algorithms have been used for the purpose of emotion classification. So, the best feature set remains a major challenge. The second limitation is represented in the classification algorithm since the key reason for the lowest or the highest accuracy, is the choice of the model and the model's configuration itself. With the rise and the success of the deep learning in different domains, several researches tend to use deep neural networks. What makes this a challenge is the lack of data as most current databases are small. This is dealt with as a problem, since neural networks are most of the time more efficient in

comparison to traditional machine learning algorithms, but they require a huge amount of data for training to achieve higher accuracies. The third limitation is concerned with human beings who sometimes struggle to recognize emotions and as an example, the audio files in the RAVDESS dataset were validated by 247 raters and the accuracy was around 60%, which poses a major challenge as to how could we design a more precise and accurate model than people themselves.

In this thesis, **four** contributions are proposed:

- Several deep learning learning models have shown remarkable effectiveness in a variety of disciplines. The most recent successful model is the Transformer, which has been employed in almost every area since 2019. They do, however, have certain constraints. To address these problems, we propose a new hybrid classification model based on a Transformer preceded by a Time Distributed Convolutional Neural Network, which enables us to apply a layer to each time slice of an input. This new architecture has improved the accuracy of one out of two databases (**Chapter 3**).

- The under-performance of the proposed model on the second dataset prompted us to shift our focus from speech classification to speech encoding. We propose a model for speech encoding, which takes an audio file as input and generates a vector with 128 values. The concept behind this contribution is to encode audio signals in such a way that they exhibit emotions rather than classifying them, i.e. audio files with the same tag will have comparable encodings (the intra-class similarity is maximal and the inter-class similarity is minimal). To acquire the perfect coding, we propose a new loss function that we call Multiplet-loss (**Chapter 4**).

- With the new proposed models, we bring a new vision into the field of speech emotion recognition. We propose the concept of multi-label classification. This new insight allows a machine to assess all possible categories of emotions that humans can identify. Such a model requires a dedicated data set in which each file must include one or more labels simultaneously. To our knowledge, all existing databases are cataloged and annotated using a single tag. Accordingly, we hand-labeled the RML dataset with the help of 50 volunteers (**Chapter 3**).

- Several research studies have focused on speech emotion recognition; however, only a few papers have addressed the Emotion Change during Conversation, which may be useful in a variety of applications. While previous studies have mostly focused on recognizing the instant of emotion change or detecting the valence and arousal, this contribution is the first to present work on emotion category change detection (**Chapter 5**).

# Chapter 1

# Emotions and Emotion Recognition from Speech

This chapter sheds light on the significance of emotions in our everyday lives, as well as the significance of Speech Emotion Recognition (SER) systems. We also outline the aims of this thesis and show the tools that we utilized throughout this thesis in detail.

## 1.1 The definition of emotions

Humans perceive emotions as reactions to events or situations. The event that causes the feeling determines the sort of emotion experienced. The evolutionary theory of emotion, introduced by Charles Darwin, contends that emotions are adaptable to our environment and increase our chances of survival. Emotions like love, for example, are adaptive because they encourage mating and reproduction. Fear, for example, protects us against predators [2].

An emotion is a multifaceted psychological state with three unique components:

- Subjective experience: While researchers concur that, there are certain fundamental common emotions that individuals all over the world experience regardless of background or culture, studies also believe that emotion may be very subjective. Anger, for example, can manifest itself in a variety of ways, ranging from slight annoyance to blazing anger. We also do not always have pure manifestations of each feeling. We all experience mixed emotions in response to various events or situations in our

life.

- Physiological response: Anyone can feel his or her stomach lurch from anxiety or his or her heart palpitate from fear, which indicates that emotions can produce strong physiological reactions. Numerous physiological responses that a person encounters when experiencing an emotion, such as sweating hands or a racing heart, are under the direction of the sympathetic nervous system.

- Behavioral or expressive response: we devote a lot of our effort to deciphering the emotional expressions of others around us. These expressions play a significant role in our entire body language and are closely related to what psychologists refer to as emotional intelligence, which is the capacity for proper interpretation of these expressions. According to research, many facial emotions are universal, including a smile to denote happiness and a frown to denote unhappiness.

People frequently use the words emotions, feelings, and moods interchangeably in common discourse, although these terms truly represent distinct things. An emotion is typically brief but intense. Emotions are also likely to have a specific and recognizable reason. They are responses to stimuli, while feelings are what we experience as a result of emotions. Feelings are impacted by our perspective of the event, which is why the same emotion may elicit various reactions in different people [3]. A mood, on the other hand, might be thought of as a temporary emotional state. A person may believe that everything is going well this week, which would explain why he is in a good mood, but moods can also be triggered by obvious factors. It can, however, often be challenging to pinpoint the precise reason for a mood. For instance, he might discover himself experiencing prolonged, unjustified melancholy.

There is no established "standard" for categorizing different emotional states as of yet. Six fundamental emotions—fear, disgust, anger, surprise, happiness, and sadness—were said to be shared by all human societies by psychologist Paul Ekman in 1972 [4]. Robert Plutchik later developed the "wheel of emotions," another technique for categorizing emotions, in the later 1980s. This model illustrated how many emotions can be joined or blended, in a manner similar to how an artist would combine primary colors to produce other colors [5]. Then, in 1999, Ekman added a number of more fundamental emotions to

his list, including excitement, contempt, shame, pride, satisfaction, and amusement [4].

## 1.2 Importance of emotions

Emotions may have a crucial impact in the way we think and react. The emotions that we experience each day may force us to take action and impact the choices we make about our life, both great and little. They have a tremendous effect on the choices we make, from what we decide to eat for breakfast to the politicians we choose to vote for in political elections. Understanding the emotions of others offers us clear knowledge about how we may need to behave in a certain scenario. Emotions may assist a decision-maker understand which components of a choice are the most important to their individual scenario. They may also help individuals make quicker judgments [6].

Researchers have also observed that individuals with specific forms of brain injury limiting their capacity to feel emotions also have a lower ability to make smart judgments [7]. Even in instances when you feel your judgments are dictated completely by logic and reason, emotions play a crucial influence. By alerting individuals around us that we are feeling pleased, sad, enthusiastic, or terrified, we are providing them critical information that they may then utilize to take action. Just as our own emotions convey vital information to others, the emotions of people around us also provide a lot of social information.

## 1.3 The need for SER system

Emotions have a crucial part in human interpersonal relationships on a daily basis. This is critical for both reasonable and intelligent decision-making. It enables us to relate to and comprehend the sentiments of others by communicating our own and providing feedback to others. Emotions have a significant influence in moulding human social interaction, as research has shown. Emotional expressions offer a great deal of information about an individual's mental state. This has spawned a new branch of study known as automatic emotion recognition, with the primary objective of comprehending and retrieving desired emotions.

Prior research has examined a variety of modalities for recognizing emotional states, including facial expressions, speech, and physiological signals. Due to a number of in-

trinsic benefits, voice signals provide an excellent source for effective computing. For instance, as compared to a variety of other biological signals (e.g., ECG), speech signals are often more easily and affordably collected. This is why the overwhelming majority of researchers are intrigued by Speech Emotion Recognition (SER). SER seeks to infer a speaker's underlying emotional state from their voice. The area has seen an increase in scientific attention during the last several years. There are several applications for detecting human emotion, including human-robot interaction, audio surveillance, commercial applications, clinical investigations, entertainment, banks, call centers, cardboard systems, and computer games. For orchestration of classrooms or E-learning, information on students' emotional states might help focus efforts on improving teaching quality. For instance, a teacher may utilize SER to choose which topics to teach and must be able to build ways for controlling emotions in the classroom. That is why the emotional condition of the student should be taken into account in the classroom. With the proliferation of chat-bots and automated response robots in call centers, speech emotion detection may be useful. Having a robot flood you with questions may be infuriating. As a result, such a system may enhance customer service. The addition of emotions to robots has been regarded as a vital aspect in achieving a human-like appearance and behavior in machines [8]. Emotion detection is very important and it can be useful in several applications:

- Emotion recognition to monitor the psychological state of a patient at home
- Emotion recognition for a video/music recommendation engine
- Emotion Recognition for a Custom Voicebot System
- Emotion recognition to analyze customer satisfaction in call centers
- Emotion recognition to adapt video game scenarios
- Emotion recognition for e-learning platforms (adapt the task to become less or more difficult)
- Emotion recognition for smarthome: Some studies have shown that brighter light can intensify emotions, while dim light does not suppress emotions, but keeps them stable. It can lead people to have the ability to make more rational decisions in low light conditions

- Emotion recognition for fraud detection (insurance, banking, etc.)

## 1.4 Architecture of SER

The goal of SER systems is to recognize emotions from a given utterance. The architecture of the SER systems consists of three main steps:

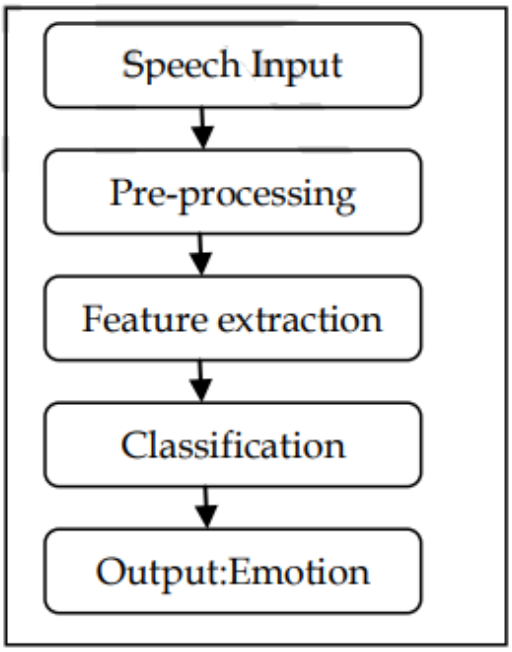


Figure 1.1: Steps of SER system with feature extraction [9]

### 1.4.1 Pre-processing

Pre-processing audio is a two-stage technique that ensures audio assets from one session to the next match before they are used in a project. The first stage is pre-editing and converting raw audio to a common format. This often entails the elimination of undesired parts, such as conversation between takes, coughs, and sneezes, as well as any outlier peaks, such as clicks, thumps, and paper rustling, to leave a clean audio file. The second stage is to eliminate undesired noise, rumble, and hum, as well as tonal and overall leveling, via the use of noise reduction, harmonic enhancement, and dynamics, among other techniques, to create a collection of clean, leveled audio assets. Pre-processing is necessary only when working with noisy datasets, i.e., utterances that contain background noise. We omit this

step since this thesis makes use of pre-cleaned datasets recorded in studios equipped with sophisticated equipment.

### 1.4.2 Feature extraction

Feature extraction starts with a collection of measured data and provides meaningful and non-redundant derived values (features), hence facilitating later learning and generalization stages and, in some circumstances, resulting in more accurate human interpretations. Dimensionality reduction is tied to the extraction of features. The purpose of feature extraction is to reduce the number of resources required to explain a large quantity of data. When evaluating complex data, the presence of several variables is one of the greatest obstacles. A large number of variables often necessitates a substantial amount of memory and computational resources, and may cause a classification algorithm to become over-fit to training data and under-perform on new samples. Feature extraction is an all-encompassing term that refers to approaches for combining variables to sidestep these problems while properly describing the data. The three basic kinds of features utilized for emotion identification are prosodic features, excitation source characteristics, and spectral features:[9]:

- Prosodic features: basically, include sound Intensity, Pitch, Energy and many other features.
- Excitation source features: they are obtained from excitation source signal which we get by suppressing vocal tract characteristics.
- Spectral features: also known as vocal tract, system features or even segmental features. These features, include Mel-Frequency Cepstral Coefficients (MFCC), linear prediction cepstral coefficients (LPCC) and perceptual linear prediction coefficients (PLPC).

In addition to the approaches stated above, there were several more feature extraction techniques used, including Gray-Level Co-Occurrence Matrix [10], The Geneva Minimalistic Acoustic Parameter Set [11], Mel-scaled spectrogram, Chromagram, Spectral contrast feature and Tonnetz representation [12] and a variety of others. As a result, determining the optimal feature set remains a significant task.

### 1.4.3 Classification

For the purposes of machine learning, the word "classification" refers to the process of using predictive modeling to assign a class label to a sample of input data. Classification requires a training dataset containing a high number of examples of inputs and outputs for learning purposes. A model will identify the optimum mapping of given input instances to given class labels using the training dataset. As a result, the training set must include a significant number of examples of each class label and be adequately representative of the circumstance. Class labels are often textual values, such as "Angry," "Neutral," or "Happy," which must be transformed to numeric values before to sending to a modeling process. In label encoding, each class label is assigned a unique number, such as "Angry" = 0, "Neutral" = 1, "Happy" = 2, etc. Numerous classification methods exist for designing classification predictive modeling problems, but there is currently no good theory for mapping algorithms to problem types; researchers are generally advised to conduct controlled experiments to determine which algorithm and algorithm configuration performs best for a given classification problem. Classification and predictive modeling systems are evaluated based on the results they provide. Classification accuracy is a commonly used metric for assessing the performance of a model based on predicted class labels.

## 1.5 Objectives of this thesis

Recently, a number of publications have been published in which new architectures and novel approaches for improving the accuracy of Speech Emotion Recognition systems have been introduced and discussed. There are articles that suggest novel feature extraction approaches, papers that propose new pre-processing techniques, and other papers that propose new combinations of many classifiers. Despite this, the accuracy is still poor and requires more work [13]. In this thesis, we focused on enhancing the accuracy of SER systems by introducing a novel model that introduces a fresh vision. In addition, we introduce a new technique for detecting changes in emotion during conversations, with the goal of improving human-computer interaction.

## 1.6 Tools used throughout this thesis

To be able able to design a robust emotion recognition system, we first need to study the impact of different algorithms on the accuracy.

### 1.6.1 Used Feature sets

Numerous strategies have been examined in previous articles. Several of these articles suggested combining several feature sets simultaneously [9][14]. This is an important stage since it has a significant impact on the system's accuracy, thus attention should be taken in selecting which features to employ. When it comes to voice processing challenges, both the spectrogram a nd the MFCC approach are the most often utilized techniques in the Natural Language Processing sector (speaker identification, speech recognition, gender recognition, etc.).

#### Spectrograms

A spectrogram is a visual depiction of a signal's frequency spectrum as it evolves over time; in other words, spectrograms are one potential representation of an audio file. A sound is a vibration that normally propagates via a transmission medium as an audible wave of pressure. Thus, it is a sinusoid of air pressure. To capture the sound frequencies, human beings use their cochlea that detects air pressure sinusoids where their frequencies lay in-between 20Hz and 20KHz. As for computers, the air pressure frequencies are captured via an audio recorder, say a microphone, then, it is transformed to a digital signal which can be understood and manipulated with a CPU (Central Processing Unit). To get more sense on our audio signal, we can visualize it graphically by plotting it on the screen. If we take a quick look at Figure 1.2, we can see that there are three sound bursts with a break between them. Although the three chunks look quite similar and appear to have the same sound, they do not actually match. The only thing we can extract is when each of them would start and when it would end. No other information can be obtained just by looking at the time domain plot. Since we have already mentioned earlier that a sound is basically a bunch of frequencies, we can try a sort of plot, in which frequencies could be shown. For this purpose, we will apply a DFT on our audio signal.

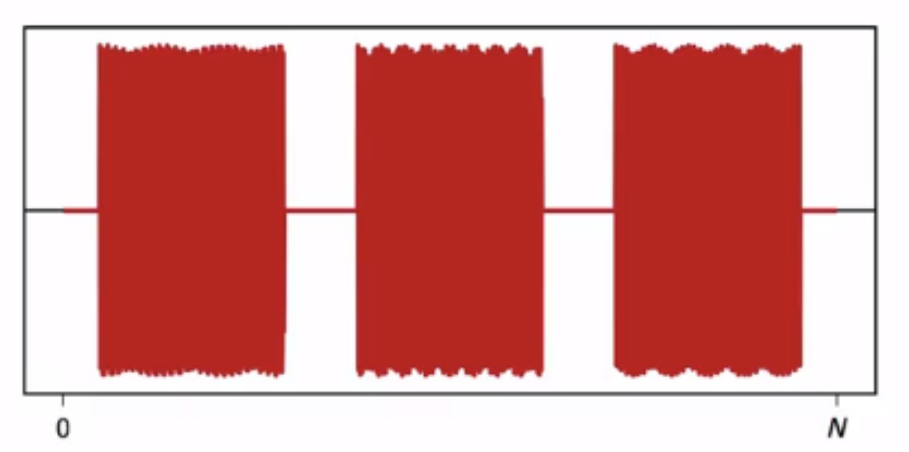


Figure 1.2: Plot of an audio signal in the time domain [15]

A DFT is a mathematical operation applied on a signal so that we can plot our signal in frequency-domain instead of time-domain. So, it is simply a change of basis.

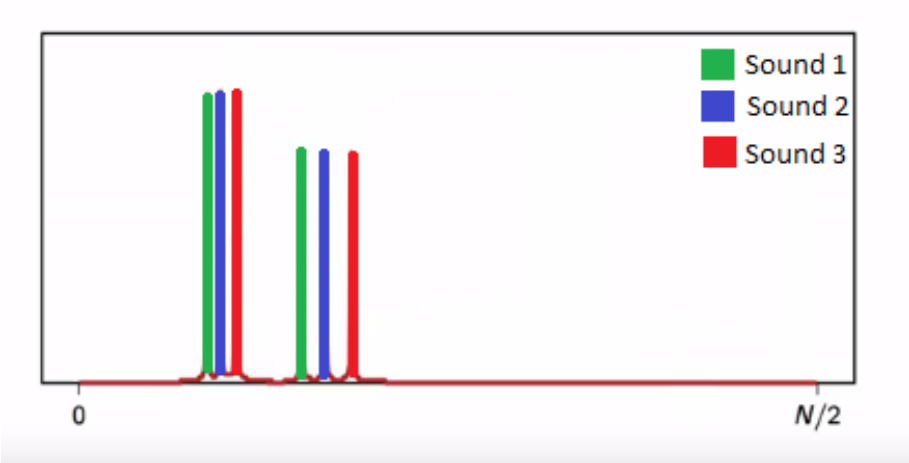


Figure 1.3: Plot of an audio signal in the frequency domain [15]

The Figure 1.3 gives a new whole perspective on our signal. Now we are able to visualize the pair of frequencies with which each chunk is composed. However, we cannot really distinguish in which order these three sounds are aligned. It is crystal clear that the frequency content is totally obscured by the time representation. So, we understand the timing but the content is unknown to us. Also, the time data is blurred by the frequency representation. So, we can understand the frequencies but we do not know when they would happen. So, we need a sort of representation that includes both, time and frequency information. And this is the idea behind the STFT . The STFT is actually taking little pieces of length L from the original signal and apply a DFT on each piece

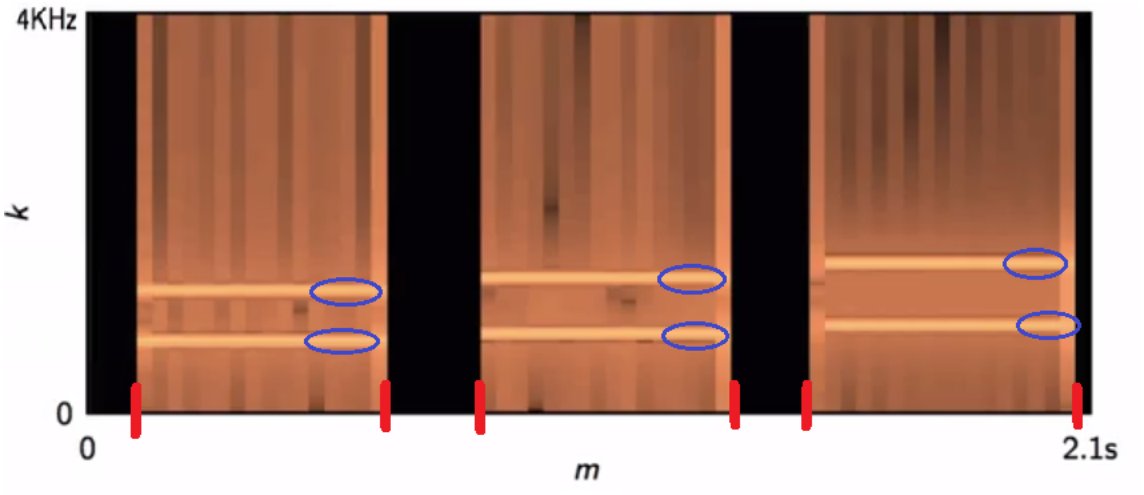


Figure 1.4: Example of a spectrogram [15]

instead of applying a DFT on the whole signal. The equation that allows us to do so is the following:

$$X[m;k] = \sum_{n=0}^{L-1} x[m+n]e^{-j\frac{2\pi}{L}\pi k} \tag{1.1}$$

Where x is the original signal, X is the obtained signal, m is the starting point of the local DFT, k is the index of each piece, L is the length of each piece. This way, we can get a time varying spectral information which we need to show it in one plot. Since the STFT is a complex function with variables, we need a four-dimensional plot to plot it properly, which is impossible. So, what we are going to do is color-code Fourier's magnitude $|X[m;k]|$ and use dark color, for low values and whitish color for high values and place the spectral slices one by one then we are going to take the logarithm of the magnitude 10 $log_{10}(|X[m;k]|)$ to compress and better map the variety of values associated with magnitude on a color scale to obtain a plot where the horizontal-axis is represented by the variable m and the vertical-axis is represented by the variable k.

Based on the sampling period Fs [1], where Fs=$\frac{1}{Ts}$, we can label our x-axis with the width of time slices (L*Ts) and the y-axis with the frequency resolution ($\frac{Fs}{L}$) to finally get our spectrogram (also known as log-spectrogram since we have used the logarithm of the magnitude) according to both time and frequency variables (Figure 1.4). Dark colors are used for small DFT coefficients, which means that the black areas in the Figure 1.4 indicate the silence regions between the three chunks. Consequently, the circled bands

[1] More details in Annex A

below correspond to high values of the DFT.

#### MFCC

The MFCCs are a set of features that we obtain when we log-scale the mel-spectrogram and then use the discrete cosine transform. The Mel-scale is a psychoacoustic scale of pitches of sound, in the sense of their identification between the low and the high, whose unity is Mel. It corresponds to subjective approximation of the psychological sensation of pitch of sound [16]. In order to switch from the f Hertz to m Mel, we use the following formula [17]:

$$m = 2595log_{10}(1 + \frac{f}{700}) \tag{1.2}$$

To put it simply, the MFCC consists of extracting a predefined number of features from each frame of an audio file. So, in essence, MFCC are vectors of values of varying sizes, the length of which is determined by the duration of the audio. However, the Log-Mel Spectrogram was shown to be efficient in distinguishing features and recognizing emotions [18][19].

### 1.6.2 Classification algorithms

With the emergence of deep learning models, computer scientists are increasingly turning to neural networks to tackle challenges in a variety of domains. That is because neural networks outperform conventional machine learning methods [20]. We used Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformers in our contributions throughout this thesis.

#### Convolutional Neural Networks (CNNs)

Convolutional neural networks are a subset of the neural network family. They are purpose-built for image processing. Their design is therefore more precise: it is built of three major blocks [21]: a convolutional layer, a pooling layer, and a fully connected layer.

- The convolutional layer: convolution is a mathematical operation that takes an image matrix I and a filter K as inputs. Convolutional neural networks are built around the convolutional layer. Its objective is to determine the existence of a collection of

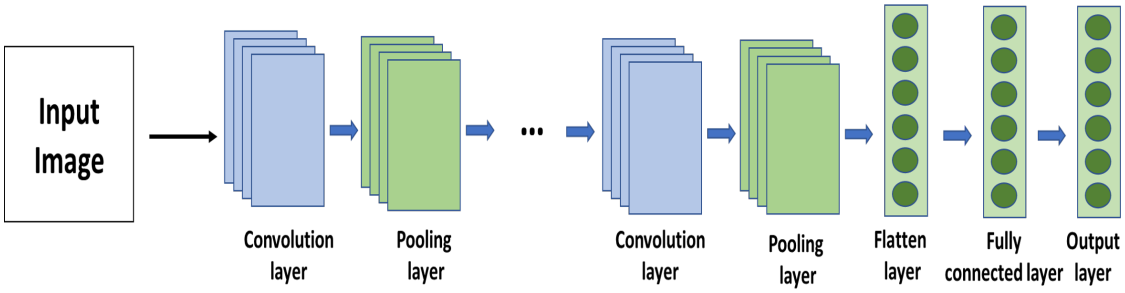


Figure 1.5: General architecture of a CNN

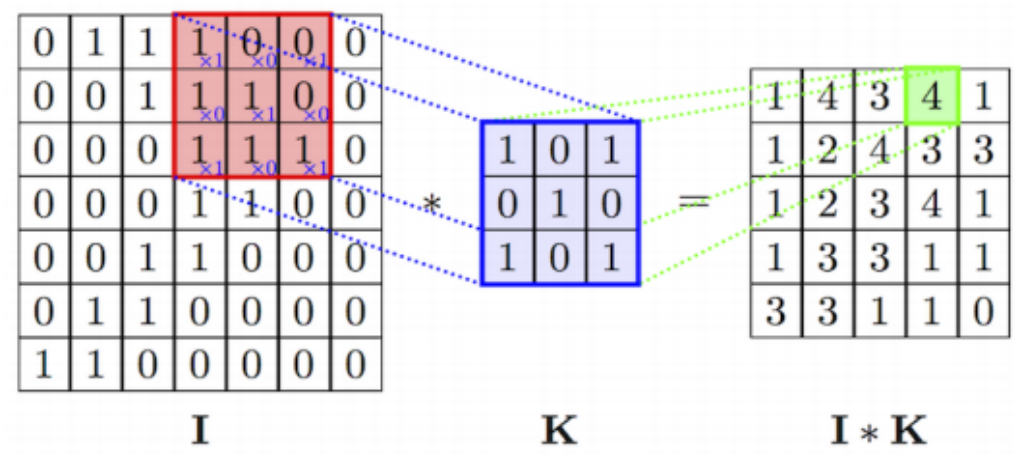


Figure 1.6: An example of a Convolution process[22]

features in the input figure. The filter is also known as Feature Map. Changing the filter, would perform different operations, such as vertical edge detection, horizontal edge detection etc. The parameters of the feature map are learned while training the network so that they automatically detect different objects. Every time we apply a Convolution, the image shrinks and a lot of information are thrown away from the edges. To prevent this from happening we can use pad the image with zeros

- Pooling layer: This layer is often located between the two convolutional layers. It takes numerous feature maps as input and performs a pooling operation on each of them, such as Max Pooling or Average Pooling.

- FC layer: The Fully-connected layer classifies the input image of the network. We flatten our matrix into vector and feed it into a FC layer. This layer performs the same functions as ordinary neural networks and makes the effort to provide classification scores.

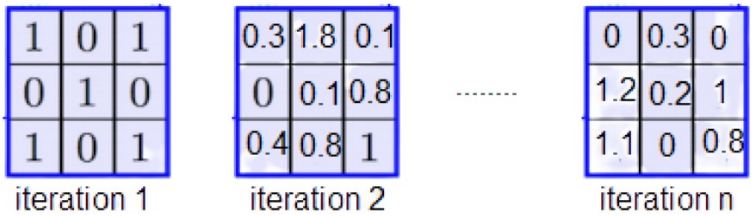


Figure 1.7: Parameter learning inside CNN

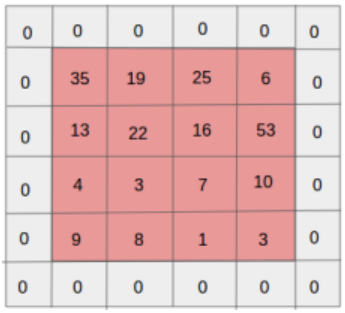


Figure 1.8: An example of a Padding process

**Recurrent Neural Networks (RNNs)**

RNNs are generally used when handling sequence data. These data can be sentences to translate, lyrics, videos, DNA sequences, audio files etc. We can apply RNNs in many areas: Speech Recognition, Music Generation, Sentiment Classification etc. Figure 1.11 shows the general architecture of an RNN where $x^i$ is the $i^{th}$ input ($i^{th}$ word or $i^{th}$ audio frame...), $y^i$ is the $i^{th}$ output, $a^i$ is the $i^{th}$ activation and $W_{aa}$ is the weight matrix. Note that all the layers share the same weight matrix, which allows the RNN to work with different length inputs (for example sentences with different number of words).

There are 4 types of RNNs: the first is many to many where the number of inputs equals to number of outputs (Figure 1.12, Top left) and it is used for example when the input is the words of a sentence and we want to determine for each word if it is a person's name or not). The second is one to many (Figure 1.12, Top right) and it can be used in music generation. The third is many to one (Figure 1.12, Bottom left) and it can be used in sentiment analysis where the inputs are the words of a sentence and the output is an integer (from 0 to 5). The fourth (Figure 1.12, Bottom right) is many to many where the number of inputs and outputs could be different. It is useful in machine translation where the input is a sentence in a language and the output is its translation in another language.

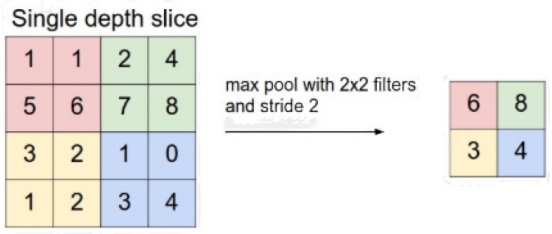


Figure 1.9: An example of a Max-Pooling process [23]

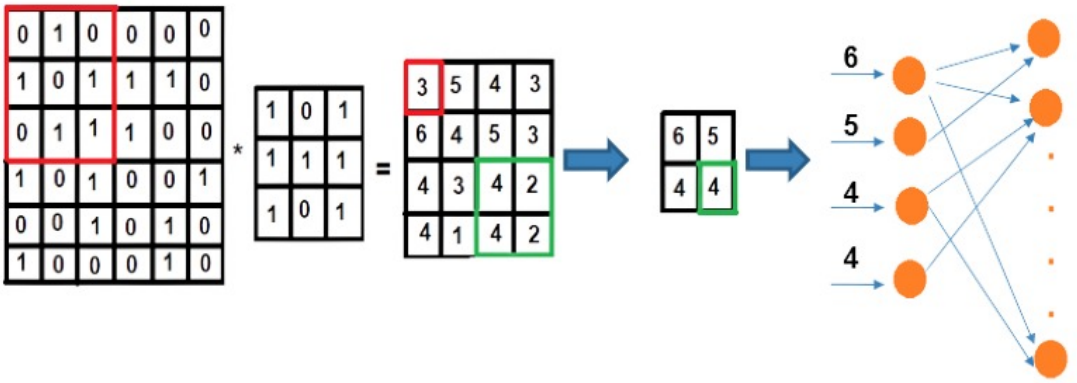

Figure 1.10: Example of calculation inside a CNN

A traditional RNN is not very good at capturing long-term dependencies. In Speech recognition for example, we need the RNNs that memorize what they have just seen (a singular or plural name), so that later in the sequence they can conjugate the verb either in plural or in singular. As a result of the diminishing gradient, RNNs cannot recall long-term dependencies. LSTMs are designed specifically to alleviate complications associated with long-term dependence. They may be thought of as basic RNNs with an additional unit. We will use a **memory cell C** which will provide some memory to remember if the name was singular or plural.

The idea is simple: we will add another parameter in each layer. In a simple RNN, each $i^{th}$ layer takes the activation of the previous layer $a^{i-1}$, but with an LSTM, each $x^i$ layer will take $a^{i-1}$ as well as $c^{i-1}$, where:

$$\hat{C}^{(<t>)} = tanh(W_c * [a^{<t-1>}, x^{(<t>)}] + b_c) \tag{1.3}$$

$$\Gamma_u = sigmoide(W_u * [a^{<t-1>}, x^{(<t>)}] + b_u) \tag{1.4}$$

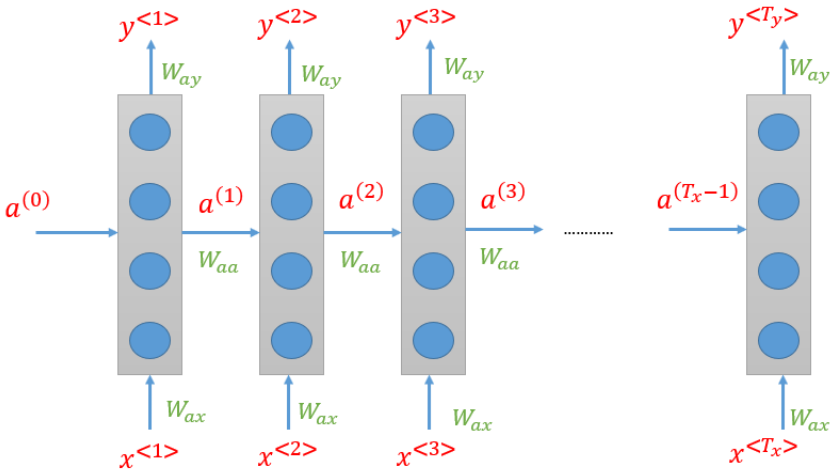


Figure 1.11: General architecture of RNN

$$\Gamma_f = sigmoide(W_f * [a^{<t-1>}, x^{(<t>)}] + b_f) \tag{1.5}$$

$$\Gamma_o = sigmoide(W_o * [a^{<t-1>}, x^{(<t>)}] + b_o) \tag{1.6}$$

$$C^{(<t>)} = (\Gamma_u * \hat{C}^{(<t>)}) + (\Gamma_f * C^{(<t-1>)}) \tag{1.7}$$

$$a^{(<t>)} = (\Gamma_o * tanh(C^{(<t>)}) \tag{1.8}$$

Additionally, BiLSTMS are a viable alternative to the LSTM in certain situations. A BiLSTM employs two LSTMs across two directions, with the first taking the series of data ahead and the second taking the series of data backward, such that effectiveness may be boosted by context information. Even though the LSTM overcomes several drawbacks, BiLSTMs offer a superior structure. They provide excellent results since they better comprehend the context, allowing models to have both backward and forward knowledge about the sequence at each time step. The difference between this method and unidirectional computation is that when the LSTM goes backward, it maintains information from the future, however when the two hidden states are merged, you may preserve information from both the past and future at any point in time. Because of their enhanced capacity

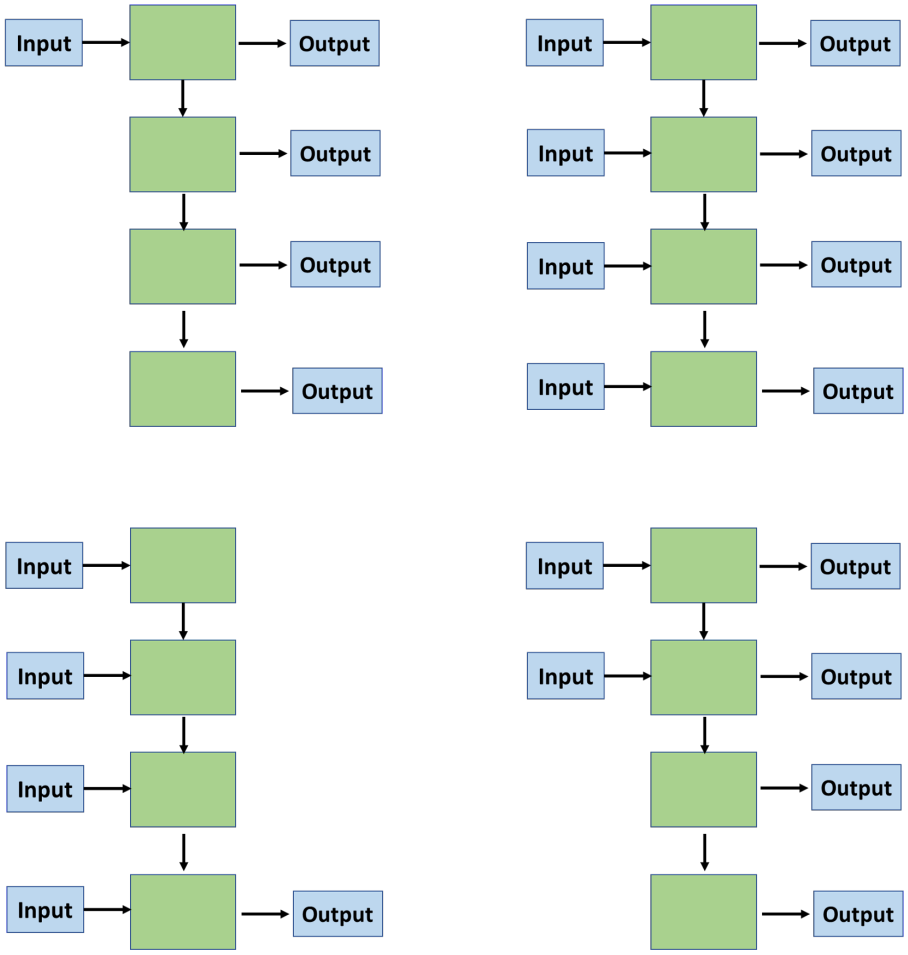


Figure 1.12: The different types of RNN

to perceive context, BiLSTMs provide great results.

### Transformers

Transformers has gotten a lot of attention recently, and with cause. They have swept over the world of NLP in recent years. The Transformer is an architecture that leverages Attention to dramatically enhance the performance of deep learning natural language processing translation models. It was initially described in the article Attention is All You Need and soon established itself as the default standard for the majority of text data applications. Transformers were created to address the issue of sequence transduction, also known as neural machine translation. That is, any work that converts an input sequence to an output sequence qualifies as a transformation task. This covers recognition of speech, text-to-speech conversion, and so on. It is required for models to possess memory in order to accomplish sequence transduction. A model must be developed to

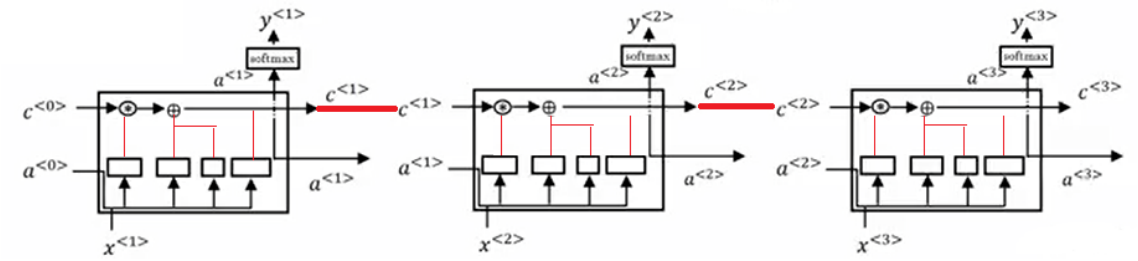


Figure 1.13: An example of LSTM architecture [24]

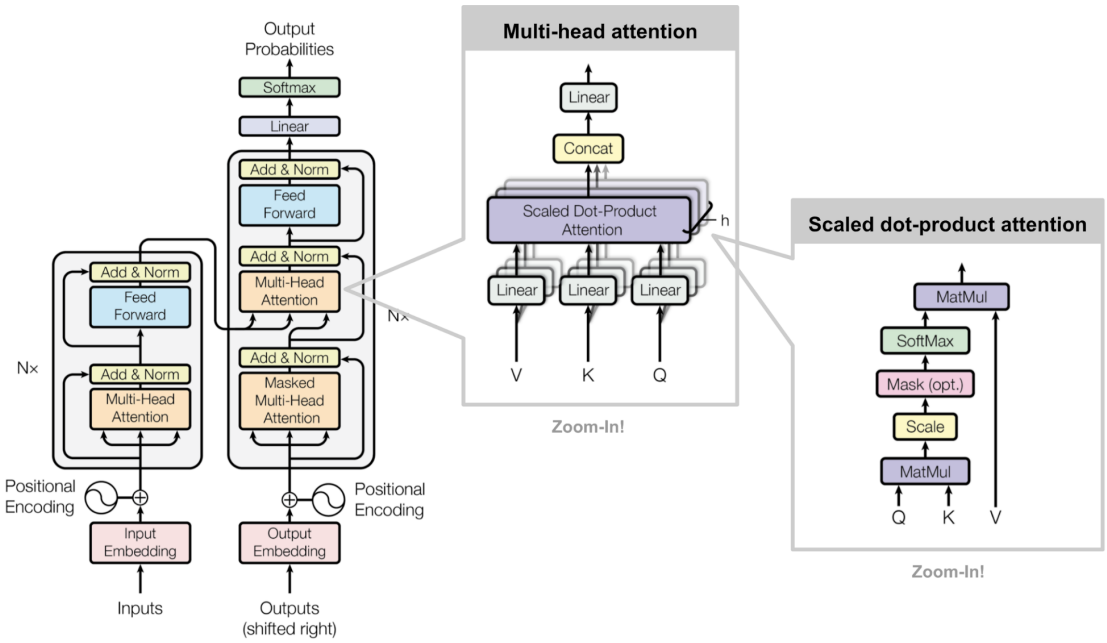


Figure 1.14: General architecture of a Transformer [25]

determine such dependencies and linkages in order to translate such phrase. RNNs, including LSTMs and GRUs (Gated Recurrent Units), were the default architecture for all natural language processing (NLP) systems until Transformers dethroned them. When the Attention mechanism was initially created, it was utilized to improve the performance of RNN-based sequence-to-sequence models. They did have two drawbacks, however: To begin, it was difficult to manage long-range dependencies between words spread widely apart in a long sentence, and second, they process the input sequence sequentially, one word at a time, which means that they cannot perform the computation for the next time step until the previous time step has been completed. This causes training and inference to be more time consuming.

As an aside, using CNNs, all of the outputs may be calculated in parallel, allowing

convolutions to be performed considerably more quickly. When it comes to dealing with long-range dependencies, convolutional layers have several limitations: in a convolutional layer, only portions of the picture (or words, if the convolutional layer is applied to text input) that are near enough to fit inside the kernel size may interact with each other. It takes a much deeper network with more layers to connect things that are located farther away. The Transformer design overcomes each of these drawbacks simultaneously. It completely eliminated the use of RNNs and relied only on the advantages of Attention to achieve its goals: All of the words in a sequence are processed simultaneously by these algorithms, significantly speeding up computing.

The transformer's primary component is the multi-head self-attention mechanism. The transformer perceives the encoded representation of the input as a collection of key-value pairs, with both dimensions equal to the length of the input sequence. The previous output is compressed into a query (of dimension) in the decoder, and the subsequent output is generated by mapping this query to the collection of keys and values.

The transformer uses a technique known as a scaled dot-product: the output is a weighted sum of the values, and the weight that is assigned to each value is determined by the dot-product of the query with all of the keys:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{n}})\mathbf{V} \tag{1.9}$$

To achieve their goals, the transformers make use of a Multi-head Attention mechanism module that is founded on Self-attention. This is the key to their success. An attention process known as self-attention, which is sometimes referred to as intra-attention [26], connects separate points in a single sequence in order to compute a representation of that sequence. It is common practice to use many instances of the Multi-head Attention method all at once. The outputs of the individual focus are then combined and linearly transformed into the anticipated dimension.

## 1.7 Datasets used throughout this thesis

There is a variety of datasets available in the field of speech emotion identification, however only a small number of them are made available free of charge to researchers

(Further information in the next chapter). A pair of commonly used datasets are used in this thesis: the RAVDESS (Ryerson Audiovisual Database of Emotional Speech and Song) dataset and the RML (Ryerson Multimedia Research Lab) dataset. These are two of the most well-known datasets in the world today. We chose to work with the RML dataset because it is almost the only free dataset that contains multiple languages, whereas this thesis is concerned with Speech Emotion Recognition in a multilingual context. We also chose to work with the RAVDESS dataset because it has been used by most recent papers to test their models, allowing us to make comparisons and perform analysis.

The RML dataset consists of 720 audios performed by 8 people. It was recorded in 6 different languages (English, Italian, Mandarin, Urdu, Punjabi and Persian) using 6 fundamental feelings: Disgust, Happiness, Fear, Anger, Surprise and Sadness. Since it is and audiovisual dataset, we converted each file to a wav format. Two files were eliminated. The first was a duplicate and the second had no speech.

The RAVDESS is an English dataset recorded by 24 actors (12 male and 12 female). The audio-only part of this dataset contains 1440 files. The used expressions in this dataset are recorded under two levels of emotional intensity (strong and normal). In normal intensity 8 categories are identified (Happy, sad, disgust, neutral, calm, angry, surprised and fearful) whereas in strong intensity 7 categories are identified (angry, sad, happy, calm, disgust, fearful and surprised).

## 1.8 Conclusion

There are several applications where a speech emotion recognition system might be handy. We have discussed what emotions are in this chapter and why it is crucial to develop a robust speech emotion recognition system to enhance human-computer interaction. Along with the frameworks and models required for our contributions, we also outlined the overall architecture of a SER system. The datasets we utilized to evaluate our model and contrast our findings with the state of the art are provided in the final section of this chapter. The state of the art, recent developments, and most significant contributions are covered in detail in the next chapter. The following chapters go into depth about our contributions.

# Chapter 2

# State of the Art

The aim of the current chapter is to present an overview of the main works on the automatic recognition of emotions since there exists a lot of studies that have been conducted in the aim of building up a potent SER system. Also, the accent will be put on related works on SER along with the databases that are used for inferring the emotion from the speech.

## 2.1 Emotional Datasets

The accuracy of the SER system is related to the features (the inputs in general) and the classifying method along with the nature of the database.

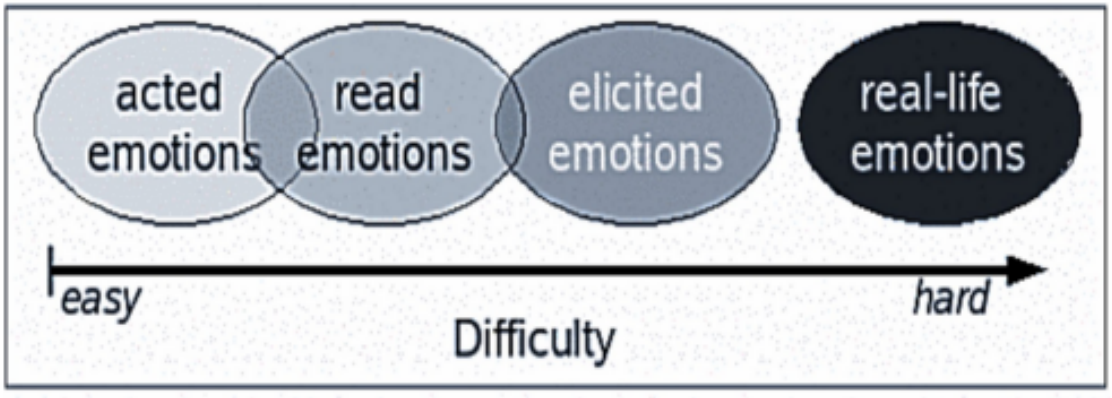


Figure 2.1: Types of emotional datasets [27]

There are three major types of databases [9]:

- Acted emotions: it is when a group of people are asked for example to read some

sentences while showing some fake emotions

- Elicited emotions: it is kind of similar to the acted emotions but people have to bring out their emotions in a way that it looks so real.
- Natural (Real-life) emotions: it is when the speech recorded in a situation where speaker expresses its real feeling. For example, a sad speech of someone in the funeral of one of his relatives.

The difficulty of the database increase whenever the spontaneity increases (as shown in Figure 2.1).

Table 2.1 puts the light on some of emotional speech databases. The researches on SER started in acted speech then shifted toward more naturally data [9] and that is what explain the dominance of acted databased over natural databases.

| Name | Language | Accessibility | Type | Contents | Emotions |
|---|---|---|---|---|---|
| RECOLA [28] | French | Only free for Academics and non-profit organizations | Natural | 9.5 hours of audio, visual, and physiological recordings of 46 French speaking participants | Multiple emotions |
| IEMOCAP [29] | English | Exclusive | Acted | 12 hours of audio and visual data | happiness, anger, sadness, frustration and neutral |
| Berlin emotional database [30] | German | Free access | Acted | 800 audio files for 5 women (between 21 and 35 years old) and 7 men (between 25 and 32 years old) | Anger, joy, sadness, fear, disgust, boredom and neutral |

Table 2.1: Example of emotional datasets

.

.

| Name | Language | Accessibility | Type | Contents | Emotions |
|---|---|---|---|---|---|
| Danish emotional database [31] | Danish | Open access with license fee | Acted | Acted Audios of two women (34 and 52 years old) and two men (38 and 52 years old) | Anger, joy, sadness, surprise, neutral |
| INTERFACE [32] | 4 different languages | Commercially available | Acted | Number of audios: 186 in English, in for Slovenian, 184 in Spanish and 175 in French | Anger, disgust, fear, joy, surprise, sadness, slow neutral, fast neutral |
| Natural [33] | Mandarin | Exclusive | Natural | 388 audios of 11 people (from call Center) | Anger, neutral |
| TESS [34] | English | Free access | Acted | 2800 audios by two female actresses (aged 26 and 64 years) | Anger, disgust, fear, happiness, pleasant surprise, sadness, and neutral |
| Korean emotional speech [35] | Korean | Exclusive | Acted | 400 audios by 4 speakers | joy, sadness, and anger |
| RAVDESS | English | Free access | Acted | 7356 audios created by 24 professional actors (12 male and 12 female) | Neutral, calm, happy, sad, angry, fearful, surprise, and disgust |
| Japanese emotional speech [36] | Japanese | Exclusive | Elicited | 4800 audios recorded by six speakers (3 males and 3 females) | Neutral, anger, pleasure, sorrow, joy, sad, normal, grief |

Table 2.1: Example of emotional datasets (continued)

| Name | Language | Accessibility | Type | Contents | Emotions |
|---|---|---|---|---|---|
| CASIA [37] | Chinese | Access on demand with fees | Acted | 500 audios by four professional actors | Happy, sad, surprise, angry and normal |
| RML | 6 different languages (English, Mandarin, Urdu, Punjabi, Persian, and Italian) | Free access with demand | Acted | 720 audios performed by 8 people | Anger, disgust, fear, happiness, sadness and surprise |
| EMOVO [38] | Italian | Free access | Acted | 588 audios recorded by six experienced actors (3 males and 3 females) | Disgust, joy, Fear, anger, surprise, sadness and neutral |

Table 2.1: Example of emotional datasets (continued)

## 2.2 Related works

Few papers relating to emotion change detection were proposed, compared to the thousands of papers devoted to speech emotion recognition.

### 2.2.1 Emotion recognition

The approach of Lima M. and Sajin S. [10] consist of generating the spectrogram of the speech then transform it to a GLCM (Gray-Level Co-Occurrence Matrix) [39]. The GLCM is a widely used technique in the domain of texture analysis. It employed basically to perform feature extraction. And in simple words, the GLCM is 2-dimensional vector that contains some derived information from the spectrogram. In their work, this technique is used to extract texture features (standard deviation, energy, mean, and entropy). The Gray-Level Co-Occurrence Matrix is fed to an SVM (support vector machine) to classify the emotions. Two different experiments have been conducted on EMODB database. The first consists of using only male speeches for training and testing and the second consists of using only female speeches for training and testing.

To extract feature, Siddique L. et al [11] used eGeMAPS (The Geneva Minimalistic Acoustic Parameter Set) feature set [40], which is one of the most used sets in the emotion recognition field and it contains 88 features. The features will be fed to a DBN (Deep Belief Network) which is basically a set of RBMs (Restricted Boltzmann Machines) stacked together. The visible layer and the hidden layer make up RBM, which is a small network with just two nodes. Every visible layer node is linked to every other node. By comparing the reconstructed input to the original input, it learns how to detect patterns in our data on its own and is then used to improve the quality of its results.

The work of Kun-Ching Wang [41] is based on six main steps. A time scaling will be applied on the input speech unify the length of the signal using the TSM (time scale modification) algorithm [42] then a gray-scale spectrogram is generated. After that, a Cubic Curve [43] will be applied on the spectrogram to upgrade its contrast to smooth the process of features extraction which will be done by using 5 Law's Masks. The original image (spectrogram) will be convoluted with every musk. At the end, we will end up with three 42-dimensional vectors which will be fed to an SVM classifier.

Jianfeng Z. et al [18] used the raw audio and its corresponding spectrogram as inputs to their model. Their model consists of a combination of LFLB (Local Feature learning Block) along with LSTM (Long-Short Term Memory) . The LFLB which is a substitute of CNN, consists of one convolutional layer, one batch normalization layer to transform the values to zero mean and unit variance, one exponential linear unit layer (also known as the activation layer) and one max-pooling layer. The model is constructed by stacking 4 LFLBs, 1 LSTM layer and 1 fully connected layer. They used two different architectures: the first is called 1D CNN LSTM where the convolutional layer and pooling layer in LFLBs are one-dimensional and the second is 2D CNN LSTM where the convolutional layer and pooling layer in LFLBs are two-dimensional. The input for the 1D CNN LSTM was the raw audio and the input for the 2D CNN LSTM was the Mel-Log-Spectrogram.

The method of Seyedmahdad M. et al [44] is based on RNNs. They have built 4 different models to address different issues. Their starting point boils down to paying more attention to frames since the speech will be divided into frames before it will be fed to the RNN and we can't guarantee that a frame's label will definitely represent the overall emotion because, for example, we may have silence within the speech. The first model,

which they called it "RNN–frame-wise", intends to appoint the overall speech's emotion to every single frame. It is a many to many RNN where each input frame has its own output label and the model is trained by back-propagating errors from every frame. The second model, which they called it "RNN–final frame", is designed to choose one RNN representation (that will be the final representation) and pass it to the output layer to determine the signal's emotion. The third model, which they called it "RNN-mean pool", perform a mean pooling on all the RNN hidden representations and then passes the result to the output layer to decide the emotion. Although the third model suffers from the silent frames issue, it performed well comparing to the two previous models. The fourth and final model, is called "RNN–weighted pool with attention". It based on the attention model mechanism that was firstly presented in machine translation. The attention is a technique that we use when training the network to teach it on which frames it should be focusing. In this model, the weighted pooling with local attention is used so that the network could be able to determine by itself which frames will represent better the speech when it comes to the overall label.

Promod Y. et al [19] proposed 3 models. The first consists of using a CNN with the phonemes as inputs since both textual and non-textual (such as laugh) information can be extracted from a set of phonemes. A phoneme is a unit of sound that differentiate words from each other. 47 phonemes were employed and an embedding generated using IEMOCAP was used to represent each phoneme as 100-dimension numeric vector. Each audio, which will be in the format of phoneme sequence, where each phoneme is represented by an embedded vector. The phoneme sequence will be fed to a CNN model composed of 1 convolutional layer, 1 pooling layer, 1 fully connected layer and one output layer to predict the emotion. Since the size of the input data in CNNs should have a fixed size, the maximum length for the input sequence is fixed at 512. The second model consists of feeding a spectrogram to a CNN composed of 4 parallel 2D convolutions and each convolutional layer is followed by a pooling layer. The features generated in the max-pool layer are flattened and fed to two successive fully connected layers. Then an output layer is added to perform classification. The third and last model is a Multi-channel CNN with both phoneme and spectrogram as inputs. This model takes the phoneme and feed it to 1 convolutional layer, 1 pooling layer and 1 fully connected layer and simultaneously takes

the spectrogram and feed it to 4 parallel 2D convolutions where each convolutional layer is followed by a pooling layer, the stack all the features in 1 fully connected layer. The features in both the fully connected layer of the first channel and the fully connected layer in channel two are combined in one fully connected layer followed by an output layer to make predictions.

The model of Yanfeng N. et al [17] consists of using a DRCNN (Deep Retinal CNN) with spectrograms as inputs. The architecture of the network is based on the AlexNet which is composed of 5 convolutional layers, 3 pooling layers and 3 fully connected layers (the last fully connected layer in AlexNet is replaced by an output layer to get the DRCNN). The idea behind the name of their model is based on imaging principle of the retinal and the convex lens, which states that no matter how long the distance between the human's eyes and an object, we would still recognize it, i.e. it does not matter if we get closer or far from an object, we can identify it. This principle is employed in an algorithm that aims to generate new data, which they called it DAARIP (Data Augmentation Algorithm Based on Retinal Imaging Principle). The DAARIP works as follows: for each spectrogram, we will maintain the original picture, generate x spectrograms smaller than the original and generate y spectrograms bigger than the original then convert all images to same size as the original. On the IEMOCAP, they achieved 41.54% accuracy on the original database and 99.25% accuracy on the augmented data. No experiments have been conducted on the original EMODB and SAVEE databases. However, with data augmentation, they have achieved 99.79% accuracy on EMODB and 99.43% accuracy on SAVEE.

The approach of Noushin H. and Hasan D. [45] comprise dividing the speech to small frames with equal length where each frame is overlapping 50% with the previous chunk. For each frame, they extracted 88 features and its corresponding spectrogram in a way that if a speech is divided to n frames, then the speech is represented with n spectrograms and 88xn matrix. For each speech, a k-means clustering algorithm is performed on the feature vector (with k=9) to select k most discriminant frames. The corresponding spectrograms of the k selected frames will be stacked together to build a 3D tensor. So finally, each speech is represented with 3D tensor which will be fed to a CNN composed of 2 convolutional layers, 2 pooling layers, one fully connected layer and an output layer.

While most of the researches are based on spectrograms, which carry information about

the amplitude of the speech, Lili G. et al [46] have built their model with the assumption that phase information may also be useful for emotion recognition from speech. The phase is represented with two different ways: the first is the modified group delay (MGDCC) which is a mathematical formula proposed by Hegde et al. [47] where the digital signal and a certain frequency are the inputs and the second is the relative phase which was proposed by Wang et al. [48] and it is obtained by fixing the phase of a certain base frequency and estimating the phase of the other frequencies relatively to the constant phase. They model consists of dividing the speech into fixed length pieces, then generate the corresponding spectrogram of each segment which will be used to extract both the relative phase and the MGDCC. All these features are stacked together in one large feature vectors V. The vector V is fed to a CNN in order to extract deep features that will be themselves fed to BLSTM in order to make the decision. Several experiments have been conducted on EMODB database with varying the inputs each time.

Nithya R. and Prabhakaran M. [49] used a CNN to extract emotions from speeches. Their idea consists of using TL (Transfer Learning). The TL is a method we want to employ a pre-trained model in our task to gain time and computation cost. It is done with freezing the weights of the first layers and train the weights of last layers with our training data. They have used Inception Net v3 model. Since the DL models can be used either as classifiers with pre-extracted features or as a way to automatically extract features and classify objects, some works are based on the hand-crafted features as inputs and the DL models as classifiers.

John W. and Rif A. [50] chose to work with 20 acoustic features from the total 88 features extracted with eGeMAPS. The features are extracted every 10 msec. Considering that the duration of a certain speech may not be the same as others, the size of the input vectors will definitely be different. To remedy this problem, each feature vector will be either truncated or zero-padded to 512 frames, and that is according to the duration of the speech. The 20x512 feature vector is fed to their model which the call EmNet. The EmNet is composed of a convolutional layer followed by a pooling layer to extract local features, then another convolutional layer followed by a pooling layer to extract global features. The output of the global convolutional layer is delivered to an LSTM layer which was followed by an output layer.

Leila K. et al [51] decided to work with 60-dimensional MFCC feature vector as well as 95-dimensional MSF (Modulation spectral features) feature vector. Both feature vectors are fed to various classifiers such as MLR (Multivariate Linear Regression), SVM and RNN. Two databases were used to evaluate this work: EMODB database and the INTER1SP Spanish emotional database. When working with EMODB database, the MLP classifier's best result was 75.90% ± 3.63% accuracy and it was obtained with both MFCC and MSF as inputs, the SVM's best result was 63.30% ± 4.99% accuracy and it was obtained with only MSF as input and the RNN's best result was 69.55% ± 3.91% accuracy and it was obtained with only MFCC as input. When working with Spanish database, the MLP classifier's best result was 82.41% ± 4.14% accuracy and it was obtained with both MFCC and MSF as inputs, the SVM's best result was 77.63% ± 1.67% accuracy and it was obtained with only MSF as input and the RNN's best result was 90.05% ± 1.64% accuracy and it was obtained with both MFCC and MSF as inputs.

Aouani and Ayed [52] retrieved a vector of 42 characteristics from each signal in their study. MFCC, Zero Crossing Rate (ZCR), Harmonic to Noise Rate (HNR) and Teager Energy Operator (TEO) are the characteristics that are utilized. They have chosen to use an Auto-Encoder (AE) to obtain a simplified data representation and to pick relevant characteristics rather than the 42 features. An SVM is used to categorize utterances and identify emotions using the output of the AE.

In the research of Mustaqeem et al. [13], an utterance is divided into many frames, and then identical frames are clustered using a k-means algorithm. One keyframe from each cluster, chosen because it is closest to the centroid, will be utilized to create a spectrogram. The classification approach has been implemented using the transfer learning methodology. The already-trained Resnet101 has been deployed. In order to recognize emotions, the CNN's output will be normalized and fed into a bidirectional long short-term memory (LSTM).

A large number of feature sets were integrated in the work of Issa et al. [12]. MFCCs, Mel-scaled spectrogram, Chromagram, Spectral contrast feature, and Tonnetz representation were all utilised. There are a total of 193 features. All characteristics were layered together and input into a One-dimensional CNN to accomplish classification and emotion identification.

The preprocessing phase is where Mustaqeem and Kwon's work [53] is mostly concentrated. To eliminate silence, noise, and unnecessary information, they employed an adaptive threshold-based method. Their approach creates a spectrogram from each signal, which it then feeds to CNN so that it can conduct classification. Compared to the initial dataset, their preprocessing step helped increase accuracy by almost 8%.

The Mel-log Spectrogram was utilized as an input for Mupidi and Radfar's model [54]. However, instead of using a traditional neural network as a classifier, they employed a quaternion convolutional neural network (QCNN). In order to extract representations from raw audio data, Pepino et al. [55] employed the pre-trained wav2vec framework. A shallow neural network receives as inputs the extracted features, the eGeMAPS descriptors, and the spectrograms. The characteristics of wav2vec generated the highest results.

A huge dataset of spectrograms was used to train a model known as VACNN (visual attention convolutional neural network) as part of Seo and Kim's contribution [56]. Convolution blocks and spatial and visual attention modules make up the VACNN model. A model that can be applied to smaller datasets is what they are trying to achieve. A bag of visual words (BOVW) is employed to extract local features as an attention vector for a new small dataset, and a fine-tuned VACNN is employed to extract features from log-mel spectrograms. The BOVW and the VACNN outputs are subjected to an element-wise multiplication, which is followed by an element-wise sum to sum the features. A softmax layer is used to recognize emotions.

### 2.2.2 Emotion change detection

The bulk of research has been on pre-segmented speech utterances, which are given a single global label (emotion). Despite its relevance, research on emotion change detection has received less attention than research on speech emotion identification. Existing research has largely focused on either determining the time of emotion shift or forecasting valence (positive or negative) and arousal changes (low or high) (Figure 2.2).

For the emotion change detection, fewer papers have been published. In [58], authors have worked on detecting the instant of emotion change and transition points from one emotion to another. A Gaussian Mixture Models (GMM) with and without prior knowledge of emotion-based methods was used to detect emotion change among only four

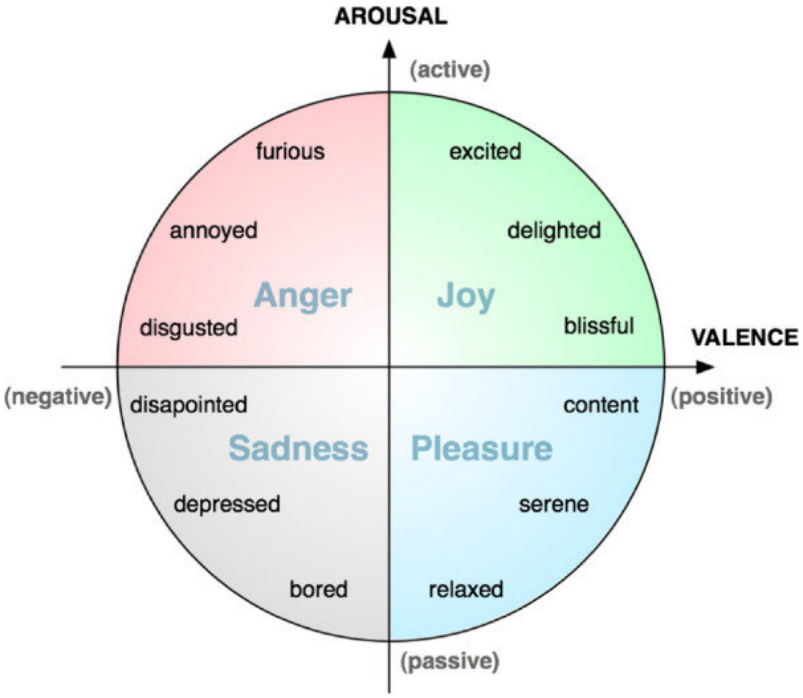


Figure 2.2: The distribution of various emotions over the arousal-valence space [57]

different emotions. However, their main focus was on arousal and valence. Their technique involves the use of a double sliding window that incorporates both prior and present fixed-length windows. Within these two windows that cover numerous frames, features are retrieved and utilized to generate probabilities depending on the frame. Scores are computed and compared to a threshold during the detection phase in order to make a decision. Scores are a linear combination of log likelihoods. A change happens when a score exceeds the threshold within the tolerance range of the actual moment of change. They utilized the Detection Error Trade-off (DET) curve and Equal Error Rate (EER) to validate their model.

In the paper [59] , authors have explored the problem of identifying points of emotional change over time in terms of testing exchangeability using a martingale framework which is a sort of stochastic process that employs conditional expectations. It occurs when a collection of random variables is repeated at a specific time. When a new data point is seen in the martingale framework, hypothesis testing is performed to determine whether a concept change occurs in the data stream or not. In this process data points (frame-based features) of speech are observed point by point. Their goal was to identify changes in emotional categories (neutral and emotional), as well as within dimensions (positive and

negative in arousal and valence). They have used two sets of frame-level acoustic features: the MFCCs and the Geneva Minimalistic Acoustic Parameter Set (eGeMAPS).

The model of Huang and Epps [60] consists of detection the emotion change points in time as well as assessing the emotion change by calculating the magnitude of emotion changes along with the types of emotion change. They have used 88-dimensional eGeMAPS features and three different regression models: Support Vector Regression (SVR), Relevance Vector Machine (RVM) and OutputAssociative RVM (OA-RVM).

Table 2.2: Summary of the most pertinent contributions

| Work | Major contribution (s) | Year |
|---|---|---|
| [41] | TSM (time scale modification) / Cubic Curve / 5 Law's Masks | 2015 |
| [44] | RNN–frame-wise / RNN–final frame / RNN-mean pool / RNN–weighted pool with attention | 2017 |
| [17] | DRCNN (Deep Retinal CNN) / DAARIP (Data Augmentation Algorithm Based on Retinal Imaging Principle) | 2017 |
| [10] | GLCM (Gray-Level Co-Occurrence Matrix) | 2017 |
| [11] | eGeMAPS /DBN (Deep Belief Network) | 2018 |
| [46] | The Modified Group Delay (MGDCC) | 2018 |
| [50] | The EmNet (Emotion Network) | 2018 |
| [19] | CNN with the phonemes / Multi-channel CNN with both phoneme and spectrogram as inputs | 2018 |
| [49] | Transfer Learning | 2018 |
| [51] | MFCC + MSF (Modulation spectral features) / MLR (Multivariate Linear Regression) | 2018 |
| [45] | A k-means clustering algorithm to select k most discriminant frames to build a 3D tensor | 2019 |
| [18] | LFLB (Local Feature learning Block) | 2019 |
| [52] | Auto-Encoder (AE) | 2020 |
| [13] | Pre-trained Resnet101 + LSTM | 2020 |
| [56] | VACNN (visual attention convolutional neural network) | 2020 |
| [54] | Quaternion Convolutional Neural Network (QCNN) | 2021 |
| [55] | Pretrained wav2vec | 2021 |

## 2.3 Conclusion

We have highlighted the most well-known emotional speech datasets in this chapter. Unfortunately, there are very few publicly accessible natural datasets, despite the fact that

natural speech is best for use in practical applications. Many acted datasets have been developed in various languages, however the average number of people in each dataset is considered to be a little low. This chapter covered a number of contributions to speech emotion recognition. While the majority of the publications concentrated on the classification part, some papers concentrated on the feature extraction part. Some of the most relevant works are listed in Table 2.2 along with their main contribution (s). A few publications on the topic of identifying changes in emotional state during conversations have also been presented. In the following chapters, we will present our contributions and findings.

# Chapter 3

# Multiple Models Fusion for Multi-label Classification in SER Systems

In this chapter, we introduce two contributions: first, unlike some previous works that propose the fusion of different feature sets and use a single classifier, we propose the fusion of the output of two different models, each of which handles a different feature set; and second, we bring a fresh vision to the field of emotion recognition by incorporating the notion of multi-label categorization.

## 3.1 Introduction

Starting from a hypothesis that we have established; we have noticed that a speech can be differently interpreted from a person to another. For the same speech, what may sound neutral to a person, may sound sad to another and so on. This hypothesis has been approved following a ground truth study that we have conducted (further information in Section 3.3.1). Giving an utterance to a person would induce the recognition of one emotion. Giving the same utterance to several people would induce the recognition of one or several emotions. From this point of view, we may claim that a machine that identifies one emotion at a time is a perception of only one human's brain. An intelligent machine is a machine that outperforms the majority of people if not all and is capable to react

and consider different scenarios based on different possibilities. Therefore, we thought of designing a multi-label classifier. This new vision will allow a machine to consider all the potential emotion categories that may be identified by humans. Such a model requires a specific dataset where each file should have one or several labels at the same time. To the best of our knowledge, all existing datasets are recorded and annotated with one label. For this reason, we have performed hand labeling on the RML dataset with the aid of 50 people. **However, to build a multi-label system, we first need to build a robust Speech Emotion Recognition model.**

## 3.2 Multiple models fusion

The process of combining numerous methods to enhance performance is referred to as multiple model fusion. There are two main categories of fusion: Aligned and Consecutive.

### 3.2.1 Aligned models

Several recent studies have focused on the emotion identification from speech. While a variety of models have been proposed, the most are based on the scheme depicted in Figure 3.1.

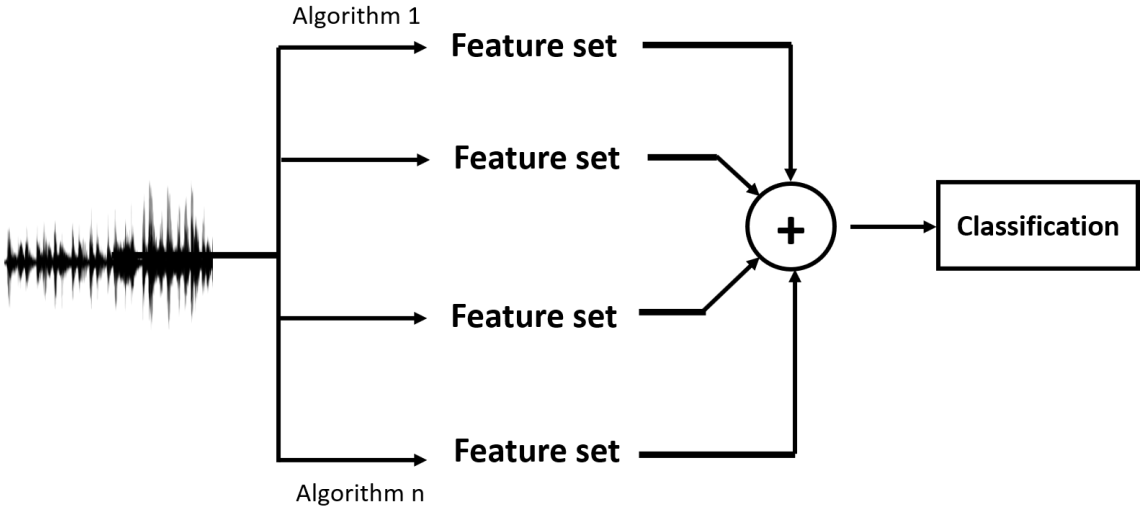


Figure 3.1: The concept of fusing various feature sets

The method comprises of processing an audio file using one or multiple algorithms to extract one or several feature sets. The features are then passed to a classifier to identify

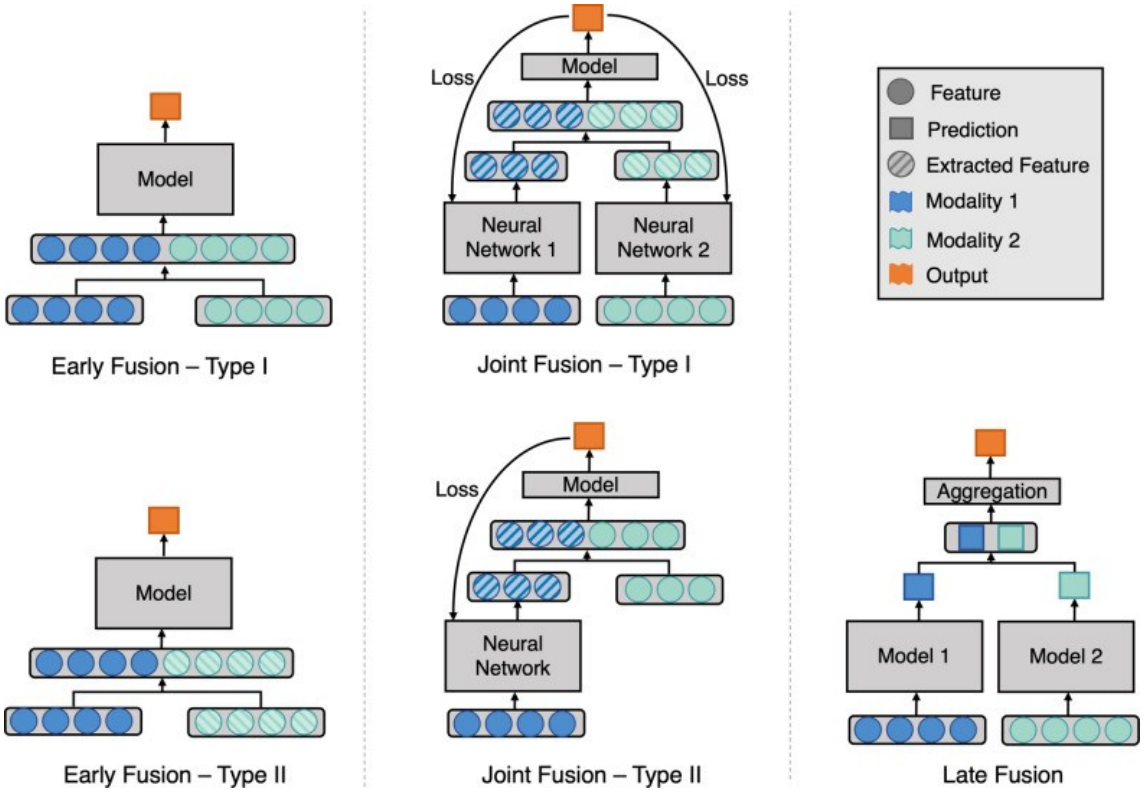


Figure 3.2: Different types of data fusion [61]

the expressed emotion in that utterance. Over the last several years, numerous feature extraction techniques have been employed to create a robust Speech Emotion Recognition system. Several studies choose to use a single feature extraction approach, whereas others merged several feature sets simultaneously. This process is known as data fusion and it has been widely used in many research fields such as health applications[61] natural language processing applications Computer vision applications, etc. There are three main fusion categories [61] as shown in Figure 3.2.

Early fusion [61], sometimes referred to as feature level fusion, is the process of integrating several input modalities into a single feature vector prior to training a single machine learning model. Connections across input modalities may take a number of forms, including concatenation, pooling, and the usage of a gated unit. Early fusion type I involves fusing the original features, while early fusion type II involves fusing recovered features by manual extraction, image analysis tools, or learned representation from another neural network. As anticipated probabilities are considered extracted features, early fusion type II refers to the process of fusing features with predicted probabilities from many modal-

ities. The act of combining learnt feature representations from various neural network layers with data from other modalities as input to a final model is referred to as joint fusion (or intermediate fusion) [61]. The primary distinction between this method and the early fusion method is that, during training, the loss is sent back to the feature extraction neural networks. As a consequence, the feature representations become more accurate with each subsequent training iteration. For the purpose of performing joint fusion, neural networks are applied. This is due to the fact that neural networks are able to transmit loss from the prediction model to the feature extraction model. Joint fusion type I is the term used to describe the process in which the feature representations of all modalities are restored. However, not all of the input characteristics are required for the process of feature extraction in order for it to be categorized as joint fusion. Late fusion, also known as decision-level fusion, is the act of combining the predictions produced by a number of different algorithms in order to arrive at a conclusion [61]. In most cases, a variety of modalities are used in the training of various models, and the ultimate conclusion is reached by compiling the results of the many models' predictions. Aggregation functions include the following: the average, majority voting, weighted voting, and a meta-classifier that is based on the predictions of each individual model. A lot of the time, the aggregation function is figured out via experimentation, and its appearance might shift depending on the goal and the input modalities.

As discussed in chapter 1, the MFCC and Spectrograms are the most well-known and often utilized feature sets for Speech Emotion Recognition systems because they are the most efficient representations for utterances in this field. A spectrogram is a visual representation of the spectrum of frequencies of a signal as it fluctuates with time. As for the MFCC, to compute them, we log-scale the mel-spectrogram and then use the discrete cosine transform. To put it simply, the MFCC consists of extracting a predefined number of features from each frame of an audio file. So, basically, MFCC are vectors of values of varying sizes, the length of which is determined by the duration of the audio. Having stated that, it will be more convenient and appropriate to deploy a distinct model for each feature extraction technique. As discussed previously, some of the previous papers [62][52][12] have suggested combining together numerous feature extraction techniques to improve the accuracy of the Speech Emotion Recognition systems. We strongly believe

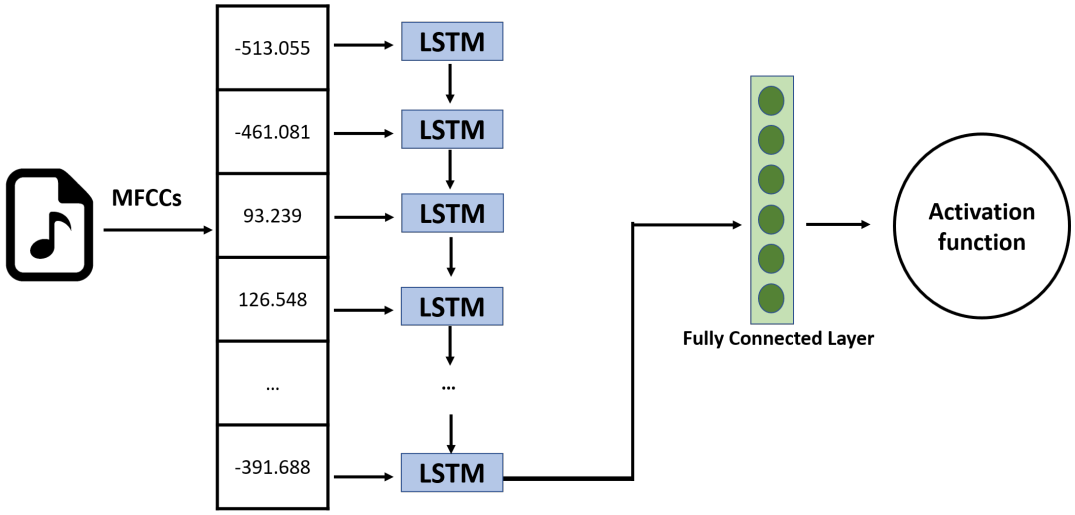


Figure 3.3: Emotion recognition using LSTM

that each technique has its own specificity. For instance, spectrograms are 2D pictures hence it is more ideal to utilize Convolutional Neural Networks as a classification model, as they are largely built for image identification applications. The fundamental benefit of the CNN over other architectures is that they automatically detect crucial elements without the need for human interaction (Figure 3.4). As for the MFCC, we suggest using the Recurrent Neural Network (RNN) layers to propagate information through the signal and to make sure that we process the frames with time notion. As mentioned earlier, the RNNs suffer from the vanishing gradient issue where an information loss may occur for extended sequences, and this can be mitigated by employing an LSTM which employs extra special units in addition to the RNN's normal units (Figure 3.3).

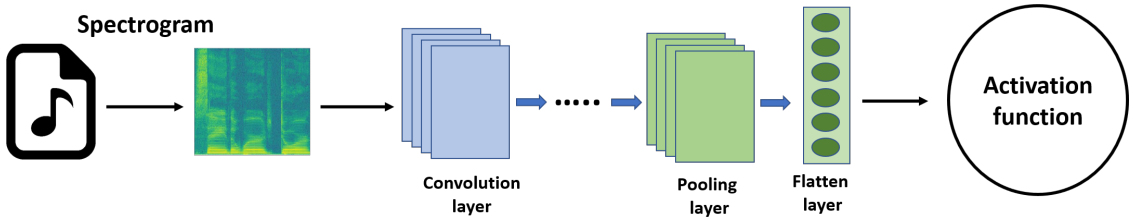


Figure 3.4: Emotion recognition using CNN

As shown in Figure 3.5, in our model, we suggest using several feature extraction techniques, but instead of combing the different feature sets together and using one classifier, we rather use different models to process the feature sets independently and then combine the outputs of all models before using an activation function that will determine the

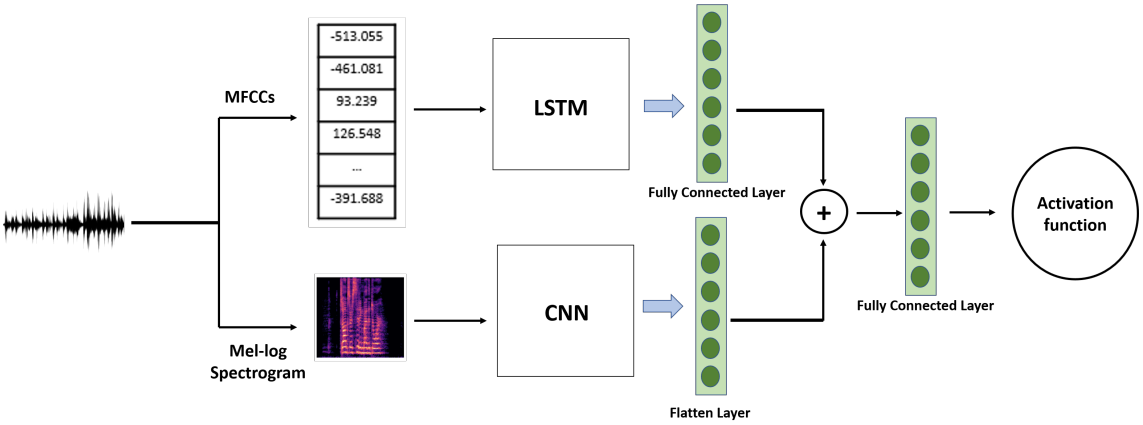


Figure 3.5: Aligned models fusion for SER systems

expressed emotion (s).

### 3.2.2 Consecutive models

Although Table 3.4 clearly illustrates that the MFCC in conjunction with the Spectrogram produces the greatest results, new research has demonstrated that the Log-Mel spectrogram is the best feature extraction algorithm [19] and can replace both the spectrogram and the MFCC.

Fortunately, with the Log-Mel Spectrogram, there will be no need for parallel classifiers since one feature set is used. However, although transformers were originally designed to work with textual data, that hasn't prevented them from being used in a variety of Computer Vision tasks, including image processing, with results comparable to convolutional neural network [63][64]. Dosovitskiy et al. proposed a Vision Transformer (ViT) [63], in which every image is divided into patches (Figure 3.6), with each patch being passed to the transformer input layer and processed similarly to a single word embedding.

Despite numerous studies demonstrating that Vision Transformers outperform CNNs, Transformers in general are still less powerful for local-invariant vision data [63], and vision transformers in particular are vulnerable against adversarial patches [65]. To avoid these concerns, we propose combining the standard transformer with CNN. The use of CNN with Transformers is not novel. It's been utilized in a lot of researches. In [66], Karpov et al. used a Transformer followed by a CNN. Zhang et al. [67] have used a CNN with Transformer in hybrid approach. Liu et al. [65] did not used a CNN but rather some

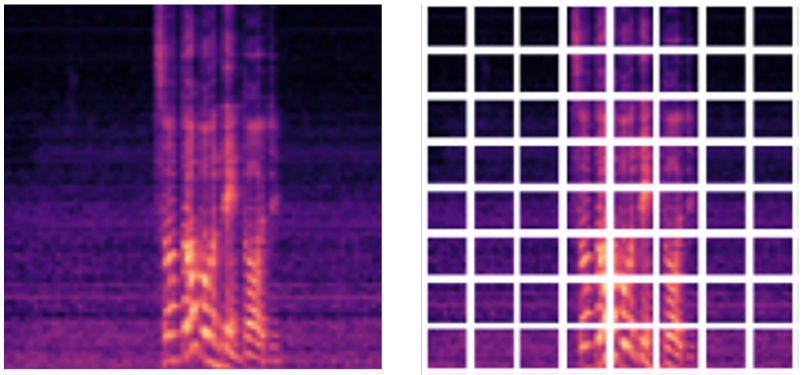

Figure 3.6: An example of a spectrogram (Left) and its patches (Right)

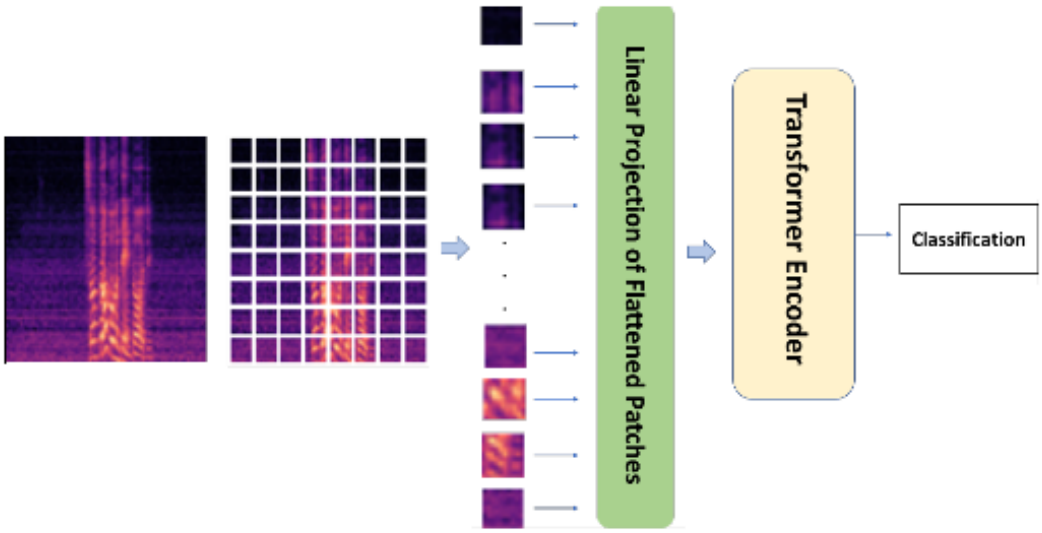


Figure 3.7: Emotion recognition using a Vision Transformer

convolution filters before the Transformer. The usage of a Time Distributed CNN with a conventional transformer rather than a vision transformer, on the other hand, is novel in this research.

The transformer accepts one embedding at a time as input, whereas the Vision Transformer accepts one patch at a time. The spectrograms however are plots with special specificity where the vertical axis presents the frequency and the horizontal axis presents the time in seconds. Rather of splitting the spectrogram into patches and feeding them to a Transformer, we want to inject a chronologically ordered series of frames (time steps). Prior to sending the frames to the transformer, we want to do per-frame convolution operations to extract key features for training the transformer. If we apply a convolution operation to each frame, each frame will have its own convolution flow, and each result

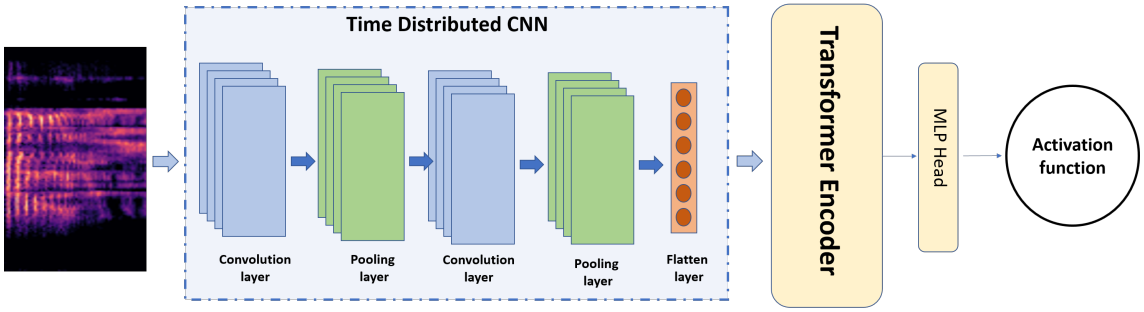


Figure 3.8: The proposed model of the first contribution

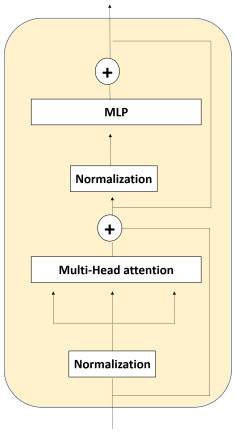


Figure 3.9: The architecture of a Transformer encoder

will be treated as a separate input for the transformer. However, if we train each convolution flow independently, we will encounter undesirable behaviours such as a lengthy and slow training process because we will need to train several convolution flows (one for each input frame), each convolution flow can have several different weights, resulting in different features detection that are unrelated, and some convolution flows will be unable to detect what other convolution flows can. We must ensure that the complete set of convolution flows can locate the same features. It is conceivable that certain convolutional flows discover something else, but this risk must be minimized.

A solution to this is to employ a Time Distributed layer, which enables us to apply a layer to each temporal slice of an input (i.e., perform the same operation on each time-step) and generate one output per input.

## 3.3 Multi-label classification

Processing an audio file in order to identify emotions conveyed is the basic objective of a speech emotion recognition system. A dataset of emotional speech with matching labels for each audio file is required in order to train such a model.

### 3.3.1 Ground Truth study

To create emotional speeches datasets, usually a set of people are requested to record their voices reading a sentence while expressing emotions. The emotion expressed by the actor is not necessarily the label of the utterance.

In the IEMOCAP dataset (see Table 3.1), three annotators were invited to determine the final label of each audio file. Each one of them was asked to listen to the speech and determine the emotion. If at least two of the three agree on specific emotion, then that emotion will be the label of the audio file regardless the emotion expressed by the actor. However, if each gives a different label from the other, then the utterance will not have a label. With regards to the RML dataset (see Table 3.2), the final label is determined by the actor, and even if the label of one of the files is set to "sad" and all reviewers agree that it should be "neutral," the final label will be definitely "sad."

Such procedure disregards an important information: the possibility of having different label. This process forces the machine to think in one and only certain way and it forbidden it to take different scenarios into consideration. Starting from here, we have conducted a Ground Truth (GT) study, in which, we have asked 50 people to listen to audio files and determine the expressed emotions. For the purpose of the study, we have used the RML dataset.

Table 3.1: Annotation of the IEOMOCAP dataset

| File | Emotion expressed by the actor | Emotion determined by: | | | Final label |
|---|---|---|---|---|---|
| | | Reviewer 1 | Reviewer 2 | Reviewer 3 | |
| 1 | Angry | Angry | Sad | Angry | Angry |
| 2 | Sad | Neutral | Neutral | Sad | Neutral |
| 3 | Happy | Happy | Angry | Fear | XXX |

In our study, the participants were aged between 17 and 65 with an average age of

Table 3.2: Annotation of the RML dataset

| File | Emotion expressed by the actor | Emotion determined by: | | | Final label |
|---|---|---|---|---|---|
| | | Reviewer 1 | Reviewer 2 | Reviewer 3 | |
| 1 | Angry | Angry | Sad | Angry | Angry |
| 2 | Sad | Neutral | Neutral | Sad | Sad |
| 3 | Happy | Happy | Angry | Fear | Happy |

27,4. The total number of males was 38. In this study we don't use the majority rule, but rather we take into consideration all possible labels, which means that each utterance could have one or more labels. At the end of the study, a single label was assigned to 684 files out of the 720, whereas 36 files had two different labels at the same time.

### 3.3.2 One class prediction vs multi class prediction

As explained in Section 3.1 and demonstrated in Section 3.3.1, emotions may be understood in a variety of ways. This concept is originally inspired by Figure 3.10, which demonstrates how a same number may be seen differently depending on someone's position. If we train a traditional Machine Learning algorithm to identify digits, the system will give each digit a probability. It will decide, for example, that the number in Figure 3.10 is 6 by 45 %, 9 by 40%, 3 by 1%, and so on, but it will ultimately choose the number with the highest probability, which is 6, without consideration for the possibility that the digit is 9. This is also applicable for emotion recognition, where the algorithm will conclude the presence of a single emotion while ignoring alternative possibilities, which is not desirable. We want the algorithm to analyze every possibility and prepare all conceivable scenarios.

### 3.3.3 One class prediction

The one class prediction consists of the common traditional scheme where an audio file is processed and one or several feature sets are extracted then fed to a classifier to classify the utterance. Within this process, only one label should be assigned to an audio file.

To do so, the activation function is set to be a softmax function which is defined as follows:

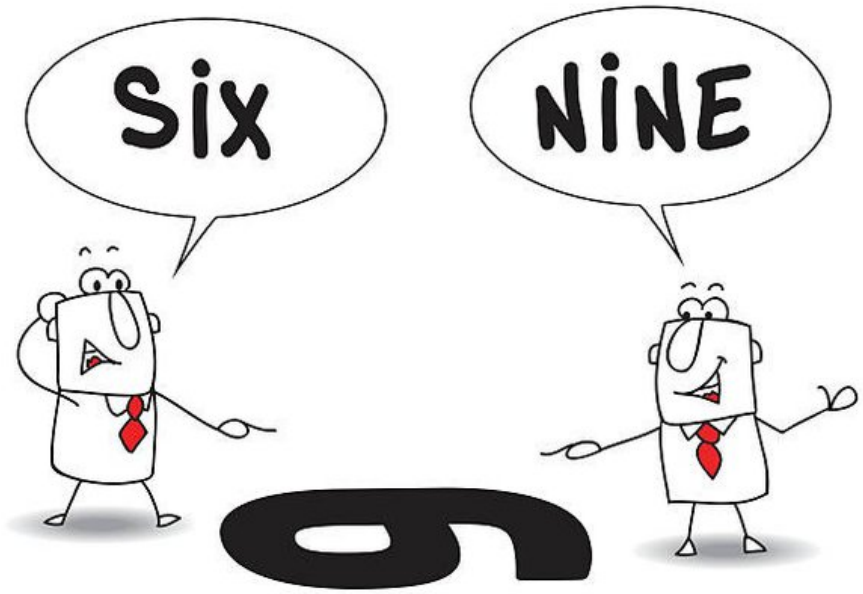


Figure 3.10: The influence of a person's position on number recognition

$$\sigma(\overrightarrow{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{3.1}$$

Where:

- K is the number of labels,
- $\overrightarrow{z})_i$ is the input of the softmax function,
- $(z)_i$ represents the elements of the input vector,
- $(e)^X$is the exponential function applied to $X$,
- $\sum_{j=1}^{K} e^{z_j}$: the normalization term at the bottom of the equation guarantees that all of the function's output values add up to one.

For a given utterance, the softmax will output K values that sum to one. Each value represents the probability that the input belongs to that class. The utterance will be assigned to the only class having the highest value.

### 3.3.4 Multiple classes prediction

This part presents our new vision to the Speech Emotion Recognition systems. As we have mentioned, people may interpret the expressed emotions differently and each one of them may assign a different emotion. With depending on majority voting and disregarding the alternative choices, a system might wrongly learn to perceive emotions and can be seen as it is forced to be biased. Furthermore, allowing a system to perform a multi-label classification, would give it the possibility to consider different scenarios.

To perform a multi-label classification, the activation function is set to be a sigmoid function which is defined as follows:

$$\delta(X) = \frac{1}{1 + e^X} \tag{3.2}$$

This function is efficient and widely used. It is mainly deployed when working with binary classification where there are only two classes. The output of the function is either 0 or 1, where 1 means that the input belongs to that class which indeed means that the input does not belong to the other class.

This function can be deployed in our proposed model by setting the number of outputs to K. This way, the output of the sigmoid function will be a vector of K binary values. If K=4 and the output of the sigmoid is **O**=[0,1,0,1], then this means that the input utterance can belong to the second or the last classes.

## 3.4 Experiments and Results

Experiments will be performed using the same data for each model in order to assess their performance and compare them.

### 3.4.1 Model tuning

The Spectrograms were created with the window size set to 2048 and the overlap set to 512, since these are the optimal settings for generating spectrograms for Speech Emotion

Recognition. For the MFCCs, we experimented with various size vectors and determined that extracting 13 features from each frame produces the best results. We examined some well-known deep architectures. We first trained each model from scratch, but the results were unsatisfactory. As a result, we ultimately chose to use shallow architectures and train them from scratch. The CNN is constructed using three convolutional layers, each followed by a Max-pooling layer. One layer of LSTM was employed for the MFCC, followed by a fully connected layer. The system's final output was activated using the softmax and sigmoid functions, while the hidden layers were activated using the ReLU function. With regards to the encoder, we fixed the number of layers to six and the model's overall dimension to 512. The multi-head attention blocks included a total of sixteen heads.The softmax and the sigmoid activation functions were used for the last output of the system whereas the ReLU activation function was used for the hidden layers. . To avoid the over-fitting, we have used a Dropout of 0.1. to compile and train the model, we have used the Adam optimizer with a learning rate equals to $10^{-4}$.

### 3.4.2 Results

#### Results with aligned model

We have conducted two experiments: the first consists of training the model to recognize one single-label and the second consists of training the model to recognize multiple labels at the same time.

Table 3.3: Results on the RML dataset.

| Work | Year | Result |
|---|---|---|
| Avots et al. [62] | 2019 | 69.30% |
| Xia et al. [68] | 2020 | 73.15% |
| Aouani. And Ayed [52] | 2020 | 74.07% |
| Issa et al. [12] | 2020 | 77.00% |
| **Ours (Single-label)** | **2021** | **83.21%** |
| **Ours (Multi-label)** | **2021** | **84.72%** |

Table 3.3 illustrates the results produced by our model with the two experiments along with comparison with state of the art.

Since all the available researches detect one single emotion and in order to have a fair evaluation, the comparison should be made using the result of Single-label accuracy.

However, in both cases we have accomplished to outperform the state of the art.

Table 3.4: The accuracy of the model on one single-label data with various feature sets.

| **Approach** | **Result** |
|---|---|
| Spectrogram + CNN. | 79.30% |
| Spectrogram + CNN. | 79.30% |
| eGeMAPS + LSTM | 65.01% |
| MFCC (42 features) + LSTM. | 72.15% |
| MFCC (21 features) + LSTM. | 71.24% |
| MFCC (13 features) + LSTM. | 75.80% |
| Spectrogram + MFCC + eGeMAPS | 80.33% |
| Spectrogram + eGeMAPS | 79.50% |
| MFCC + eGeMAPS | 73.22% |
| Spectrogram + MFCC | 83.21% |

Table 3.4 illustrates the results of different most used feature extraction techniques. We have used the top three most used techniques: Spectrograms, MFCCs and eGeMAPS. We have considered several combinations to see which ones will be more adequate. Table 3.4 clearly shows that the best result was obtained by combining the MFCC and the Spectrogram.

**Results with consecutive model**

Our approach was tested using two datasets: the RML and RAVDESS. Each dataset is partitioned into 80% train, 10% validation, and 10% test using stratified sampling to maintain the proportion of samples per class.

Table 3.5: Accuracy of the Proposed model (RML dataset)

| **Work** | **Year** | **Result** |
|---|---|---|
| Avots et al.[62] | 2019 | 69.30% |
| Xia et al. [68] | 2020 | 73.15% |
| Aouani. And Ayed [52] | 2020 | 74.07% |
| Issa et al. [12] | 2020 | 77.00% |
| **Ours** | **2022** | **83.88%** |
| **Ours (with SpecAugment)** | **2022** | **84.76%** |

We have trained the different models simultaneously using the same data. Table 3.5 and Table 3.6 illustrate the comparison of the result produced by our proposed model (Time Distributed CNN + Transformer) on the RML and the RAVDESS datasets respectively, whereas Table 3.7. illustrates the comparison between different architectures with the

Table 3.6: Accuracy of the Proposed model (RAVDESS dataset)

| Work | Year | Result |
|---|---|---|
| Hajarolasvadi and Demirel [45] | 2020 | 68.00% |
| Mustaqeem et al. [13] | 2020 | 71.61% |
| Muppidi and Radfar [54] | 2021 | 77.87% |
| Mustaqeem and Kwon [53] | 2020 | 79.50% |
| Mustaqeem and Kwon [69] | 2020 | 80.00% |
| **Ours** | **2022** | **82.72%** |
| Seo and Kim [56] | 2020 | 83.33% |
| Pepino et al. [55] | 2021 | 84.30% |
| **Ours (with SpecAugment)** | **2022** | **84.63%** |

RAVDESS and the RML datasets respectively. Table 3.7 clearly shows that the best result was obtained by using the Time-Distributed CNN along with the Transformer.

Table 3.7: Accuracy of different architectures.

| Work | RAVDESS dataset | RML dataset |
|---|---|---|
| ViT | 62.63% | 65.43% |
| CNN+ViT | 73.51% | 76.86% |
| CNN | 73.59% | 79.30% |
| Time Distributed CNN + ViT | 82.13% | 82.41% |
| Time Distributed CNN + Transformer | 82.72% | 83.88% |

## 3.5 Analysis and discussion

With the diversity of feature extraction techniques for Speech Emotion Recognition systems, it remains a challenge to choose the best representation of the utterances. Hence, rather of relying on a single data modality, we created a model that combined data from many modalities in order to acquire additional and more comprehensive information for a more performing Speech Emotion Recognition system. Our model is based on Late Fusion (LF), a term that describes the process of merging predictions from many models to arrive at a final conclusion, thus the term "decision-level fusion".

Rather of employing many running in parallel models, we were able to utilize just one classifier with the help of the Log-Mel Spectrogram, saving both time and resources. At the beginning, a lot of CNN architectures have been considered such as Inception, ResNet, VGG-19. We first trained each of the models from scratch but the results were

Table 3.8: Impact of the data on the system's accuracy (RML dataset).

| | **CNN** | **ViT** | **Time Distributed CNN + Transformer** |
|---|---|---|---|
| **Original dataset** | 79.30% | 65.43% | 83.88% |
| **Classic augmentation techniques** | 80.21% | 65.88% | 83.94% |
| **SpecAugment** | 81.11% | 57.31% | 84.76% |

Table 3.9: Impact of the data on the system's accuracy (RAVDESS dataset).

| | **CNN** | **ViT** | **Time Distributed CNN + Transformer** |
|---|---|---|---|
| **Original dataset** | 73.59% | 62.63% | 82.72% |
| **Classic augmentation techniques** | 74.20% | 65.37% | 83.55% |
| **SpecAugment** | 74.41% | 58.12% | 84.63% |

not auspicious. This can be explained by the fact that such deep architectures require a huge amount of data. The Transfer learning technique is one of the solutions that should be considered when there is no much data to work with. It has been used in previous SER systems[13][49] but failed to attain good results, as in our work. With the rise of ViT, that outperformed the CNNs, we have considered testing them both.

CNN employs pixel arrays, but a Vision Transformer divides pictures into visual tokens, similar to how word embeddings are represented when using transformers to text. The vision transformer splits a picture into fixed-size patches, embeds each one appropriately, and uses positional embedding as an input to the transformer encoder. Because there are less inductive biases, Vision transformer requires a bigger dataset to be pre-trained on. In that case, such models surpass CNNs in terms of computing efficiency and accuracy. Although Vision Transformers are so robust for image classification, the existing datasets for Speech Emotion Recognition are relatively small and that can explain the reason why using a hybrid architecture composed of a Time Distributed CNN with a simple transformer, outperforms both the CNN and the Vision Transformer.

Although with managed to get good results with various models, the accuracy is still poor. The fact that the datasets employed for emotion recognition from speeches are modest, validates the fact that deep learning models need a large quantity of data to be adequately trained, otherwise they would not be able to achieve high levels of accuracy in their predictions.

While deep networks cannot operate effectively in the absence of adequate training

examples, it is feasible to enhance existing data to increase its effective size, which has resulted in considerable increases in the accuracy of deep networks across a wide variety of areas.

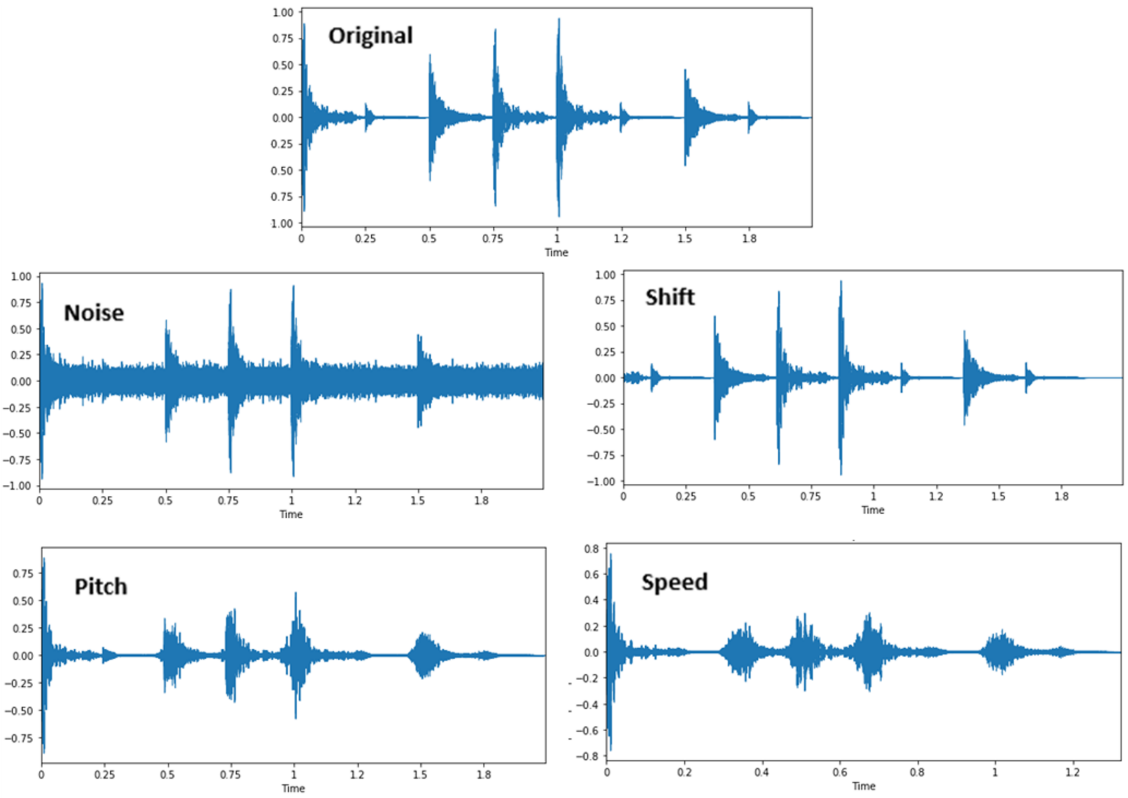


Figure 3.11: Example of some classic audio augmentation techniques

Audio Data Augmentation is the process of modifying an existing dataset in order to create a larger dataset. In reality, this results in the appearance of a bigger dataset, since several enhanced copies of a single input are presented to the model during training.. It also has the effect of making the network more robust because it is forced to learn relevant features during training. Data augmentation was used in a lot of domains such as Computer Vision application, speech recognition and even in Speech Emotion Recognition. There are lots of techniques. In our work, we have used the most known four techniques which are adding white noise with the original signal, shifting the audio signal by a constant factor to move it to the right along time axis, time stretching by changing the speed without affecting the sound's pitch and finally changing the pitch without affecting the speed.

Traditional ways of enhancing auditory input, on the other hand, incur significant computing costs and, in certain cases, need the collection of new data. A novel augmenta-

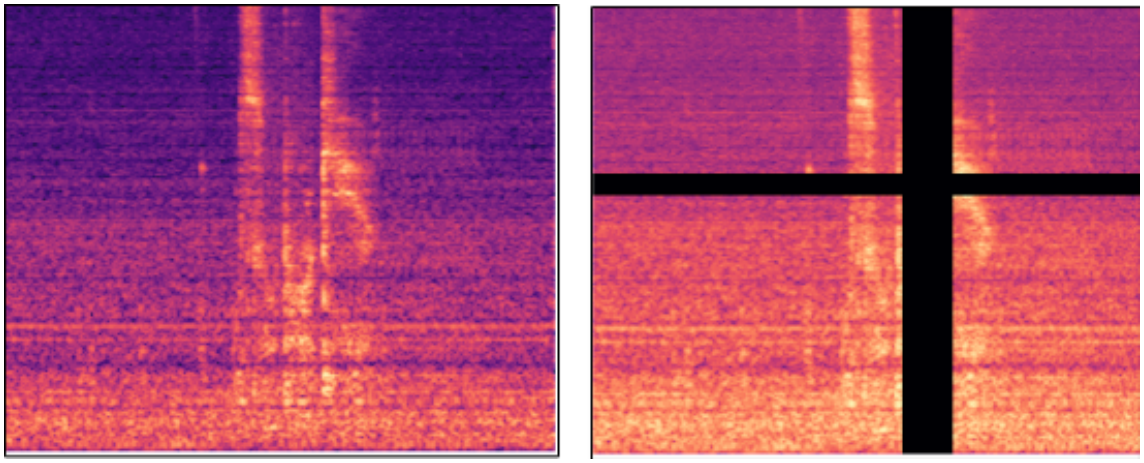

Figure 3.12: An example of SpecAugment

tion approach, SpecAugment, has been developed by D. S. Park et al. [70], which directly augments the audio spectrogram instead of augmenting the input signal as is often accomplished. This strategy is straightforward, computationally inexpensive to implement, does not need the collection of extra data, and yielded superior outcomes when compared to typical data augmentation strategies. SpecAugment modifies the spectrogram in a variety of ways, including warping it in the temporal axis, masking chunks of successive frequency channels, and warping blocks of speech in the time direction. These enhancements are intended to aid the model in remaining resilient in the presence of time-direction deformations, partial loss of frequency information, and partial loss of tiny parts of speech from the signal. Both Table 3.8 and Table 3.9 clearly demonstrate the influence of the data on the accuracy of the models. The more the amount of data we have, the higher the accuracy we get. The SpecAugment approach, however, has failed with the ViT, despite its efficacy. This failure may be explained by the fact that ViTs are vulnerable against adversarial patches.

Along with model fusions, we introduce a fresh perspective of multi-label classification. As mentioned in section 3.3.4, the softmax activation function may come in handy to determine the distribution of each emotion category's probability. Nevertheless, the main focus will be upon the category that has the highest value. With our new perspective, the system will be set to learn and recognize one or more emotions at the same time. This may be useful for many applications such as automatic analysis of behavior in conversation etc., where we don't want a machine to jump quickly to a conclusion without taking into

consideration all the possibilities.

## 3.6 Conclusion

So far, we have succeeded to improve the accuracy of the Speech Emotion Recognition systems. Unlike previous researches, we have combined the output of several models instead of combining several feature sets which helped in getting better results. We have tested and validated our work with the RML dataset since it was the only one that we have annotated manually. Also, our model was learnt to recognize multiple labels rather than one label and it achieved higher accuracy than single-label classification, which is enough reason to recommend considering multi-label data should for future work.

# Chapter 4

# MuLER: Multiplet Loss for Emotion Recognition

In the previous chapter, we have proposed the fusion of multiple models to improve the accuracy of existing SER systems. Although we have succeeded to improve the performance comparing to most of existing models, there still some needed improvements. In this Chapter, we propose a model that learns to encode speeches and map them into a 128-dimensional vectors. Its goal is to generate similar vectors for utterances having the same label. The model aims to ameliorate the accuracy by increasing intra class similarity and minimizing the inter-class similarity of the embeddings.

## 4.1 Introduction

As discussed in Chapter 2 (state of the art), all current models attempt to classify emotional speeches, that is, to assign each utterance to a certain category. Classification is a task that practically everyone performs on a daily basis. Since the day we are born and begin to learn, we begin classifying objects. Every day, we learn to tell the difference between people and animals, between cars and airplanes, and between many kinds of animals. As a result, we basically conduct classification all day long, even when we are not intending to. Classification is not just performed by humans, but also by intelligent machines (Object recognition, Facial recognition, Gender recognition, etc.). However, classification is a difficult process for humans, let alone machines. Due of their likeness to

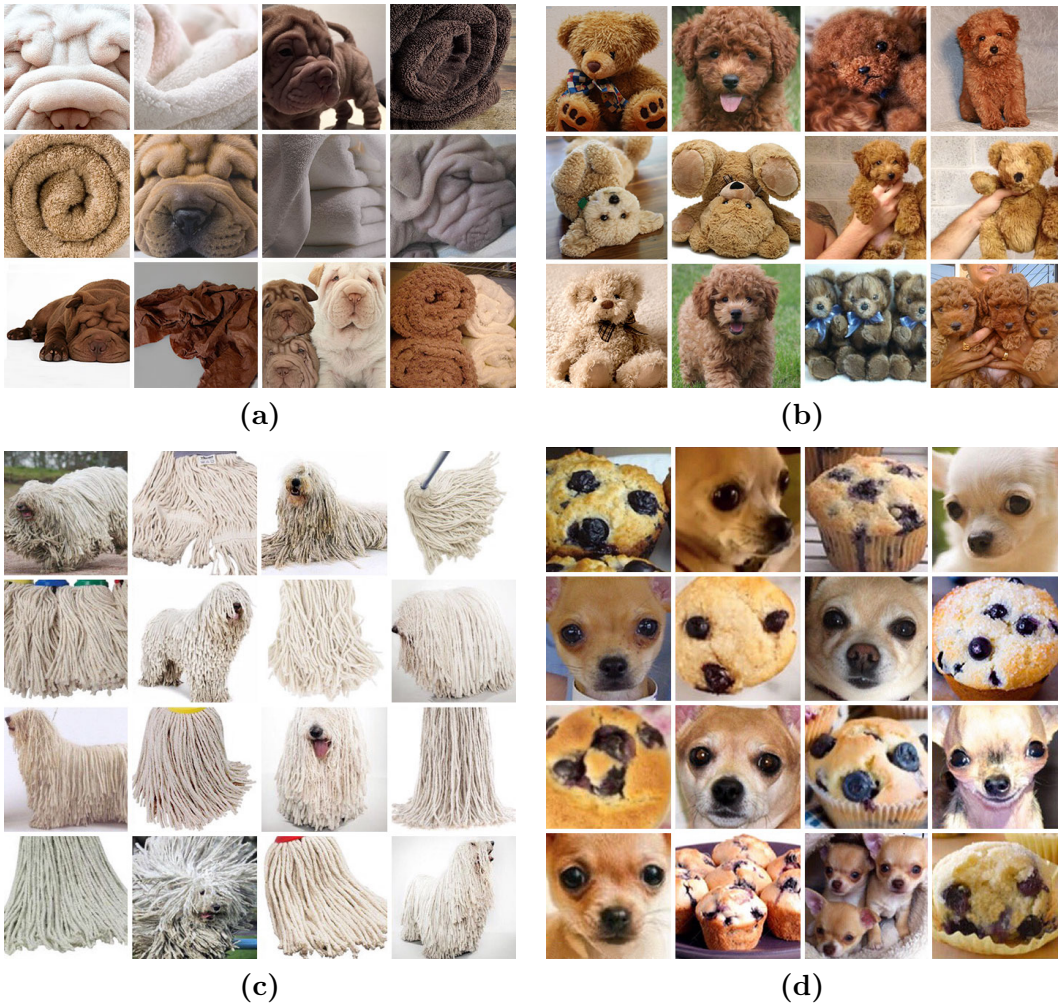


Figure 4.1: Example of confusing pictures: **(a)** Shar Pei Vs Towel **(b)** Plush Vs Puppy **(c)** Komondor Vs Mop **(d)** Dog Vs cookie

other objects, objects can occasionally be deceptive.

An adult can clearly tell the difference between a mop and a Komondor (a dog that looks like a mop), a plush and a puppy, a cookie and a dog, a towel and a Shar Pei (a dog that looks like a towel), and so on. But that's not always that simple. Figure 4.1 [1] illustrates how difficult classification may be and how much effort is needed. To verify this, we showed the images to a group of 10 adults and asked them to identify the dogs from other objects. Only seven out of ten people were able to properly identify the items on their first try, with the other three requiring a second attempt before they were successful.

Classifying emotions is also difficult since each individual has a unique manner of expressing his or her feelings. When the audio files in the RAVDESS dataset were evaluated

[1] https://imgur.com/a/K4RWn

on 247 raters (Adults), the human accuracy was about 60%, which presents a huge issue in terms of how we might construct a more exact and accurate model than humans themselves..

The idea of the proposed model consists of encoding the audio signals, instead of classifying them, i.e., utterances with the same label will have similar encodings.

## 4.2 Data encoding

Data encodings have been employed in a variety of fields, including face recognition, word embeddings, and so on. Data encodings imply that a model will be trained to map input data to a specific vector, i.e., each file will be provided with its own set of values.

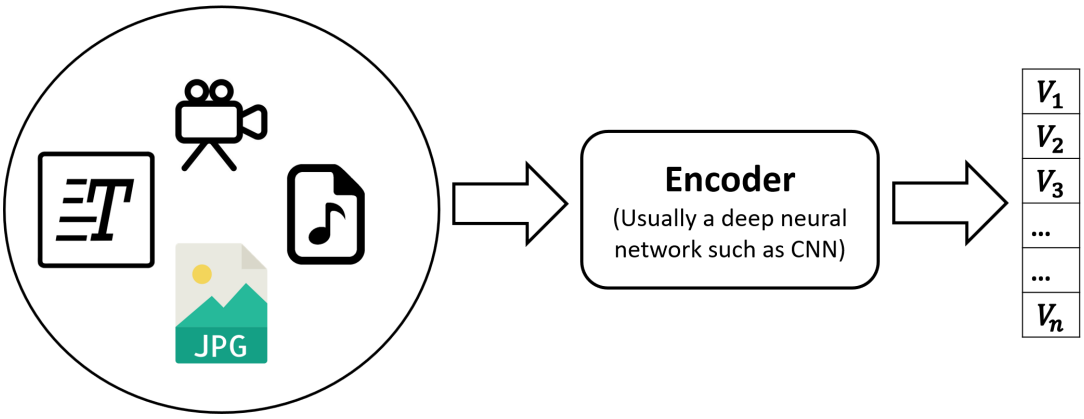


Figure 4.2: The process of data encoding

### 4.2.1 Problems with feature extraction

In the SER domain, we are facing some limitations and challenges. The first one concerns the feature extraction part since it plays a major role in every machine learning model's success. In recent years, a lot of feature extraction algorithms have been used for the purpose, so, the best feature set remains a major challenge.

We have previously discussed the work of Seyedmahdad M. et al [44] and described their four proposed model. The same models were also used on handcrafted features. All the models were also tested on IEMOCAP database. The first model has achieved 57.20% accuracy, the second model has achieved 53.00% accuracy, the third model has achieved 62.70% accuracy and the fourth model has achieved 63.50% accuracy. The best result was

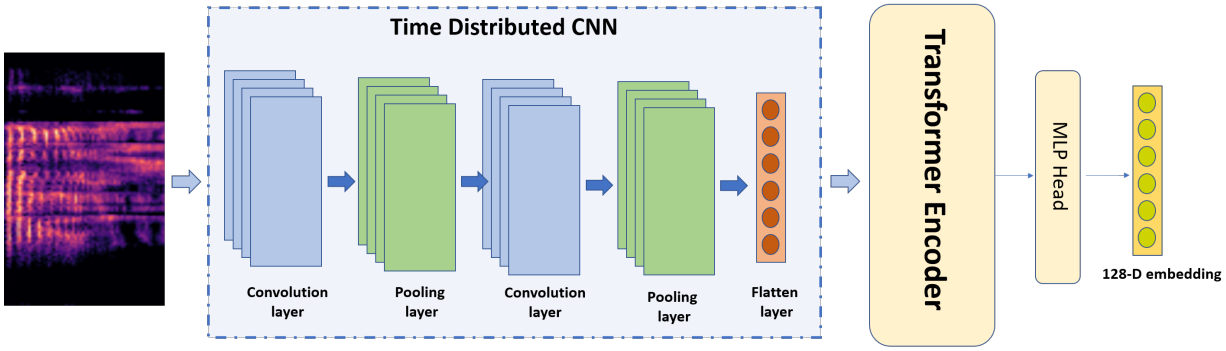


Figure 4.3: The architecture of the data encoder

obtained with the fourth model. From this paper, we can conclude that the inputs have a huge impact on the model's accuracy: the first two models performed better with the raw spectral as an input, the last two models performed better with the handcrafted features as inputs.

### 4.2.2 Encoders

The principle is simple: two different inputs should have two different vectors. Two different inputs with the same label should have similar vectors. The goal of our system is to generate encodings for utterances where two utterances that have the same label should have similar encodings. To get encodings, we first need a model (encoder) that encodes speeches.

In Chapter 3, experiments has shown that the Time Distributed CNN with the Transformer model has scored the best accuracy comparing to other proposed models. For this reason, this model will be deployed as an encoder by replacing the last layer (Activation function) with a Fully-connected of size 128 followed by $L_2$ normalization. The last layer will represent the encodings of a single utterance.

### 4.2.3 Triplet Loss

The Triplet loss [71] is a loss function that was introduced for the first time in 2015 to perform face recognition and identification. In face recognition for instance, we need to be able to compare two unknown faces and declare whether they are from the same person

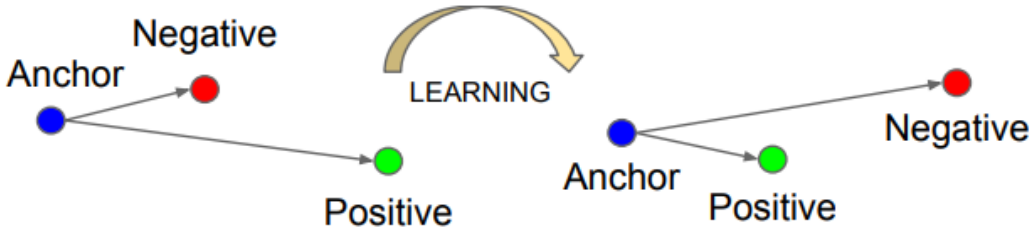


Figure 4.4: The triplet loss [71]

or not. Triplet loss in this situation is a technique to learn good embeddings for each face. Faces belonging to the same person should be clustered closely together in the embedding space. The objective of this loss is to decrease the distance between two embeddings of two files with the same label. As shown in Figure 4.4, In order to ensure that, in the embedding space, two instances with the similar labeling have their embeddings close together , and two instances with different labeling have their embeddings far apart, the triplet loss is used.

Applying this loss in our work will be done as following: First, we will construct a batch of M triplets, where one triplet represents the encodings of three different utterances: $f(x^A)$ is the encoding of an anchor file $x^A$, $f(x^P)$ is the encoding of a positive file $x^P$ where $x^P$ is different from $x^A$ but its label is the same as the anchor's label and $f(x^N)$ is the encoding of a negative file $x^N$ where $x^N$ is different from both $x^A$ and $x^P$ and its label is different from the anchor's label. Second, we will train our CNN using the following loss:

$$L = \sum_{i}^{M} [|| f(x_i^A) - f(x_i^P) ||_2^2 - || f(x_i^A) - f(x_i^N) ||_2^2 + \alpha] \tag{4.1}$$

Where $[z]_+ = max(0, z)$, $\alpha$ is a margin that is enforced between positive and negative pairs and $f(x_i^A)$ represents the encoding of the $i^{th}$ anchor file in the batch and so on. Minimizing the Loss L will induce the minimization of $|| f(x_i^A) - f(x_i^P) ||_2^2$ which is the distance between the anchor $x^A$ and the positive $x^P$ and the maximization of $|| f(x_i^A) - f(x_i^N) ||_2^2$ which is the distance between the anchor $x^A$ and the negative $x^N$. This way, two files having the same label will have similar encodings.

### 4.2.4 Quadruplet Loss

The Triplet loss is efficient, but despite its success and the good results, the authors in [72] have proven that there is still a weaker generalization capacity from train data to test data. Figure 4.5 (a) and (b) exhibit the impact of applying two models (e.g., with triplet

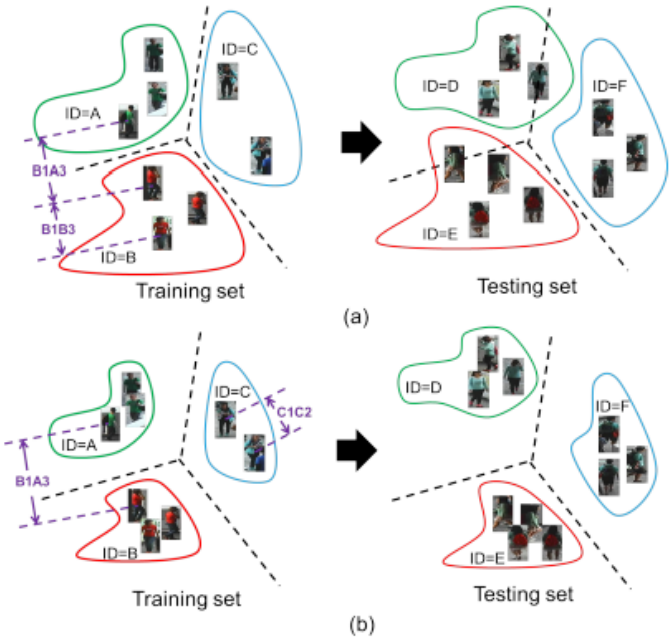


Figure 4.5: The difference between triplet loss (a) and quadruplet loss (b)[72]

loss and quadruplet loss) learnt on the same dataset. As can be seen, the quadruplet loss based model generates data with a low intra-class variance and a large inter-class variance, and so outperforms the triplet loss based model on the testing set. As a result, they developed a novel loss function they dubbed Quadruplet loss. Their technique entails dealing with quadruplets rather than triplets, which requires the addition of additional embedding $f(x^{N_2})$ of a negative audio file $x^{N_2}$, where $x^{N_2}$ is different from $x^A$, $x^P$ and $x^N$ and its label is different from the labels of $x^A$, $x^P$ and $x^N$. With the new encoding, they have added a new constraint to their loss function which became:

$$L = \sum_{i}^{M}[g(x_i^A, g(x_i^P)^2 - g(x_i^A, g(x_i^N)^2 + \alpha_1]_+ + \sum_{i}^{M}[g(x_i^A, g(x_i^P)^2 - g(x_i^N, g(x_i^{N_2})^2 + \alpha_2]_+ \quad (4.2)$$

Where g(x,y) is a learned metric that represents the distance between two images where the larger g(x,y) is, the more dissimilar x and y are. Minimizing the Loss L will

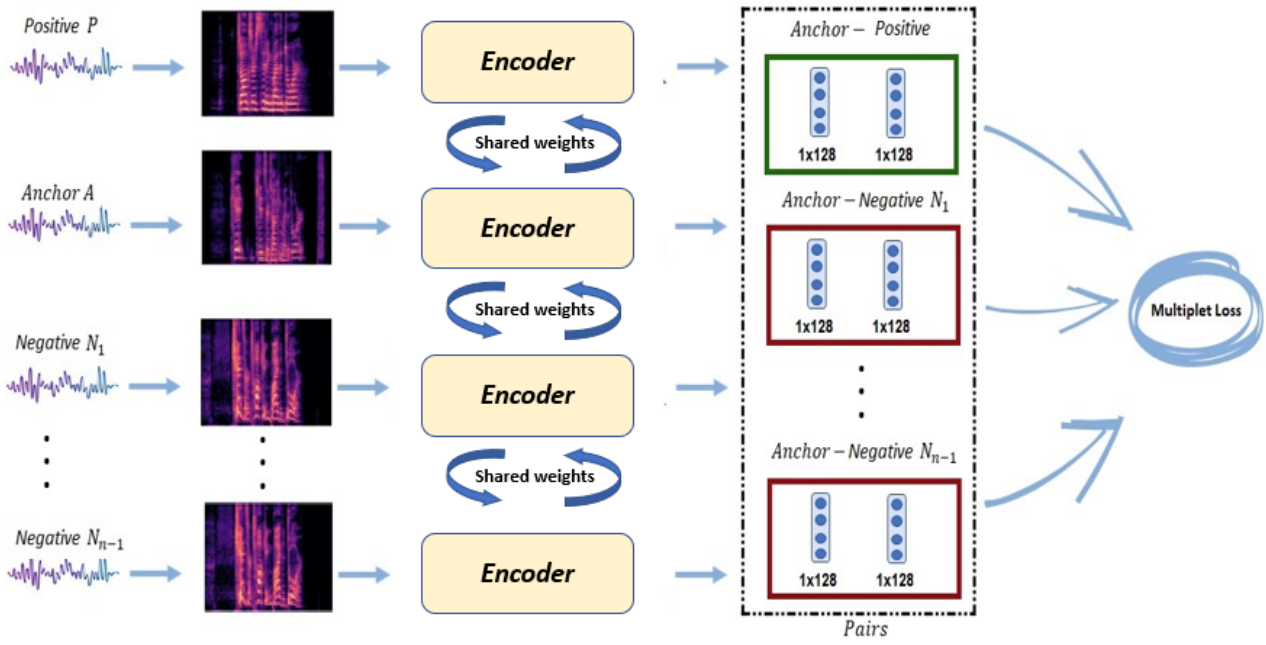


Figure 4.6: The proposed model of the second contribution

induce the minimization of the distance between positive pairs, the maximization of the distance between the positive and the negative, and also the maximization of the distance between the two negatives that have different labels.

## 4.3 Proposed model

While the purpose of the triplet loss is to withdraw a positive utterance to an anchor and to push away the negative utterance, the quadruplet loss adds a new constraint on top of the triplet loss's constraint to maximize the distance between the two negatives. The added constraint aims to maximize the intra-class distance.

### 4.3.1 Multiplet loss

Both of the losses are mainly used in the person ID domain, where for each anchor, there is a huge number of negatives. However, in the emotion recognition domain, the number of all classes varies generally between 4 and 7 and sometimes gets up to 10 (depending on the dataset). So, the number of negatives is relatively small. From there we get the idea behind our multiplet loss. Our loss function consists of pulling closer a positive utterance and pushing away all the negatives at the same time. Given a dataset of n emotions and

a batch T of size M, containing M multiplets, where a multiplet is composed of an anchor utterance, a positive utterance and n-1 negative utterances, the loss function is defined as follows:

$$L = \sum_{i=1}^{M} \sum_{j=1}^{n-1} [|| f(x_i^A) - f(x_i^P) ||_2^2 - || f(x_i^A) - f(x_i^{N_j}) ||_2^2 + \alpha_j] \tag{4.3}$$

With this new loss, for each batch, we will choose an anchor $x^A$, pulls towards it a positive utterance $x^P$ and pushes away all the negative utterances $x^{N_j}$ with $(1 < j < n-1)$, instead of pushing one random negative at a time as in the Triplet loss. The $\alpha_j$ is a margin that is enforced between positive and $j^{th}$ negative pairs.

### 4.3.2 Margin thresholds and Multiplet selection

In this work, we are facing two big challenges: a suitable samples selection for the training and the margin thresholds selection. The margins are enforced between positive and negative pairs, and what we want is to have margins that separate negatives from positive ones. So, their values have a big impact on our system's accuracy. However, an appropriate margin threshold cannot be defined beforehand, especially when we work with a big number of thresholds. What we really want is to maximize the accuracy of our model. So, if we consider $f(x)$ an objective function that takes margin thresholds and returns $-Accuracy$, then, for a set of different margins x that can take any value in the domain X, we want to find the margins $x^*$ that maximizes the accuracy:

$$x^* = \underset{x \in X}{\operatorname{argmin}} f(x) \tag{4.4}$$

The Grid search or the Random search can be deployed to determine the set $x^*$ that maximizes the accuracy. However, they are not the best choice, for two reasons: First, they will be computationally expensive and second, those searches do not pay attention to previous results and they will continue to scan through the entire continuum of estimators while the optimal solution can be located in a small region. For those particular reasons, we went for choosing optimal values for our margins using the Bayesian Optimization (BO) approach. Instead of trying the different combination on the objective function and to reduce the computational cost, the BO uses a probabilistic model of the objective function,

also known as the surrogate model. There are a lot of common choices for surrogate models [73] such as Gaussian Process Regression, Random Forest Regression and Tree-structured Parzen Estimator. Experiments showed that the Gaussian Process (GP) [73] outperforms other methods. The Gaussian process works by building a joint probability of the input margin thresholds and the accuracy of the model given that set of margin thresholds. It returns a mean and standard deviation approximation of the objective function. What we need, is to find a set of hyper-parameters that perform best on the surrogate. To do this, we need to calculate the probabilistic score of this set of hyper-parameters x and choose the one that scores the best. The Expected Improvement (EI) is the most common choice. It is expressed mathematically as:

$$EI_{y^*}(x) = \int_{-\infty}^{+\infty} max(y^* - y, 0) PM(y \mid x) dy \tag{4.5}$$

Where here $y^* = min\Big\{f(x_i), 1 \leq i \leq n\Big\}$ is the best value found so far, PM is the posterior GP and y is the actual value of the objective function. The goal here is to maximize the Expected Improvement. The margins that will maximize the surrogate will most likely return good results on the original objective function. By calculating a lot of different combinations on the surrogate model and trying only the best one on the objective function, we ensure getting the best margin thresholds while assuring the reduction of the computational cost. While working with the surrogate model, we will keep track by saving a history of (hyper-parameters, score) pairs that will be used for the update. The process will be repeated until max iterations or time is reached. The accuracy of the validation set is used to determine the best margin thresholds. The number of iterations was set to 25. For the suitable sample's selection, we chose to work with online hard negative mining since it yields the best performance [74][72]. For each batch, we compute the embeddings, then for each anchor we select the furthest positive and the closest negative from each negative class.

## 4.4 Experiments and results

At each iteration, different data combinations are used to train the encoder. The anchor, the positive and all the negatives will be passed through an encoder to get an embedding of size 128, that is to say that every audio signal will be presented as a vector of size 128. Thereafter, all the vectors will be introduced at the same time to the Multiplet loss function, which ensures that the values of the embeddings of the same label are similar.

### 4.4.1 Datasets

Our work was validated with 2 datasets: The RML and the RAVDESS datasets. Every database is divided randomly into two folds: the first contains 80-85% of the data and it will be used to train the network and the second contains 15-20% of the data and it will be used for testing the model. The splitting will be done in a way that ensures that the folds are produced with maintenance for each class the proportion of samples (Figure 4.7). For example, if a database contains 220 neutral speeches, 160 sad speeches, 195 angry speeches and 80 happy speeches then the training fold will contain 176 neutral speeches, 128 sad speeches, 156 angry speeches and 64 happy speeches whereas the testing fold will contain 44 neutral speeches, 32 sad speeches, 39 angry speeches and 16 happy speeches.

### 4.4.2 Hyper-parameters

For our model, we have used the Adam optimizer with a learning rate equals to 0.00006. A ReLU activation function was used for each layer except for the last one. We have used a dropout of 0.1 to prevent over-fitting. Each batch is composed of 200 multiplets. The number of epochs was set to 1800. As for the BO, we have used the skopt library . The number of calls was set to 35, and the bounds on each dimension are 0.001 and 1.0.

### 4.4.3 Classification

The main goal of our work is to encode speeches into vectors where speeches having the same label are encoded similarly. For this reason, a CNN was deployed to produce the embeddings. However, once the test-set is encoded, a K- Nearest Neighbors (K-NN) will be used to calculate the accuracy and to check if we succeeded in generating encodings

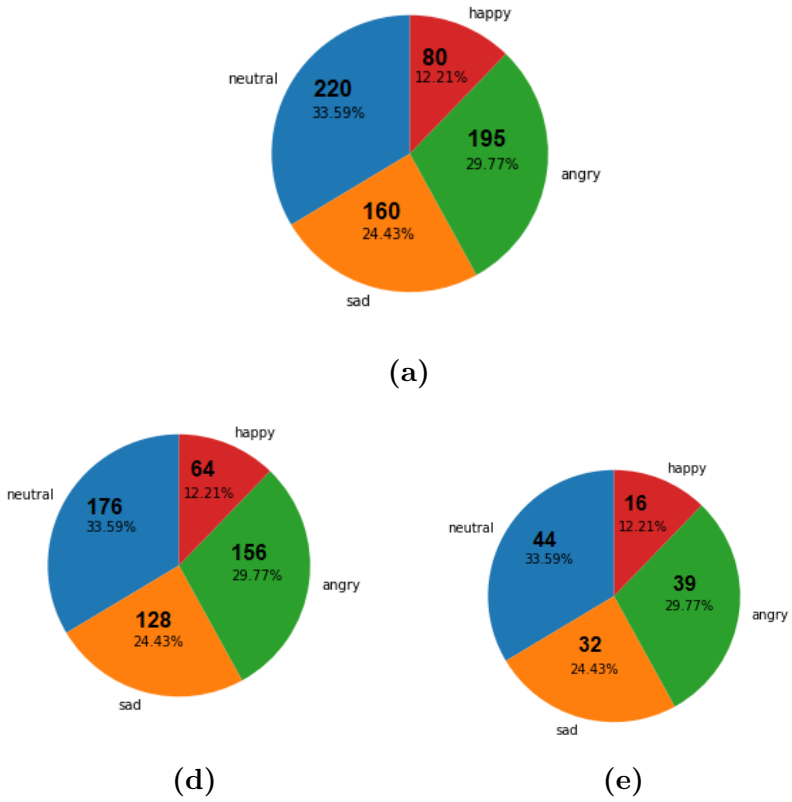


Figure 4.7: **(a)** Original dataset **(b)** Train set after split **(c)** Test set after split

that maximize the inter-class similarities and minimize intra-class similarity. The K value depends on the dataset. K was set to 11 for the RML dataset and 13 for the RAVDESS dataset.

### 4.4.4 Evaluation metrics

Along with the test accuracy, a confusion matrix is used to evaluate the efficiency. A confusion matrix is a tool for evaluating the classification efficiency of the model in relation to certain test data [75]. It is a two-dimensional matrix, with one dimension indexed by the actual label of an instance and the other dimension indexed by the label assigned by the model. For a three-class classification assignment, Table 4.1 provides an instance of confusion matrix with classes A, B and C.

From Table 4.1, we can note that class A contains in total 13 objects among which 10 are well-classified, and 3 are mislabeled, class B contains in total 7 objects among which 6 are well-classified, and 1 is mislabeled and class C contains in total 11 objects among

Table 4.1: An example of Confusion Matrix

| | Assigned class | | | |
|---|---|---|---|---|
| | | A | B | C |
| Actual class | A | 10 | 2 | 1 |
| | B | 0 | 6 | 1 |
| | C | 0 | 3 | 8 |

which 8 are well-classified, and 3 are mislabeled. For each class i, the total numbers of false negative (TFN), false positive (TFP), and true negative (TTN) will be calculated on the basis of the following equations:

$$TFN_i = \sum_{j=1, j\neq i}^{n} x_{ij} \tag{4.6}$$

$$TFP_i = \sum_{j=1, j\neq i}^{n} x_{ji} \tag{4.7}$$

$$TTN_i = \sum_{j=1, i\neq j}^{n} \sum_{k=1, k\neq i}^{n} x_{jk} \tag{4.8}$$

Where $n$ is the number of classes. The total true positive $TFN_{all}$ is obtained as follow:

$$TTP_{all} = \sum_{j=1}^{n} x_{jj} \tag{4.9}$$

We can also calculate the accuracy of each class i apart:

$$accuracy_i = \frac{TTP + TTN}{TTP + TTN + TFP + TFN} \tag{4.10}$$

And the overall accuracy (test accuracy) is calculated as follows:

$$accuracy = \frac{TTP_{all}}{TTP + TTN + TFP + TFN} \tag{4.11}$$

### 4.4.5 Results and comparisons

The datasets were split into a training set, a validation set and a testing set. To make sure that both models (the triplet loss model and the multiplet loss model) are trained with the same data, and to be able to compare them, they both were trained simultaneously

Table 4.2: Result of the encoding-based model on the RML and RAVDESS datasets

| Dataset | Work | Year | Accuracy |
|---|---|---|---|
| RML | Avots et al. [62] | 2019 | 69.30% |
| | Aouani. And Ayed [52] | 2020 | 74.07% |
| | Issa et al. [12] | 2020 | 77.00% |
| | Xia et al. [68] | 2020 | 73.15% |
| | **Ours (Triplet)** | **2021** | **88.89%** |
| | **Ours (Multiplet)** | **2021** | **91.66%** |
| RAVDESS | Hajarolasvadi and Demirel [45] | 2020 | 68.00% |
| | Mustaqeem et al. [13] | 2020 | 71.61% |
| | Muppidi and Radfar [54] | 2021 | 77.87% |
| | Mustaqeem and Kwon [53] | 2020 | 79.50% |
| | Mustaqeem and Kwon [69] | 2020 | 80.00% |
| | Seo and Kim [56] | 2020 | 83.33% |
| | Pepino et al. [55] | 2021 | 84.30% |
| | **Ours (Triplet)** | **2021** | **85.41%** |
| | **Ours (Multiplet)** | **2021** | **88.19%** |

Table 4.3: Confusion Matrix of the RAVDESS dataset

| | Angry | Happy | Disgusted | Fear | Sad | Surprised | Neutral | Calm |
|---|---|---|---|---|---|---|---|---|
| Angry | **78.95** | 5.26 | 10.53 | 0 | 0 | 5.26 | 0 | 0 |
| Happy | 5.26 | **78.95** | 0 | 10.53 | 5.26 | 0 | 0 | 0 |
| Disgusted | 0 | 0 | **94.74** | 0 | 5.26 | 0 | 0 | 0 |
| Fear | 0 | 5.26 | 0 | **89.48** | 5.26 | 0 | 0 | 0 |
| Sad | 5.26 | 0 | 5.26 | 0 | **84.22** | 0 | 0 | 5.26 |
| Surprise | 0 | 5.26 | 5.26 | 0 | 0 | **89.48** | 0 | 0 |
| Neutral | 0 | 0 | 0 | 0 | 0 | 0 | **90** | 10 |
| Calm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **100** |

using the same training data. A lot of experiments have been conducted. Table 4.2 summarizes the best accuracies obtained by the two models along with the comparison with the state of the art.

The multiplet loss improved the accuracy by around 3% on the two datasets. Table 4.3 presents the confusion matrix of the Multiplet loss using the RAVDESS dataset whereas 4.4 presents the confusion matrix of the Multiplet loss using the RML dataset.

## 4.5 Analysis and discussion

It is necessary to establish certain criteria to evaluate the Multiplet loss's performance in comparison to the Triplet loss because it is an extension of the latter..

Table 4.4: Confusion Matrix of the RML dataset

| | Angry | Happy | Disgusted | Fear | Sad | Surprised |
|---|---|---|---|---|---|---|
| Angry | **100** | 0 | 0 | 0 | 0 | 0 |
| Happy | 0 | **100** | 0 | 0 | 0 | 0 |
| Disgusted | 8.33 | 0 | **83.34** | 0 | 0 | 8.33 |
| Fear | 0 | 8.33 | 8.33 | **75** | 8.34 | 0 |
| Sad | 0 | 0 | 0 | 0 | **100** | 0 |
| Surprised | 0 | 0 | 0 | 0 | 8.33 | **91.67** |

### 4.5.1 Emotion Encoding

The triplet-loss is an effective tool that has been widely used in many fields. A speech emotion recognition model based on end-to-end triplet loss has already been proposed in the work of Kumar et al. [76], but the results are incomparable since their model was tested on all 7356 files (song + speech audio files ) from the RAVDESS dataset, whereas our model, as well as the other state-of-the-art models, were tested on only 1440 files (speech audio files) However, the triplet loss's weak generalization capacity led to the implementation of the Quadruplet-loss. For the same reason, we came up with the Multiplet loss for emotion recognition. Chen et al. [72] stated that using a learned metric improves the accuracy. However, using the encodings and the Euclidean distance in our work gave a better result. We managed to get 91.66% accuracy with the RML dataset and 88.19% accuracy with the RAVDESS dataset. The difference can be interpreted with hardness of the dataset. As mentioned earlier, the human accuracy with the RAVDESS dataset is only 60%.

### 4.5.2 Intra-class distance vs inter-class distance

An intra class distance reflects the distribution in space of all samples that belong to the same class. For better separability, it must be small. The intra-class distance of a model is the maximum value among the intra-class distance of each class of the n classes (n is the number of emotions).

$$intra - class\ distance = \max_{1 \leq k \leq n} \left\{ \max_{x,y \in C_k} d(x,y) \right\} \tag{4.12}$$

Table 4.5: inter-class and intra-class distances

| | Distances | |
|---|---|---|
| | Inter-class | Intra-class |
| RAVDESS | 0.094 | 1.5508 |
| | 0.129 | 1.19 |
| RML | 0.997 | 0.099 |
| | 1.483 | 0.065 |

Where $C_k$ represents the $k^{th}$ class and d(x,y) represents the Euclidean distance. An inter-class distance represents the difference between two classes. For better separability, it must be big. The inter-class distance of a model is the minimum value among the inter-class distance of each pair of classes.

$$inter - class\ distance = \min_{1 \leq i,j \leq n; i \neq j} \left\{ \min_{x \in C_i, y \in C_j} d(x,y) \right\} \tag{4.13}$$

So, based on the previous definitions, a good model is the model that has the smallest intra-class distance value and the biggest inter-class distance value.

Table 4.5 shows that multiplet loss outperforms the triplet loss in terms of inter/intra class distances.

### 4.5.3 Comparison between the Triplet and Multiplet losses

To illustrate the distinction between the two losses, we will run both algorithms concurrently using the colored shapes in Figure 4.8, each of which symbolizes a different emotion.

The steps are the following:

- Step 1: Choose an Anchor point (circled in black).
- Step 2: Select the farthest positive (circled in green) and the closest negative (s) (circled in red).
- Step 3: Pull the positive towards the Anchor and push away the negative (s)

After 4 iterations (Figure 4.9 – Figure 4.16), we can clearly see that the Multiplet loss outperformed the Triplet loss in term of accuracy (clustering) and execution time.

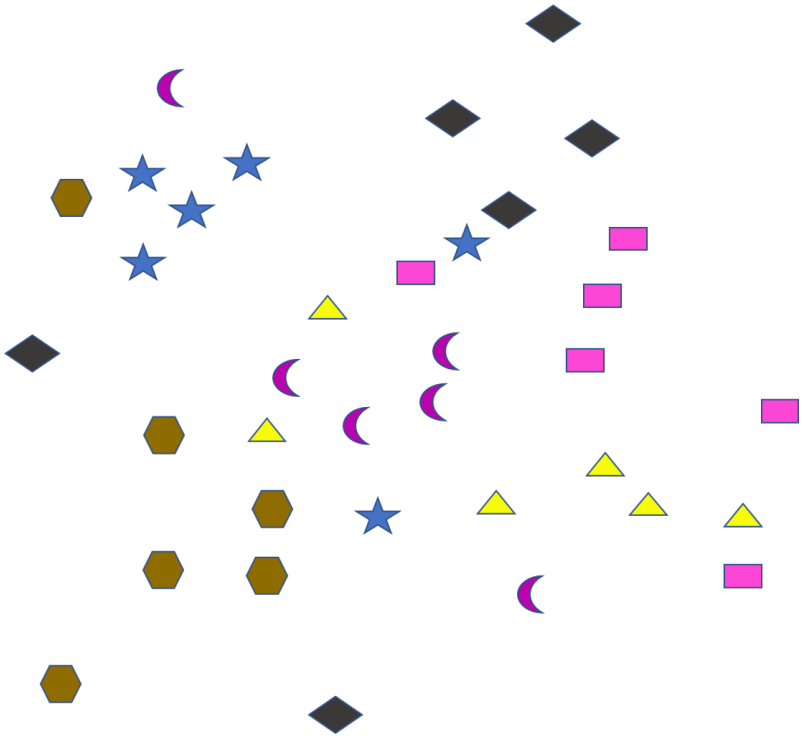

Figure 4.8: A plot of data points where each shape represents an emotion

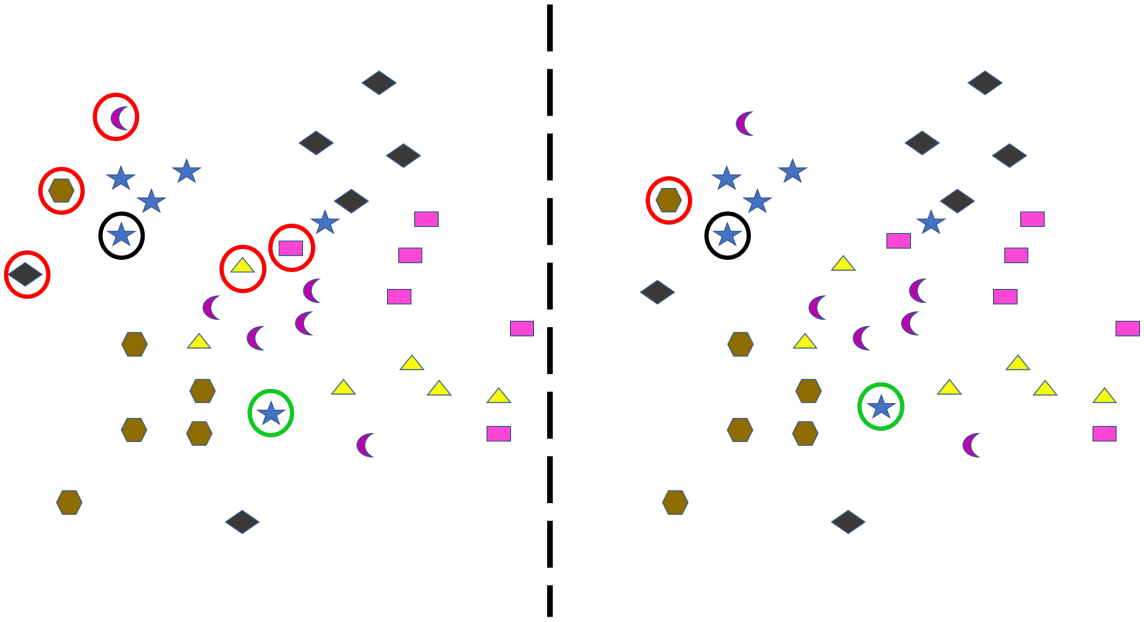

Figure 4.9: Iteration 1; Steps 1+2 (Multiplet loss on the left and Triplet loss on the right)

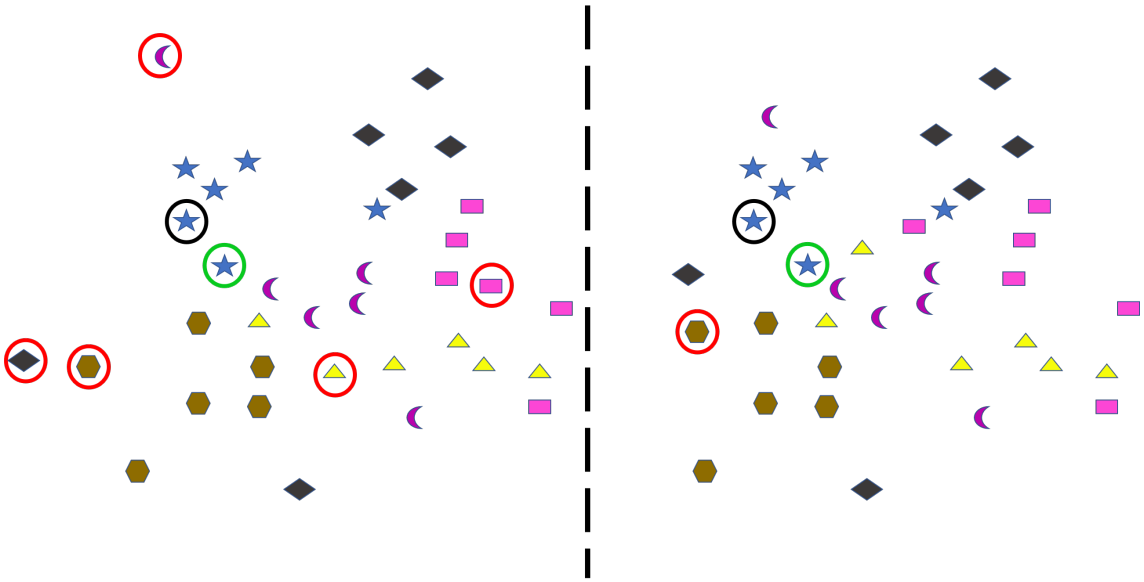

Figure 4.10: Iteration 1; Step 3 (Multiplet loss on the left and Triplet loss on the right)

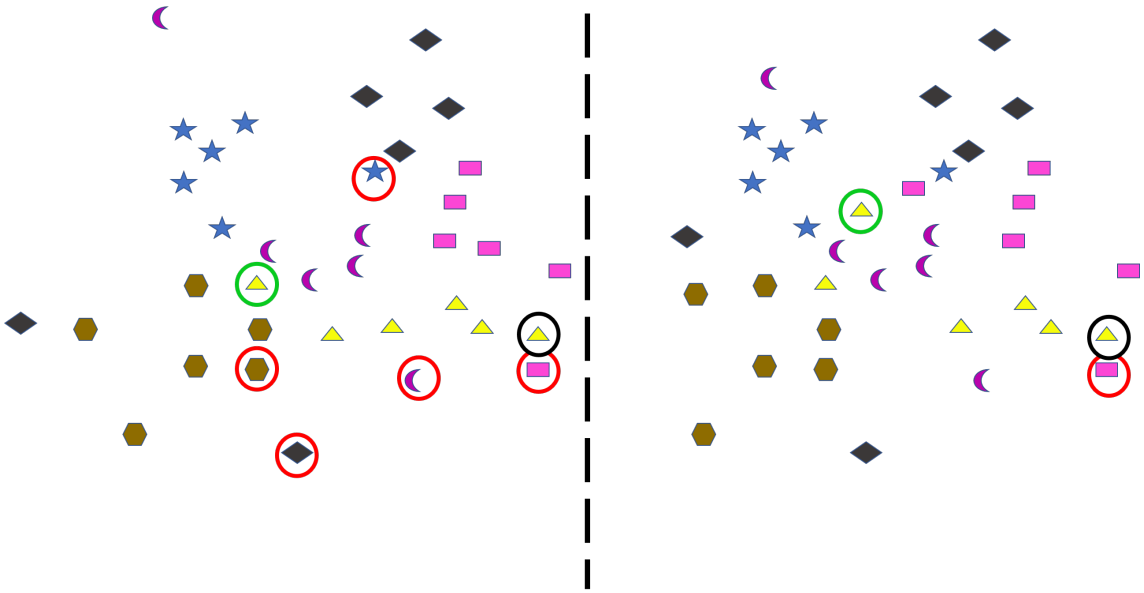

Figure 4.11: Iteration 2; Steps 1+2 (Multiplet loss on the left and Triplet loss on the right)

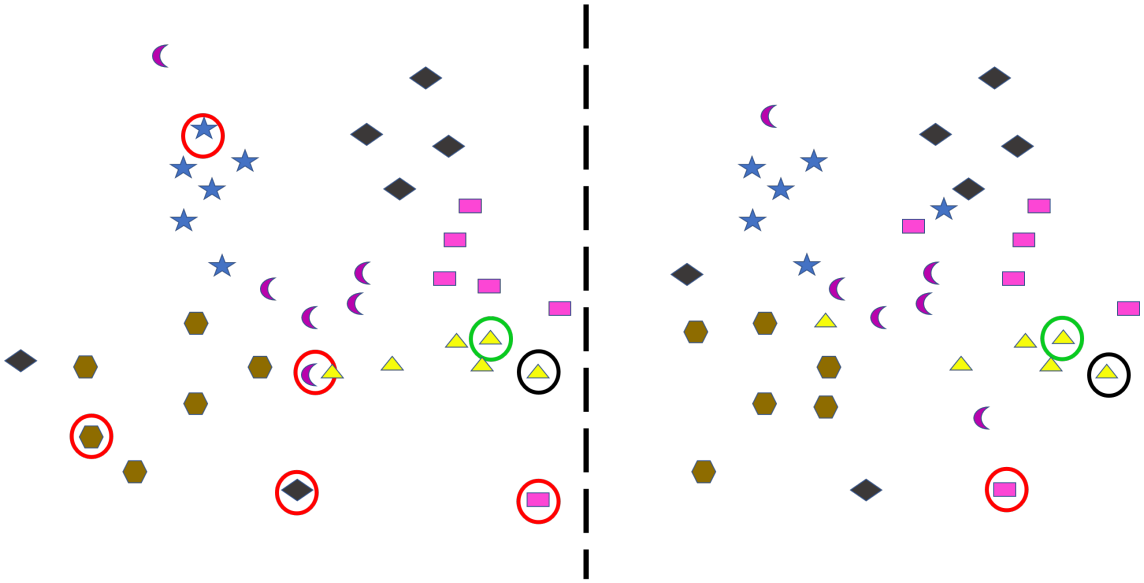

Figure 4.12: Iteration 2; Step 3 (Multiplet loss on the left and Triplet loss on the right)

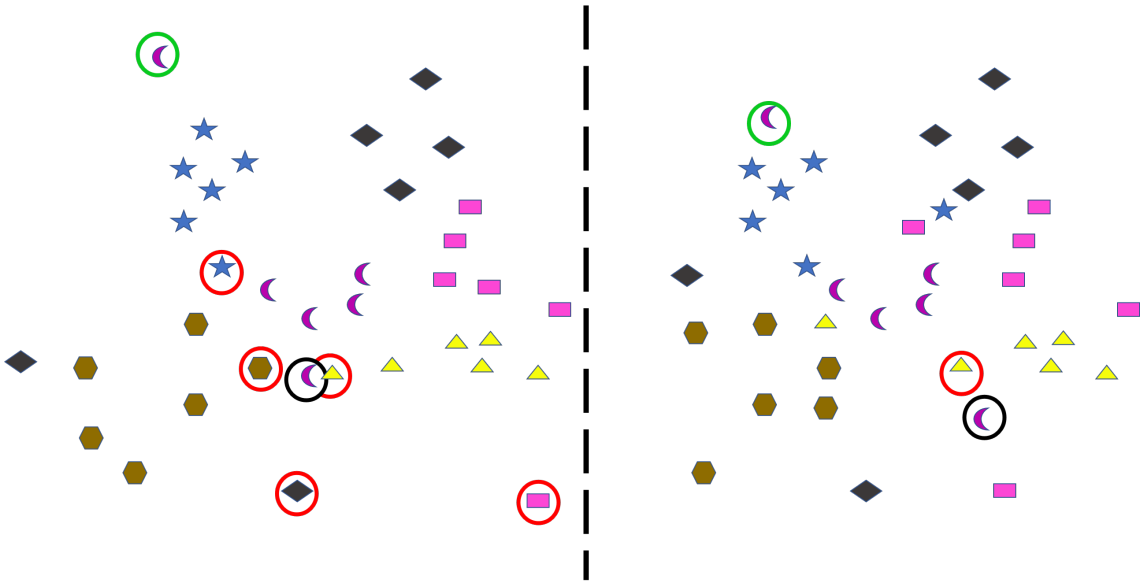

Figure 4.13: Iteration 3; Steps 1+2 (Multiplet loss on the left and Triplet loss on the right)

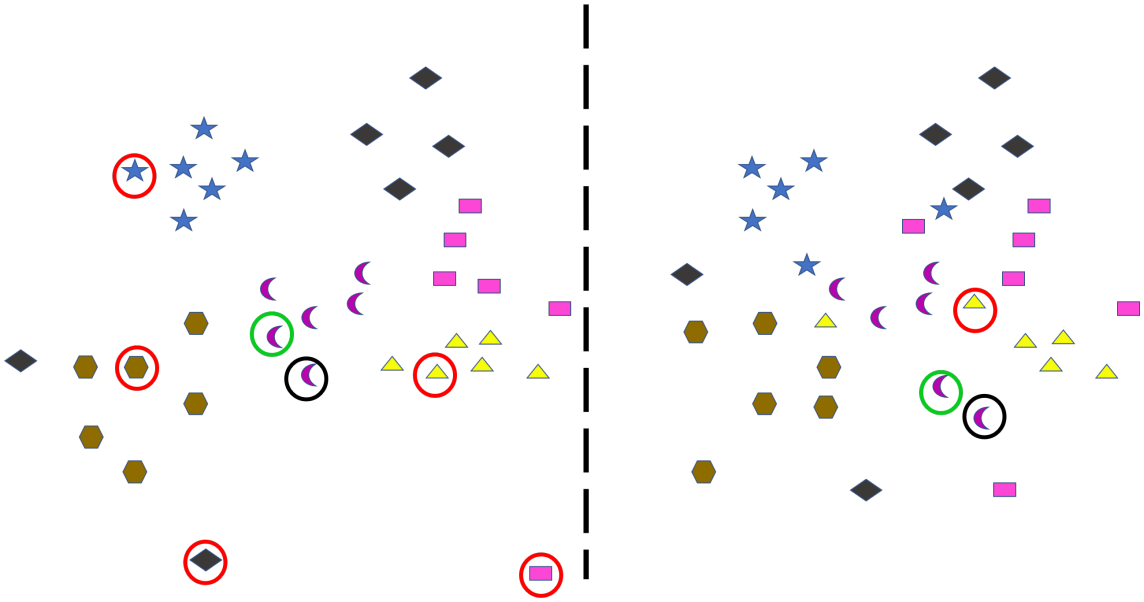

Figure 4.14: Iteration 3; Step 3 (Multiplet loss on the left and Triplet loss on the right)

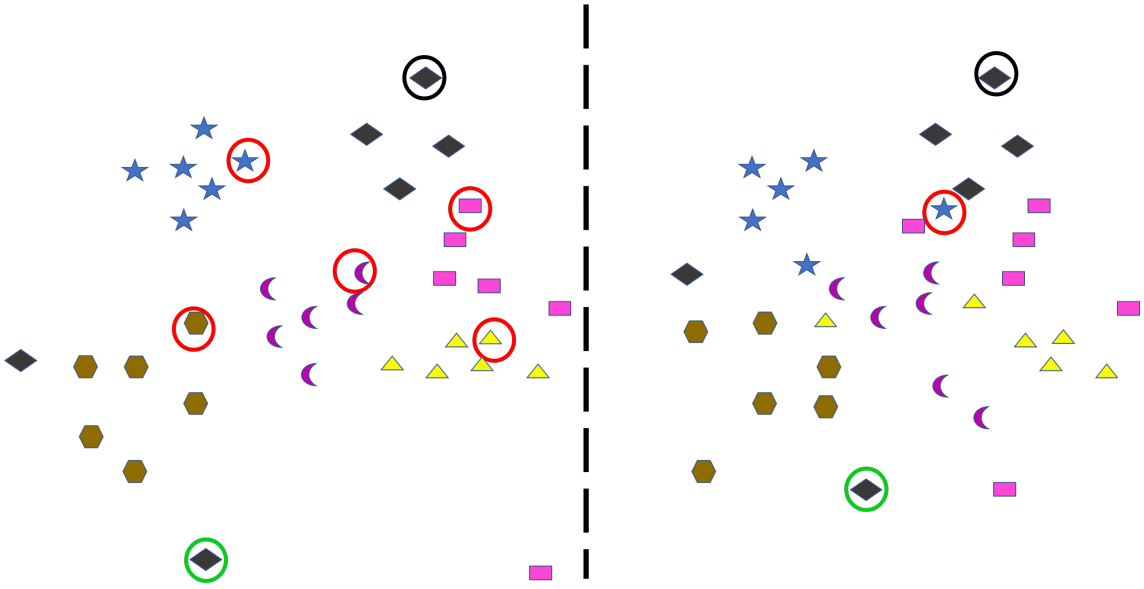

Figure 4.15: Iteration 4; Steps 1+2 (Multiplet loss on the left and Triplet loss on the right)

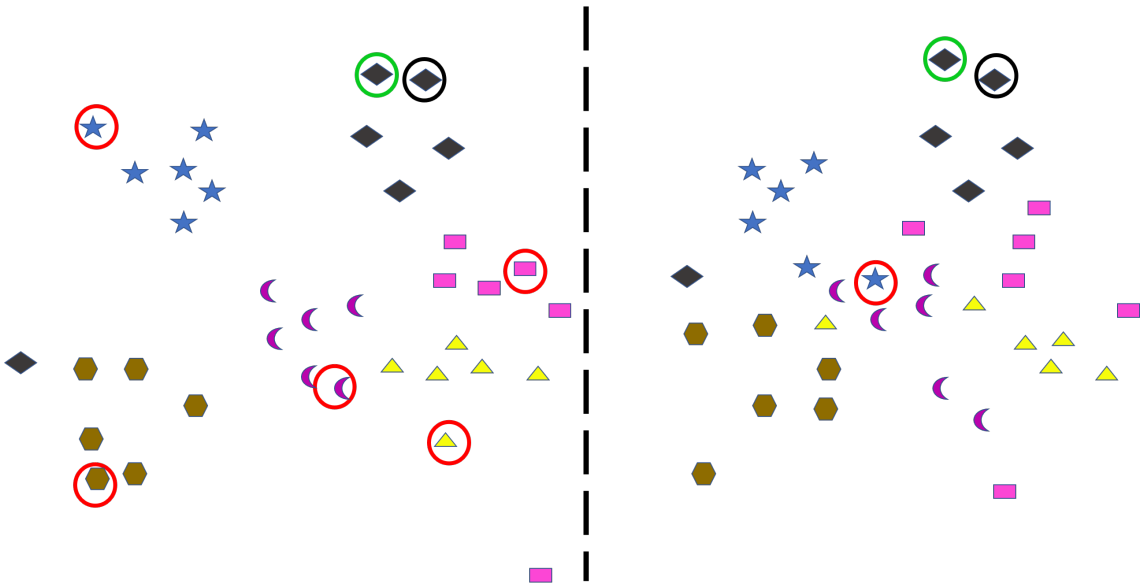

Figure 4.16: Iteration 4; Step 3 (Multiplet loss on the left and Triplet loss on the right)

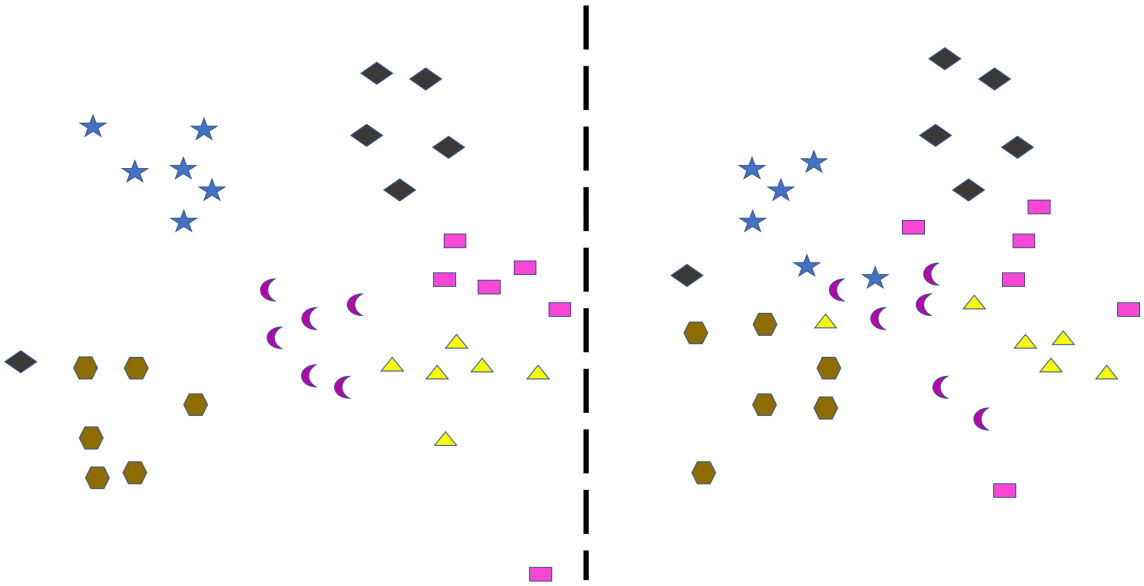

Figure 4.17: Final result after 4 iterations (Multiplet loss on the left and Triplet loss on the right)

### 4.5.4 t-SNE

When using the t-SNE technique, a probability distribution is generated that depicts the similarities between neighbors in both a high-dimensional space and a lower-dimensional space. We will attempt to convert the distances into probabilities by looking for similarities. It is divided into 3 steps:

- Step 1 : In high dimensions, we calculate the similarity between the points in the initial space. A Gaussian distribution is centered on each point $x_i$. The density under this previously established Gaussian distribution is then measured for each point $x_j$ (i distinct from j). Finally, we do a point-by-point normalization. We thus obtain a list of conditional probabilities noted:

$$p_{ij} = \frac{exp(-(\|x_i - x_j\|)^2/2\sigma^2)}{\sum_{k \neq l} exp(-(\|x_k - x_l\|)^2/2\sigma^2)} \tag{4.14}$$

  The standard deviation is defined according to a value called perplexity which corresponds to the number of neighbors around each point. This value is fixed by the user in advance and makes it possible to estimate the standard deviation of the Gaussian distributions defined for each point $x_i$. The greater the perplexity, the greater the variance.

- Step 2: We need to create a lower dimensional space in which we will represent our data. Obviously at the beginning we do not know the ideal coordinates on this space. We are therefore going to randomly distribute the points over this new space. The rest is quite similar to step 1, we calculate the similarities of the points in the newly created space, but using a t-Student distribution [2] and not Gaussian. In the same way we obtain a list of probabilities denoted:

$$q_{ij} = \frac{(1 + (\|y_i - y_j\|)^2)^{-1}}{\sum_{k \neq l}(1 + (\|y_k - y_l\|)^2)^{-1}} \tag{4.15}$$

- Step 3: We would like that the similarity measures in the two spaces coincide in order to represent points in the lower dimensional space. As a result, we need to use

[2] https://en.wikipedia.org/wiki/Student's t-distribution

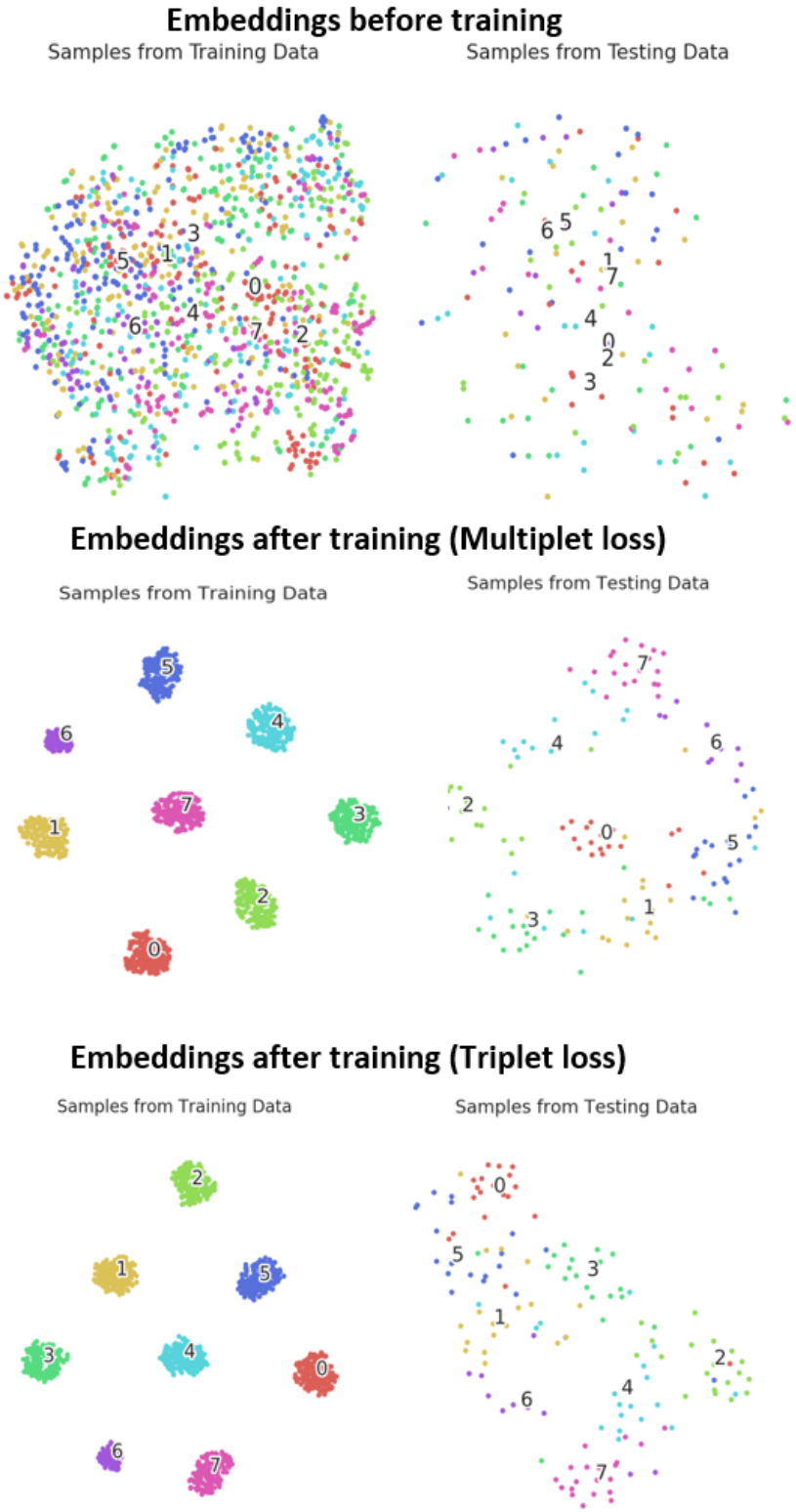


Figure 4.18: t-SNE of the Embeddings of the RAVDESS dataset before and after training

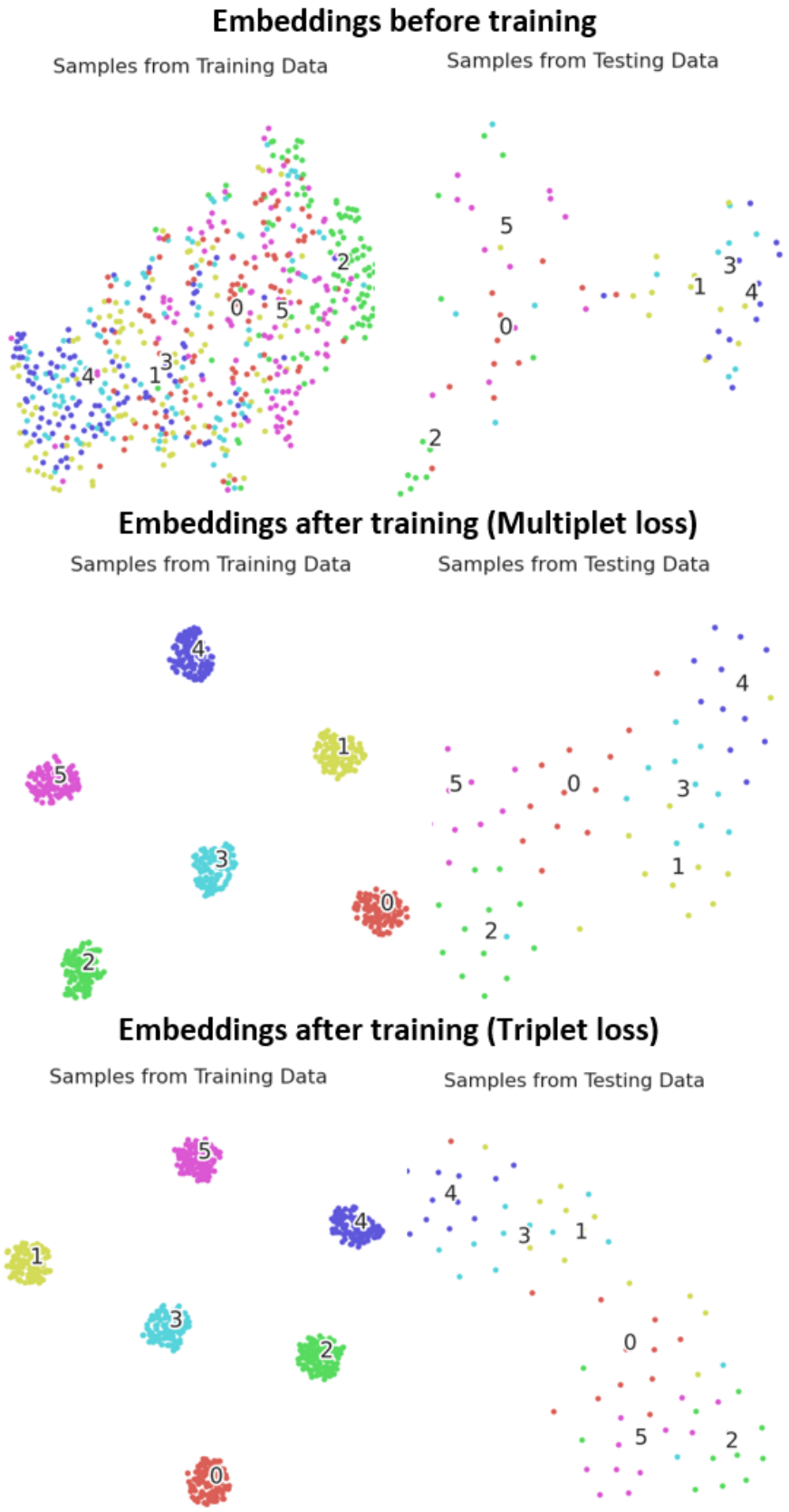


Figure 4.19: t-SNE of the Embeddings of the RML dataset before and after training

the Kullback Leibler (KL) measure [3] to compare the similarity of points in the two spaces. Then, using a descent gradient, we strive to reduce it in order to attain the best possible $y_i$ in the low-dimensional space. This is equivalent to minimizing the difference between the probability distributions in the original and lower-dimensional spaces..

Figure 4.19 presents the t-distributed stochastic neighbor embedding (t-SNE) the plot of sample of embeddings from the RML dataset before and after training the model. Class 0 refers to Happiness, Class 1 refers to Disgust, Class 2 refers to Anger, Class 3 refers to Fear, Class 4 refers to Surprise and Class 5 refers to Sadness. The same sample was used for both losses and the plots show that the new Multiplet loss outperforms the Triplet loss. Only points from two classes (1 and 3) were miss-encoded with the Multiplet loss whereas points from four classes (0, 1, 2 and 3) were miss-encoded. Figure 4.18 presents the t-distributed stochastic neighbor embedding (t-SNE) the plot of sample of embeddings from the RAVDESS dataset before and after training the model. Class 0 refers to Happiness, Class 1 refers to Disgust, Class 2 refers to Anger, Class 3 refers to Fear, Class 4 refers to Surprise, Class 5 refers to Sadness, Class 6 refers to neutral and Class 7 refers to calm. It is clear that the multiplet loss separates the different embeddings from each other better than the triplet loss.

## 4.6 Conclusion

In contrast to the other contributions, we have adopted a novel vision in this chapter that entails encoding an audio file rather than carrying out classification tasks. Our encoder was first trained using the triplet loss, after which we presented a new loss function, the multiplet loss, which is an extension of the triplet loss. In terms of accuracy, inter-class similarity, and intra-class similarity, the multiplet loss has outperformed the triplet loss.

Nevertheless, despite their outstanding outcomes, both still have certain shortcomings that will be thoroughly covered in the last chapter.

[3] https://en.wikipedia.org/wiki/Kullback-Leibler_divergence

# Chapter 5

# Detection of Emotion Categories' Change in Speeches

In the past few years, a lot of research has been conducted to predict emotions from speech. The majority of the studies aim to recognize emotions from pre-segmented data with one global label (category). Despite the fact that emotional states are constantly changing and evolving across time, the emotion change has gotten less attention. Mainly, the exiting studies focus either on prediction arousal-valence values or on detecting the instant of the emotion change. This chapter introduces a new model to detect emotion categories change.

## 5.1 Introduction

In conversations, emotions add significance to the speech and help us understand each other. Human emotions have a fundamental part in all social phenomena and some decisions can be made based on the expressed feelings, so they should be explored in depth. Within this context, allowing machines to understand emotions would produce significant improvement in the human-computer interactions in a way that the context and the circumstances of a given conversation would be easily identified and become crystal clear to machines. Emotions are dynamic in nature and they constantly change throughout time [60], hence, an intelligent system should be able to identify changes in emotions as they occur when speakers participate in human-computer interaction during which their

emotions are identified based on behavioral cues, so that it may react accordingly. Most of the conducted studies have been focusing on pre-segmented speech utterances, where each utterance has one global label (emotion). Such models are not efficient for emotion detection change since the recognition of emotions using pre-segmented speech utterances leads to a loss of continuity between feelings and does not give insights into emotion changes [59]. However, despite its importance, research on emotion change detection has gotten less attention than other research aimed at recognizing and predicting emotions from speeches. It is an interesting research area that only few papers have attempted to address. Existing researches have mainly focused on either detecting the instant of emotion change i.e., detecting when exactly an emotion change has occurred or on predicting the change of valence (positive or negative) and the arousal (low or high). To the best of our knowledge, this is the first work that introduces emotion categories change detection system i.e., detecting if a change has occurred from one category (angry, sad, neutral, etc) to another. In other words, within the same conversation or let's say within the same part of speech of a given person, if s/he was talking with a particular emotion then suddenly a change took place in his/her tone, the system would be able to detect such change. We aim to design a system that can interpret emotional states in speeches and/or conversations and detect every emotional change either from one person's long speech or the change that occurs when two or more different people have a conversation. Our proposed model is based on the Connectionist Temporal Classification (CTC) loss. It takes a long sequence of data as an input, processes it through a Convolutional Neural Network (CNN) followed by a Recurrent Neural Network (RNN) to detect pertinent features and feeds it to the CTC which will in return determine the sequence of emotion categories presented in the input speech. To evaluate our model's performance, we have introduced two new evaluation metrics: the ECER (Emotion Change Error Rate) and the ECD (Emotion Change Detection).

## 5.2 Proposed model

Our primary objective is to anticipate a series of emotions based on a particular input. In an idealistic situation, we would have a labeled dataset of conversations in which each

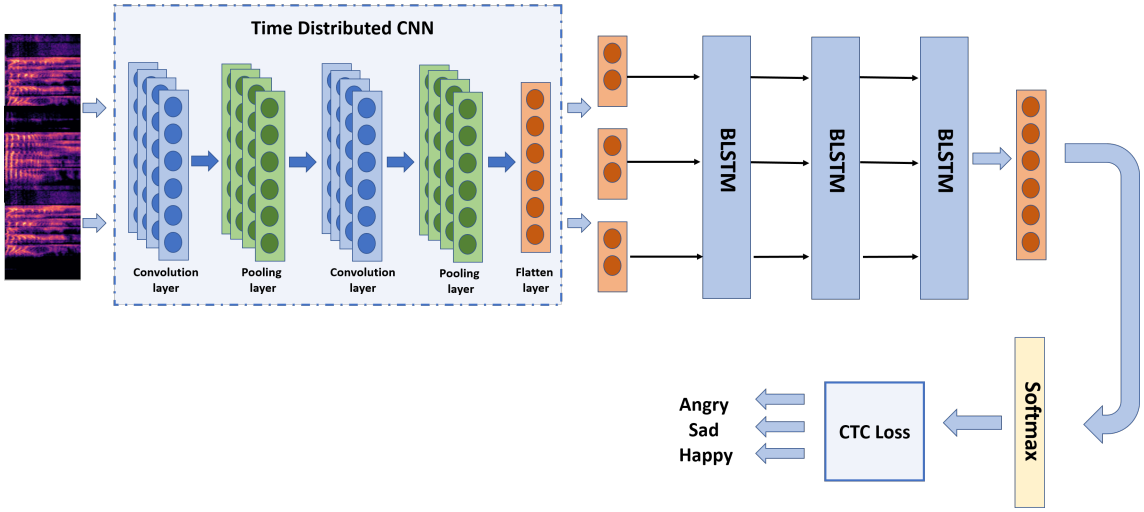


Figure 5.1: Proposed model for emotion change detection

label indicates the start, end, and emotion of each speaker; however, this is not the case with current datasets, and therefore we must improvise.

### 5.2.1 CNN-LSTM

As previously stated, since spectrograms are two-dimensional plots, it is more appropriate to employ a CNN or a ViT as a classification model, as they are primarily built for image recognition tasks. While the audio signals will be converted to pictures, the CNN alone is insufficient when dealing with sequential data. As a result, we considered implementing a CNN-BLSTM architecture.

The CNN layers were used to extract a sequence of features and RNN layers were used to propagate information through this sequence. Yet, the CNN models are commonly known for receiving and processing only one image at a time. That will be ideal if every input corresponds to one label (emotion) but in our case, every input is aligned with one or more successive emotions. What we need is to determine the sequence of emotions so it is required to repeat several emotion-detection tasks. We can think about cutting up a sequence of data into several frames and determine the emotion category on each single frame. A solution to our problem is to use the Time Distributed Layers . So, the problem is that we have several audio frames that are chronologically ordered and we need to be able to inject a sequence as input, and to make predictions of what that sequence is showing. For this model, the expected result is not a single label but rather a series of

labels to present the emotion changing along the whole utterance. First, we need to find features to recognize emotion within one frame, and then use these features in BLSTM to try to detect the categories change. So, we need to repeat several emotion detection and only after that we check all possible present categories.

Now the input will be supplied in the form of numerous spectrograms, each of which represents a single frame and should be fed into the model. If, on the other hand, we consider a typical Sequential neural network, each input is linked to the whole neuron list in the first layer. This is acceptable when working with a single image, but when working with many images, we must avoid merging them together, since the whole pixel list of all images will be transmitted to the first layer. That is, if we provide many images in the first layer, the images will be combined. And this is precisely what we do not need. If we take four spectrograms concurrently, the input form should be (4,X,Y,Z), where both X and Y denote the size of each spectrogram and Z denotes the channel number. A two-dimensional CNN cannot process such an input shape, and since the network must be divided into separate filter lists for each picture input, we must feed one image to one "convolution block", the second to another, and so forth (Figure 5.2).

We want to ensure that the full list of convolution flows can locate the same features and, eventually, that the model can be extended with more common layers. Thus, as described before in Chapter 3, the time-distributed layer suffices, since it can perform the same transformation on a list of input data. Obviously, each layer that is Time Distributed will have the same weights as the other layers in the stack. If we inject 4 spectrograms, the weights are not modified 4 times, but only once, and then distributed to all of the blocks specified in the current Time Distributed layer. That saves us valuable calculating time.

We are close to completing the final functional model at this point. The last element is to make sure that spectrograms are processed using the time concept. To put it another way, we want to process the frames in chronological sequence. The recurrent neural networks are a particularly effective kind of neural computing for this purpose. Here the RNN is utilized to ensure that the frames are processed with a concept of time in mind. As mentioned in the Chapter 1, the RNNs have a vanishing gradient issue where information, in long sequences, is lost as the gradient decreases. The best alternative is

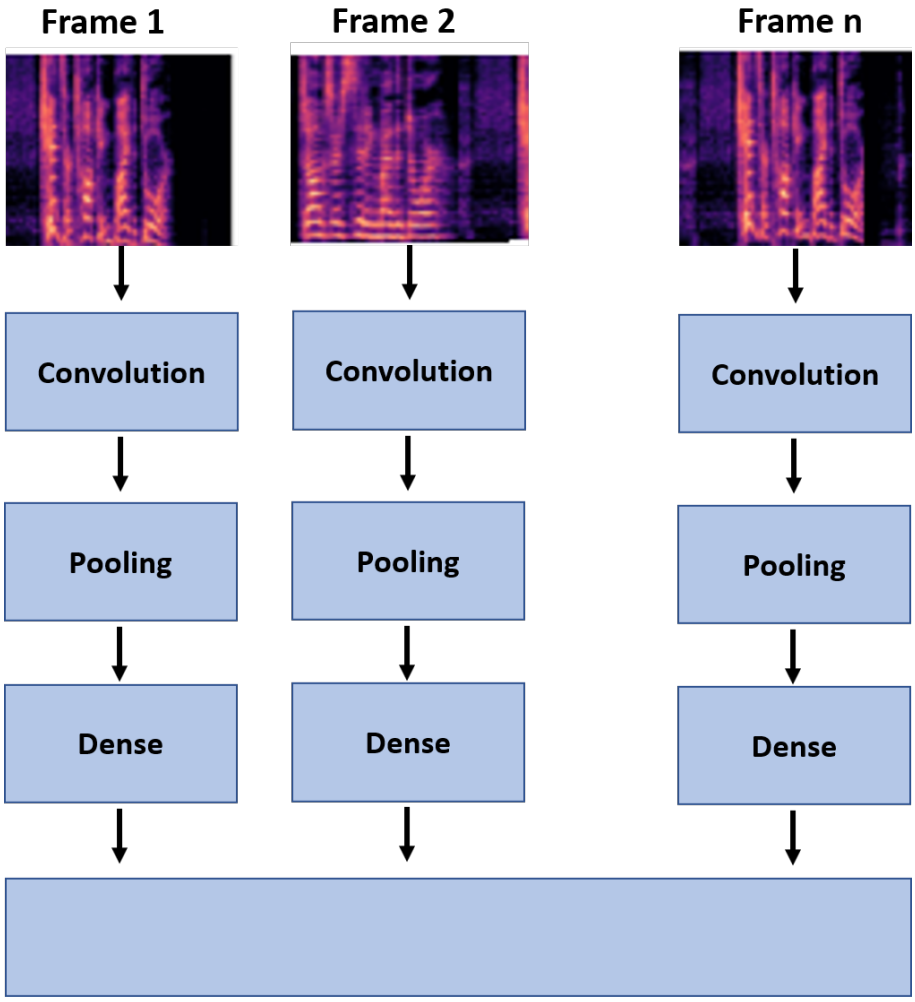


Figure 5.2: Parallel convolution flows

using the BiLSTM.

The last layer of the CNN-BLSTM architecture will be passed to a fully connected layer with a softmax function as an activation function. Usually, the softmax layer contains n units where n represents the number of labels in the dataset. In our work, the softmax will contain n+1 units where the additional unit represents the blank label (the separation between two emotions). Its units reflect the likelihood that a given label will be present at a given time step. In the following sections, we will explain in depth the reason behind adding an extra unit.

### 5.2.2 CTC (Connectionist Temporal Classification)

Consider the following scenario: we have a collection of merged audio clips (conversations) and their associated labels. Regrettably, we have no way of knowing how the emotion categories in the labels correspond to the audio. This complicates the task of training a speech emotion recognizer. Without this alignment, we are unable to use basic techniques. We may establish a rule in which every ten seconds corresponds to one of the emotion categories. However, since people's speaking speeds differ, this sort of rule may always be violated. We cannot guarantee that each individual will talk for no more than ten seconds during a conversation. Another option is to manually correlate each emotion category with its corresponding location in the audio. This works effectively from a modeling perspective if the ground truth of every time step is known. However, this is excessively time expensive for any decently significant dataset.

Connectionist Temporal Classification (CTC) is a technique for overcoming the problem of of unknown input-output alignment. It is particularly well-suited for applications such as voice recognition and detection of category changes. To be more precise, consider mapping input sequences $X = [X_1, X_2, X_3, ..., X_T]$, for example, audio, to matching output sequences $Y = [Y_1, Y_2, Y_3, ..., Y_M]$, for example, emotion categories. We're looking for an exact mapping from X to Y. There are obstacles that prevent us from using simpler supervised learning techniques, such as:

- The lengths of X and Y may vary.
- The ratio of X and Y's lengths might change.
- We lack an exact alignment of X and Y (correspondence of the elements).

The CTC algorithm overcomes these obstacles. It gives a distribution of output values over all possible Y values for a given X. This distribution may be used to infer a likely result or perhaps to calculate the likelihood of obtaining a certain result. Not all approaches for estimating the loss function and doing inference are traceable. CTC will be required to carry out each of these tasks efficiently.

We want to train our model in such a way that it maximizes the probability it assigns to the right solution for a given input. This requires an efficient calculation of the conditional

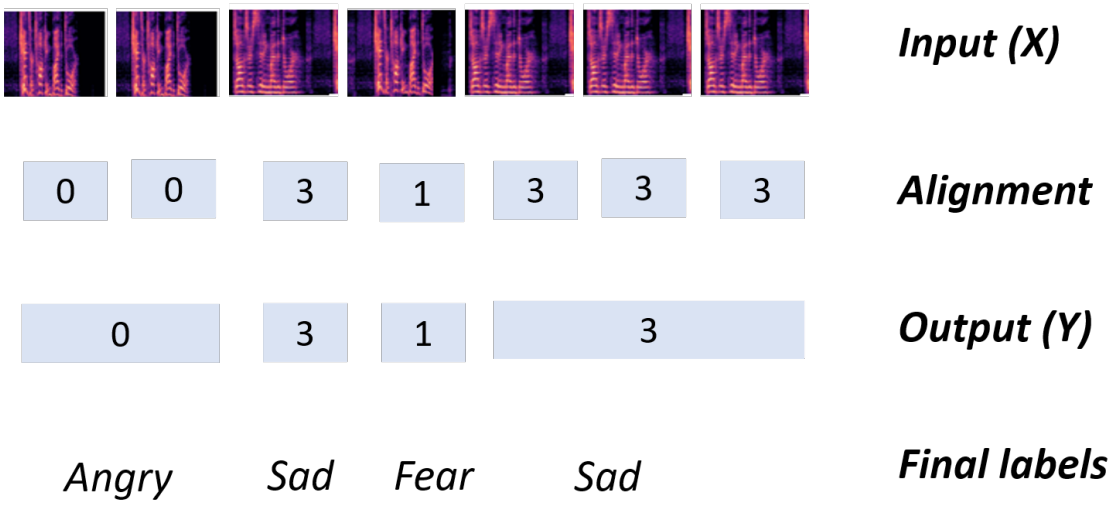


Figure 5.3: The steps of the CTC

probability $p(Y \mid X)$ , which should be differentiable so that gradient descent may be used. Following that, we'd want to utilize the model to infer a probable Y given an X. With CTC, we shall settle for an affordable option.

Given an X, the CTC method may assign a probability to every Y. Calculating this likelihood requires a thorough understanding of how CTC views input-output alignments. The CTC algorithm is alignment-free, which means it does not require input and output alignment. CTC, on the other hand, operates by summing the likelihood of all potential alignments between the two to yield the chance of an output given an input. To understand how the loss function is finally determined, we must first grasp what these alignments are. Consider a basic way to motivating the exact shape of the CTC alignments. Let's look at an example. Assume the input is six frames long (six spectrograms) and Y = [Angry, Sad, Angry]. Assigning an output label to each input step and collapsing repetitions is one method for aligning X and Y.

This approach has a significant flaw: it is often unnecessary to require each input step to be aligned with some result. For instance, the input may have periods of silence without a matching output. To get around these issues, CTC adds an additional output token to the list of permitted outputs. Occasionally, this novel token is referred to as a blank token. Hereafter, we'll refer to it as $\epsilon$. The epsilon token has no significance and is thus omitted from the output. CTC allows alignments that are identical in length as the input. After merging repetitions and deleting $\epsilon$ tokens, we permit any alignment that maps to Y.

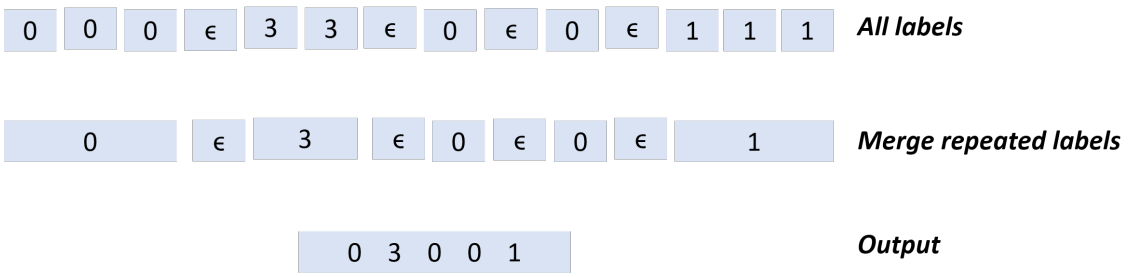


Figure 5.4: Label merging in the CTC

Consider the speech recognition case. If Y has two consecutive occurrences of the same character, an epsilon must be placed between them for a valid alignment. With this mechanism, we can tell the difference between alignments that collapse to **bee** and those that collapse to **be**; similarly, we can tell the difference between alignments that collapse to **off** and those that collapse to **of** , because omitting a letter can change the entire meaning of a word, let alone a sentence.

The CTC alignments exhibit several noteworthy features [77]. To begin, the allowed alignments relating X and Y are monotonic. We may either keep the matching output unchanged or progress to the next one if we go to the next input. A second characteristic is the many-to-one alignment of X and Y. A single input component may be aligned with one or more output components, but not the other way around. This implies the existence of a third property: Y cannot be more than X in length.

The CTC alignments provide a straightforward path from the probabilities at every time-step to the likelihood of an output sequence. To be more exact, the CTC goal for a single pair (X, Y) is as follows:

$$p(Y \mid X) = \sum_{A \in A_{X,Y}} \prod_{t=1}^{T} p_t(a_t \mid X) \tag{5.1}$$

Where the sum denotes marginalizes over the set of acceptable alignments and the products denotes the step-by-step computation of the probability for a particular alignment.

To estimate the per-time-step probabilities, $p_t(a_t \mid X)$, models trained using CTC commonly employ a recurrent neural network (RNN). Generally, the RNN performs well because it takes context into consideration in the input. If we are not cautious, computing the CTC loss can be rather costly. We could use a more direct method and compute

the score for each alignment as we go, adding them all up. The issue is that there may be an infinite number of alignments. This would be far too sluggish for the majority of problems. Fortunately, we can calculate the loss considerably more quickly using a dynamic programming approach. The critical idea is that if two alignments get at the same result in the same phase, they can be combined. Because an epsilon might occur before or after any character in Y, it's easier to illustrate the process using a sequence $Z = [\epsilon, y_1, \epsilon, y_2, \epsilon, y_3, \epsilon, ..., \epsilon, y_M, \epsilon, ]$ that contains them. We'll concentrate on the sequences at the start, ending, and between each character.

After the loss function has been accurately estimated, the next step is to compute the gradient, and then the model will be trained. The CTC loss function is differentiable with relation to the per-time-step output probabilities since all it is is the sums and products of the probabilities at each time step. With this knowledge, we are able to do an analytical calculation to determine the loss function's gradient in terms of the outcome probabilities, and we can then proceed with back-propagation as we normally would. Instead of explicitly maximizing the likelihood for a training set D, the parameters of the model are adjusted to reduce the amount by which the negative log-likelihood $\sum_{X,Y \in D} -log(p(Y \mid X))$ is increased.

After training the model, we'd want to utilize it to determine the most probable output for a particular input. To put it another way, we need to resolve:

$$Y^* = \underset{Y}{\operatorname{argmax}}\, p(Y \mid X) \tag{5.2}$$

At each time step, one heuristic is to choose the most probable outcome. This yields the alignment with the greatest likelihood:

$$A^* = \underset{A}{\operatorname{argmax}} \prod_{T=1}^{T} p_t(a_t \mid X) \tag{5.3}$$

Y is then obtained by collapsing duplicates and removing epsilon tokens. This heuristic is effective in a wide variety of applications, particularly when the majority of the probability mass is assigned to a single alignment. This technique, however, may sometimes overlook obvious outcomes with a far greater likelihood. The issue is that it ignores the

possibility of many alignments for a single output. Consider the following. Assume that [angry, angry, epsilon] and [angry, angry, angry] have a smaller probability individually than [happy, happy, happy]. However, their combined probability exceed those of [happy, happy, happy]. The naïve heuristic will offer Y = [happy] as the most probable hypothesis wrongly. Y = [angry] should have been picked. To correct this, the algorithm must take into consideration the fact that [angry, angry, angry] and [angry, angry, epsilon] produce the same result.

This issue may be rectified by doing a modified beam search. Due to the limited computer power available, the modified beam search may not always identify the most likely Y. However, it has the benefit of enabling us to trade off more processing (a larger beam size) for an asymptotically improved solution. A typical beam search creates a fresh series of hypotheses at each input step. The new set of hypotheses is constructed by extending each hypothesis to include all possible output characters and preserving just the top selections. We may expand the default beam search to include several alignments for the same output. Rather of keeping a list of internal alignments, we preserve the output prefixes after compressing repeats and removing epsilon characters. At each step of the search, we accumulate points for each alignment that corresponds to a certain prefix, such as audio, and its related output sequences.

So to recap and make it simpler, we want to map to sequence of audio signals $X = [X_1, X_2, X_3, ..., X_T]$ to a corresponding label sequence $Y = [Y_1, Y_2, Y_3, ..., Y_M]$. Unfortunately, such alignment is hard to obtain since emotions are not fixed and unchangeable, yet, they constantly change throughout time and the ratio of the lengths of both sequences can vary. One other thing to be mentioned is that, when there is no accurate alignment, manual alignment is not practical and it is time-consuming. The CTC loss averts all these challenges by taking as inputs, the output of the CNN-BiLSTM along with the corresponding sequence of ground-truth labels and accomplish the task without any assistance. It will provide an output distribution across all potential outputs of a specific input. This distribution can be used to infer a likely output or to estimate the likelihood of a particular output. So, what we want is to get the most likely output. We can do that by calculating Equation 5.2.

Note that earlier we have stated that the output of the model is n+1. The additional

unit, which is a blank that denotes the separation between two different speeches having different emotions. Which means that whenever a change of labels occurs, the blank label is added. For this reason, the CTC uses a function F to map the sequence of probabilities S to a sequence of predicted labels Y. The function F works by eliminating the repeated labels along with the blank label. So, given a sequence S, which denotes the output of the softmax, the conditional probability of having an output sequence Y given an input sequence X is:

$$p(Y \mid X) = \sum_{S \in F^{-1}(Y)} p(S \mid X) \tag{5.4}$$

So, the input of the CTC will be the output of the softmax function of different time-steps. The CTC will parse the output, once it finds the blank label, it eliminates it and combines all successive similar labels into one label.

## 5.3 Data preparation

Since the existing datasets are already pre-segmented, we create a new dataset by combining two or more utterances together to obtain long input sequences. Combining the utterances involves combining the labels of each single utterance, so we go from an utterance and its corresponding label to a list of utterances and its corresponding list of labels. Two datasets have been used to test our model: the first is the RAVDESS and the RML

In order to be able to detect the emotion change, we need long duration audio sequences of one or more person expressing several emotions successively, which is not the case with existing datasets for Speech Emotion Recognition. The existing datasets are composed of pre-segmented speeches where each audio file contains one single person talking and expressing one single emotion. For this reason, we have randomly combined several speeches together. Each sequence contains from one to four different speeches. The speeches are combined randomly where a sequence could contain either several speeches of the same person or different persons.

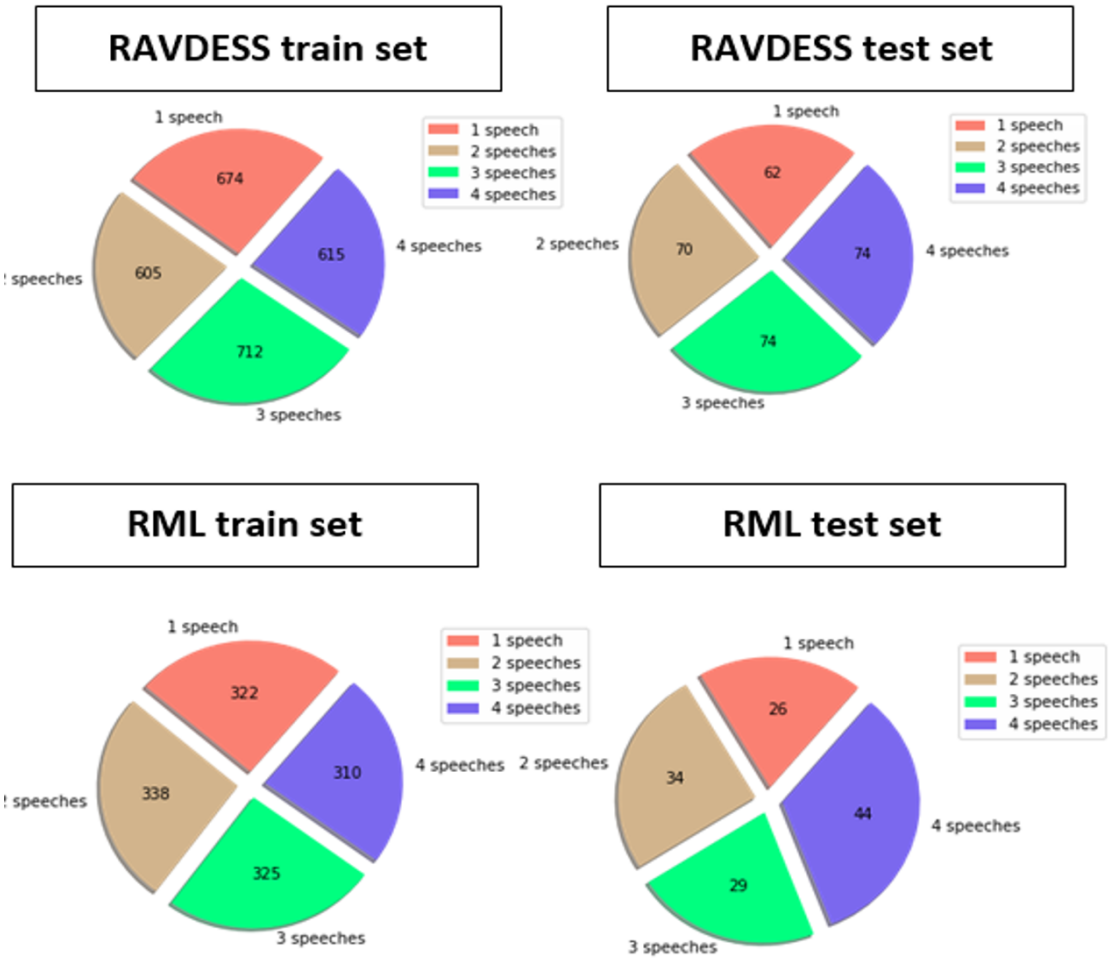


Figure 5.5: The combinations of sequences for both datasets

## 5.4 Experiments and results

Approximately, 80% of the data were used for training, 10% to fine tune and validate the model whereas 10% of the data were used to test it.

### 5.4.1 Model tuning

The CNN is composed of two Convolutional layers with ReLU activation, two Max-Pooling layers and a flatten layer. We have used three layers of BiLSTM followed by a Dense layer and a softmax layer. A dropout of 0.1 value is used to prevent the over-fitting. For the CTC, we have used the CTC Keras model [78]. As for the optimization, we have used the Adam optimizer with a learning rate equals to $10^{-4}$.

### 5.4.2 Evaluation metrics

As mentioned in Section 5.1, there was not much research in the domain of speech change detection which makes it hard to establish comparisons. For this particular reason, we propose the ECER metric which is inspired from the WER [79] that is used to determine a speech recognition system's performance. Hence, this metric could be used as reference for future researches.

### 5.4.3 Emotion Change Error Rate (ECER)

Given two sequences of labels, the first represents the GT labels and the second represents the model's prediction, the ECER is calculated as follows:

$$ECER = \frac{S + D + I}{N} \tag{5.5}$$

Where S is the number of labels that were replaced, D is the number of the labels that were disregarded, I is the number of labels that were inserted, C is the number of correct labels and N is the number of emotions in the GT sequence (N=S+D+C). The Accuracy is thus can be calculated as:

$$Accuracy = 1 - ECER \tag{5.6}$$

These two metrics do not only measure if the system has successfully detected emotions change, but they also measure whether or not the system has recognized the expressed emotion.

### 5.4.4 Emotion Change Detection (ECD)

Although the ECER tells a lot about the system performance, the goal here is to determine whether or not our model is capable of detecting all the emotional changes that have occurred in a sequence. Given a test set T of size m and a prediction list P of size m, the Emotion Change Detection (ECD) rate is calculated as follows:

$$ECD = \frac{1}{m} \sum_{t \in T, p \in P}^{m} E(t, p) \tag{5.7}$$

Table 5.1: ECER & Accuracy on the two datasets

| | | ECER | Accuracy |
|---|---|---|---|
| RML | Original dataset | 58.34% | 41.66% |
| | Augmented dataset | 16.32% | 83.68% |
| RAVDESS | Original dataset | 62.3% | 37.70% |
| | Augmented dataset | 21.19% | 78.81% |

Table 5.2: ECD of the two datasets

| | | ECD |
|---|---|---|
| RML | Original dataset | 77% |
| | Augmented dataset | 100% |
| RAVDESS | Original dataset | 65% |
| | Augmented dataset | 100% |

Where E(X,Y)=1 if the length of X equals the length of Y and 0 if not.

### 5.4.5 Results

For each one of the datasets, two experiments have been conducted: the first using the original dataset and the second using the augmented dataset. Table 5.1 shows the ECER and the Accuracy of each one of the experiments. For both original datasets (without augmentation) the ECER was too high and the accuracy was too low. The training accuracy was also too low leaning that the model suffers from under-fitting and it was not able to learn. The model was too deep and the amount of data was not enough for such model. With more data, both the ECER and the accuracy were improved significantly for both datasets.

Table 5.2 shows the ECD of the two datasets. Since the model was not able to learn due to the lack of data, the ECD was a little bit low. However, with more data to well train the model, the results were improved.

## 5.5 Analysis

First, we have tested our model on totally random data sequences. The results have shown that the size of the dataset matters and affects the accuracy i.e., the more data we have, the better results we get. And although the RAVDESS has bigger size, it achieved less accuracy values compared to the RML and this can be explained by the fact that

RAVDESS is considered to be as one of the hardest datasets since the human accuracy for this dataset is around 60Second, we have adjusted manually some of the input sequences for both training and testing datasets. For example, with a dataset of four emotion categories, we would have five possible outputs $y_1, y_2, y_3, y_4, y_5$ where $y_5$ denotes the blank, that we will use to detect the change. If the output of the model is $[y_1y_1y_5y_2y_5y_3y_3y_3y_3y_5y_1]$ then the final result after using the CTC should be $[y_1y_2y_3y_1]$. We have formed some data sequences where there are two consecutive speeches of different people but with the same label and two successive consecutive speeches of the same person with the same label. The goal here is to determine whether the CTC will be capable of separating between two consecutive utterances with the same label or it will just consider them as one single speech. The ECD was always 100% which means our model has successfully learned to separate between different speeches even though they have the same label. This could be helpful when trying to deploy our model in real time emotion detection system in conversation, which means the system will be capable of determining the emotional state of each speaker independently of the other. Getting an ECD equals to 100% and a low value for the accuracy, can be interpreted by the fact that the model has succeeded to detect all the emotion changes through all the sequences, yet it failed to recognize the emotions, i.e., for each input sequence, the model succeeded to detect a change has been occurred but sometimes fails the determine what is the label. The task of well recognizing the emotions remains a challenge as, to the best of our knowledge, none of the recent researches have achieved more than 90% accuracy for both datasets.

## 5.6 Conclusion

Detecting categories changes in emotional speeches has been the focus of this chapter. We have introduced a model which was capable of successfully detecting emotion categories changes. Yet, the model in some cases struggles to recognize emotions.

To evaluate our model, we have proposed new evaluation metrics: the ECER to determine the performance of the system in detecting the emotional change and recognizing emotions, and the ECD to determine whether or not the model has detected all emotional changes. The amount of the data was not enough to train the neural network so we

have used data augmentation techniques to increase the amount of data, which helped in improving the accuracy.

# Chapter 6

# Conclusions and Perspectives

The main objective of this thesis was to design a robust system for Speech Emotion Recognition.

## 6.1 Summary of contributions

Through this thesis, we was able to present several significant contributions and new perspectives:

- We have introduced a new classification model based on Time Distributed CNN and classic Transformer, which helped improve the accuracy on a single original dataset (before adding augmented data).
- The failure of the proposed model on the second database prompted us to shift our focus from speech classification to speech encoding. We suggested a model for speech encoding, which takes an audio file and generates a vector with 128 values. First, we have used the Triplet loss which improved the results comparing to previous researches, then we have proposed our new loss, an extension of the Triplet loss which we called Multiplet loss. The Multiplet loss outperformed the Triplet loss in term of both accuracy and execution time.
- Emotions can be interpreted differently depending on the person. As a result, we considered developing a multi-label classifier. This new insight allows a machine to assess all possible categories of emotions that humans can identify. Such a model

requires a dedicated dataset in which each file must include one or more labels simultaneously. To our knowledge, all existing databases are cataloged and annotated using a single label. Accordingly, we hand-labeled the RML dataset with the help of 50 volunteers.

- Emotions are generally expressed when making conversation with other people. Existing models determine the emotions from a single speech of a single person, which urged the need to implement a system that detects the emotion change in conversations. We introduced a system that was the first of its kind, that detects the emotion categories change.

The previously mentioned contributions helped to build a robust Speech Emotion Recognition system. Nonetheless, there are still some open concerns, such as: is the model suitable for industrial deployment? Is it enough to identify emotions based on real-world data?

Real-world data must be utilized to train the model if these questions are to have any hope of being answered.

As we stated in the first chapter, current datasets are recorded in studios with highly sophisticated technology in order to minimize noise and provide better signals, which is not the case with real-world data, which is why we need to create new datasets. As a result, it is advised that a new realistic dataset be constructed using data with noisy backgrounds (Phone calls, conversations in public places, etc.)

## 6.2 Perspectives

Despite the fact that our contributions managed to improve the accuracy in comparison to the state of the art, some improvements are still required.

### 6.2.1 Short term perspectives

Encoding the speeches and generation vectors turns out to be very efficient in improving the accuracy. However, there still some concerns about its efficiency. Let's pick up from where we left (chapter 4). Figure 6.1 shows the 5th step of training the encoder using both Triplet and Multiplet losses.

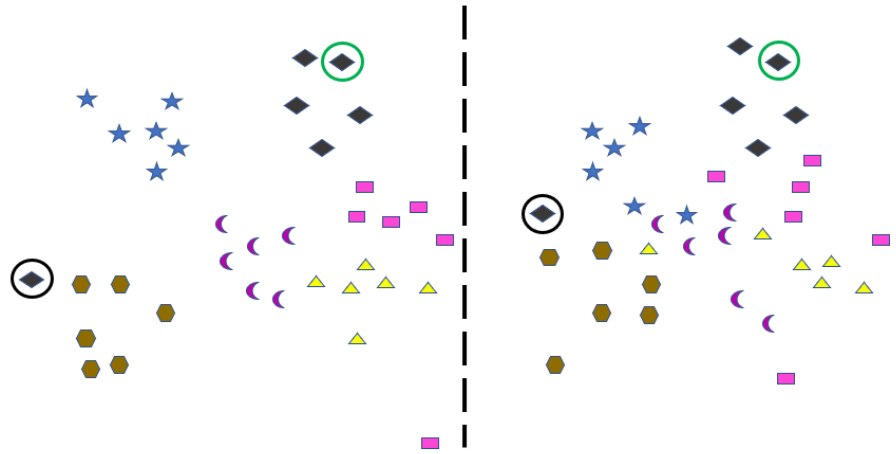

Figure 6.1: The fifth iteration of Multiplet and Triplet losses functions

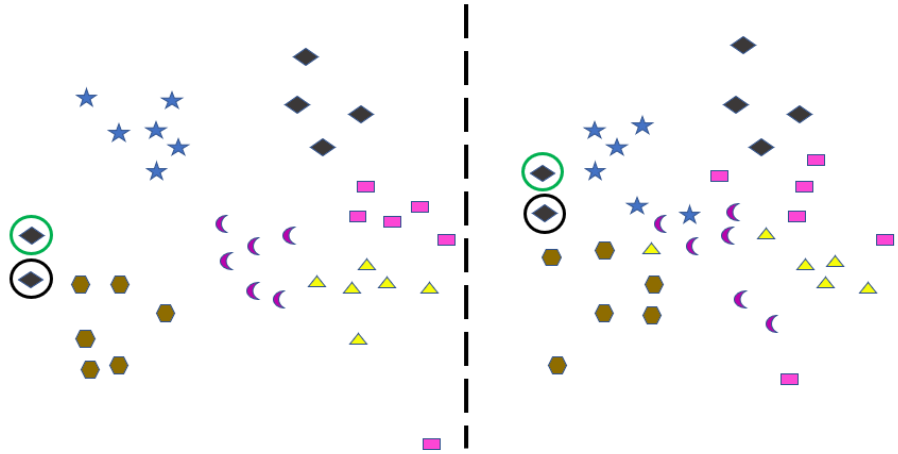

Figure 6.2: The fifth iteration of Multiplet and Triplet losses functions: Result

Now let's assume for the next step the model will pick the points encircled with the red circle.

Once the algorithm executes, it will choose a positive point for the anchor and pulls it towards it. Figure 6.2 this step will cause confusion among our groups, which will in turn cause confusion among the results and the accuracy.

For such reason, we think to, if possible, consider fixing a centroid for each group and then map all the embeddings near to each corresponding centroid.

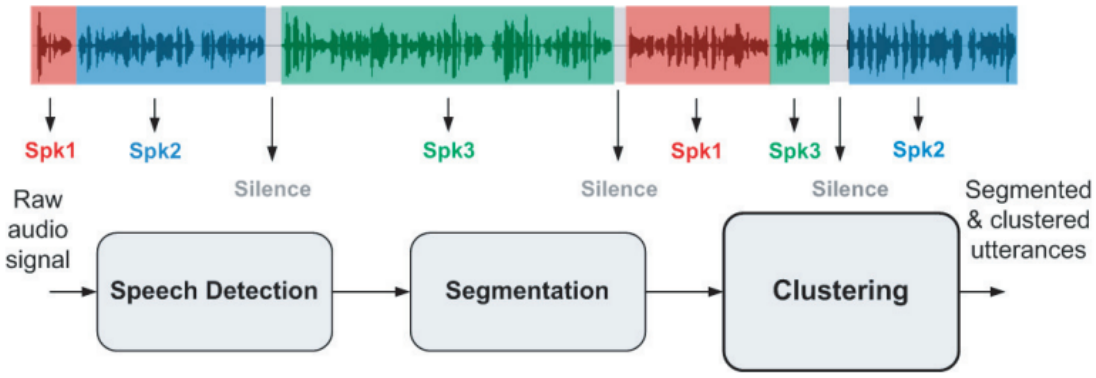


Figure 6.3: An example of speaker diarization system [80]

Table 6.1: The influence of the speaker's gender on the diarization system

| Speaker 1 | Speaker 2 | Result of the diarization system |
|---|---|---|
| Woman | Man | Works well |
| Woman | Woman | Wrong clustering |
| Man | Man | Wrong Clustering |

### 6.2.2 Medium term perspectives

While working on the detection of emotional changes, we sought to examine the emotional evolution of a single individual. This type of study can serve a number of purposes, including assessing the level of customer satisfaction in call centers. To do this, we first need to identify the portions where each person is speaking using a speaker diarization system (Figure 6.3). To avoid confusion with the task of speaker identification, which involves comparing a new speaker's identification to a collection of pre-trained speaker models, the goal of speaker diarization entails identifying speech segments belonging to the same speaker without any previous information.

We used several predefined diarization systems such as [81][82]. The diarization system performs admirably when neutral speeches are used in conversation (Speeches with label=Neutral); however, when emotional speeches are used in conversation, the system fails to distinguish between segments of different people unless the speakers are of different genders. Table 6.1 shows that of the two participants in the conversation are a man and a woman, the system will successfully accomplish the diarization task, otherwise, it gets more confusing for it. In order to avoid this, we need increase our attention on diarization system but taking emotional speeches into consideration.

### 6.2.3 Long term perspectives

It has already been revealed that we are experimenting with acted emotional datasets, which may have an impact on the accuracy of the system when it is used in the industry. The very first step in addressing this problem is to collect a more realistic dataset that includes noise and interfering conversations. That would enable us to assess the accuracy of the results in real-world scenarios. Once the data has been obtained, we should devote some time and effort to the pre-processing phase that will be required in order to remove the noise from the background signals.

# Bibliography


[1] Eduardo Bericat. "The sociology of emotions: Four decades of progress". In: *Current Sociology* 64.3 (2016), pp. 491–513.

[2] Joseph E LeDoux. "Evolution of human emotion: a view through fear". In: *Progress in brain research* 195.5 (2012), pp. 431–442.

[3] Cookson LJ. "Differences between feelings, emotions and desires in terms of interactive quality". In: *Advances in Social Sciences Research Journal* 2 (2015).

[4] Ekman P. "Basic emotions". In: *Handbook of Cognition and Emotion* 29 (2005), pp. 45–60. DOI: doi:10.1002/0470013494.ch3.

[5] Plutchik R. "In search of the basic emotions". In: *Contemp Psychol J Rev* 29 (1987), pp. 511–513.

[6] Lerner JS et al. "Emotion and decision making". In: *Annu Rev Psychol* 66 (2015), pp. 799–829.

[7] Shaver TK et al. "Long-term deficits in risky decision-making after traumatic brain injury on a rat analog of the Iowa gambling task". In: *Brain Res.* 1704 (2019), pp. 103–113.

[8] Lech Margaret et al. "Real-Time Speech Emotion Recognition Using a Pre-trained Image Classification Network: Effects of Bandwidth Reduction and Companding". In: *Frontiers in Computer Science* 2 (2020). DOI: 10.3389/fcomp.2020.00014.

[9] Shaikh Nilofer R. A. et al. "A Review. International Journal of Scientific Engineering Research". In: *PloS one* 6.4 (2015).

[10] Mathew Lima and Salim Sajin. “Real World Speech Emotion Recognition from Speech Spectrogram Using Gray-Level Co-Occurence Matrix”. In: *International Journal of Innovative Research in Computer and Communication Engineering* 5.4 (2017).

[11] Siddique Latif et al. “Transfer Learning for Improving Speech Emotion Classification Accuracy”. In: *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*. Ed. by B. Yegnanarayana. ISCA, 2018, pp. 257–261.

[12] Dias Issa, M Fatih Demirci, and Adnan Yazici. “Speech emotion recognition with deep convolutional neural networks”. In: *Biomedical Signal Processing and Control* 59 (2020), p. 101894.

[13] Mustaqeem, Muhammad Sajjad, and Soonil Kwon. “Clustering-Based Speech Emotion Recognition by Incorporating Learned Features and Deep BiLSTM”. In: *IEEE Access* 8 (2020), pp. 79861–79875.

[14] Shashidhar G., Koolagudi K., and Sreenivasa Rao. “The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English”. In: *Int J Speech Technol* (2012), pp. 99–117.

[15] Switzerland EPFL (École polytechnique fédérale de Lausanne). *Digital Signal Processing*. URL: `https://www.coursera.org/learn/dsp1` (visited on 04/24/2022).

[16] Stanley Smith Stevens, John Volkmann, and Edwin Broomell Newman. “A scale for the measurement of the psychological magnitude pitch”. In: *The journal of the acoustical society of america* 8.3 (1937), pp. 185–190.

[17] Yafeng Niu et al. “A breakthrough in Speech emotion recognition using Deep Retinal Convolution Neural Networks”. In: *CoRR abs/1707.09917* (2017).

[18] Jianfeng Zhao, Xia Mao, and Lijiang Chen. “Speech emotion recognition using deep 1D & 2D CNN LSTM networks”. In: *Biomedical signal processing and control* 47 (2019), pp. 312–323.

[19] Promod Yenigalla et al. “Speech Emotion Recognition Using Spectrogram & Phoneme Embedding”. In: *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*. Ed. by B. Yegnanarayana. ISCA, 2018, pp. 3688–3692.

[20] Anwer Slimi et al. “Emotion recognition from speech using spectrograms and shallow neural networks”. In: *Proceedings of the 18th International Conference on Advances in Mobile Computing & Multimedia*. 2020, pp. 35–39.

[21] K.O’Shea and R. Nash. “An introduction to convolutional neural networks”. In: *arXiv preprint, arXiv:1511.08458*. (2014).

[22] *Example of 2D Convolution*. URL: `https://github.com/PetarV-/TikZ/tree/master/` (visited on 12/19/2021).

[23] *Example of Max pooling*. URL: `https://datascientest.com/convolutional-neural-network` (visited on 12/19/2021).

[24] *Deep Learning Specialization*. URL: `https://www.coursera.org/specializations/deep-learning` (visited on 07/05/2021).

[25] *General architecture of a Transformer*. URL: `https://deepfrench.gitlab.io/deep-learning-project/` (visited on 04/24/2022).

[26] Lilian Weng. *Attention? Attention!* URL: `https://lilianweng.github.io/posts/2018-06-24-attention/` (visited on 03/24/2022).

[27] Thurid Vogt, Elisabeth André, and Johannes Wagner. “Automatic Recognition of Emotions from Speech: A Review of the Literature and Recommendations for Practical Realisation”. In: *Affect and Emotion in HCI* (2008), pp. 75–91.

[28] F. Ringeval et al. “Introducing the RECOLA Multimodal Corpus of Remote Collaborative and Affective Interactions”. In: *EmoSPACE*. 2013.

[29] C. Busso et al. “IEMOCAP: Interactive emotional dyadic motion capture database”. In: *Journal of Language Resources and Evaluation* 42.4 (2008).

[30] F. Burkhardt et al. “A Database of German Emotional Speech”. In: *INTERSPEECH*. 2005.

[31] Inger S. Engberg et al. “Design, recording and verification of a danish emotional speech database”. In: *Proc. 5th European Conference on Speech Communication and Technology (Eurospeech 1997)*. 1997, pp. 1695–1698.

[32] Vladimir Hozjan et al. “Interface Databases: Design and Collection of a Multilingual Emotional Speech Database”. In: *LREC*. 2002.

[33] Donn Morrison, Ruili Wang, and Liyanage C. De Silva. “Ensemble methods for spoken emotion recognition in call-centres”. In: *Speech Communication* 49 (2007), pp. 98–112.

[34] Kate YDupuis and M K. Pichora-Fuller. *Toronto Emotional Speech Set (tess)*. University of Toronto, Psychology Department, 2010.

[35] Vladimir Hozjan et al. “Study on Emotional Speech Features in Korean with Its Application to Voice Conversion”. In: *ACII*. 2005, pp. 342–349.

[36] Pierre-Yves Oudeyer. “The production and recognition of emotions in speech: features and algorithms”. In: *Int. J. Human-Computer Studies* 59 (2003), pp. 157–183.

[37] China Institute of automation Chinese academic of science. *CASIA-Chinese Emotional Speech Corpus*. URL: `http://shachi.org/resources/27` (visited on 04/24/2022).

[38] Giovanni Costantini et al. “EMOVO Corpus: an Italian Emotional Speech Database”. In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*. 2014, pp. 3501–3504.

[39] R. Haralick, K. Shanmugam, and I. Dinstein. “Textural Features for Image Classification”. In: *IEEE Trans. on Systems, Man and Cybernetics* SMC–3.6 (1973), pp. 610–621.

[40] Florian Eyben et al. “The Geneva Minimalistic Acoustic Parameter Set (GeMAPS) for Voice Research and Affective Computing”. In: *IEEE Transactions on Affective Computing* (). DOI: `10.1109/TAFFC.2015.2457417`.

[41] Kun-Ching Wang. “Speech Emotional Classification Using Texture Image Information Features”. In: *International Journal of Signal Processing Systems* 3.1 (2015).

[42] David Dorran, Robert Lawlor, and Eugene Coyle. “High Quality Time-Scale Modification of Speech Using A Peak Alignment Overlap-Add Algorithm (Paola)”. In: *ICASSP*. 1988.

[43] David G. Glynn. “On cubic curves in projective planes of characteristic two”. In: *Australasian Journal of Combinatorics* 17 (2018), pp. 1–20.

[44] Seyedmahdad Mirsamadi, Emad Barsoum, and Cha Zhang. “Automatic Speech Emotion Recognition Using Recurrent Neural Networks with Local Attention”. In: *ICASSP*. 2017, pp. 2227–2231.

[45] Noushin Hajarolasvadi and Hasan Demirel. “3D CNN-Based Speech Emotion Recognition Using K-Means Clustering and Spectrograms”. In: *Entropy* (2019), p. 497.

[46] Lili Guo et al. “Speech Emotion Recognition by Combining Amplitude and Phase Information Using Convolutional Neural Network”. In: *INTERSPEECH*. 2018. DOI: 10.21437/Interspeech.2018-2156.

[47] R. M. Hegde, H. A. Murthy, and V. R. R. Gadde. “Significance of the modified group delay feature in speech recognition”. In: *IEEE Trans. Audio, Speech and Language Proc.* 14 (2007), pp. 190–202.

[48] L. Wang et al. “Relative phase information for detecting human speech and spoofed speech”. In: *INTERSPEECH*. 2015, pp. 2092–2096.

[49] Roopa S. Nithya, M Prabhakaran, and P Betty. “Speech Emotion Recognition using Deep Learning”. In: *International Journal of Recent Technology and Engineering IJRTE* 7 (2018), pp. 2277–3878.

[50] John W. Kim and Rif A. Saurous. “Emotion Recognition from Human Speech Using Temporal Information and Deep Learning”. In: *INTERSPEECH*. 2018.

[51] Leila Kerkeni et al. “Speech Emotion Recognition: Methods and Cases Study”. In: *ICAART*. 2018, pp. 175–182.

[52] Aouani Hadhami and Yassine Ben Ayed. “Speech emotion recognition with deep learning”. In: *Procedia Computer Science* 176 (2020), pp. 251–260.

[53] Mustaqeem and Soonil Kwon. “A CNN-assisted enhanced audio signal processing for speech emotion recognition”. In: *Sensors* 20.1 (2019), p. 183.

[54] Aneesh Muppidi and Martin Radfar. "Speech emotion recognition using quaternion convolutional neural networks". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, pp. 6309–6313.

[55] Leonardo Pepino, Pablo Riera, and Luciana Ferrer. "Emotion recognition from speech using wav2vec 2.0 embeddings". In: *arXiv preprint arXiv:2104.03502* (2021).

[56] Minji Seo and Myungho Kim. "Fusing visual attention CNN and bag of visual words for cross-corpus speech emotion recognition". In: *Sensors* 20.19 (2020), p. 5559.

[57] Ashkan Yazdani et al. "Multimedia content analysis for emotional characterization of music video clips". In: *J Image Video Proc* 26 (2013). DOI: https://doi.org/10.1186/1687-5281-2013-26.

[58] Zhaocheng Huang, Julien Epps, and Eliathamby Ambikairajah. "EMOVO Corpus: an Italian Emotional Speech Database". In: *INTERSPEECH*. 2015, pp. 1329–1333.

[59] YZhaocheng Huang and Julien Epps. "EMOVO Corpus: an Italian Emotional Speech Database". In: *ICASSP*. 2016, pp. 5195–5199.

[60] Zhaocheng Huang and Julien Epps. "Prediction of Emotion Change from Speech". In: *Frontiers ICT* 5 (2018).

[61] Huang S. C. et al. "Fusion of medical imaging and electronic health records using deep learning: a systematic review and implementation guidelines". In: *NPJ digital medicine* 3.136 (2020), e0196391. DOI: https://doi.org/10.1038/s41746-020-00341-z.

[62] Egils Avots et al. "Audiovisual emotion recognition in wild". In: *Machine Vision and Applications* 30.5 (2019), pp. 975–985.

[63] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[64] Hugo Touvron et al. "Training data-efficient image transformers & distillation through attention". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 10347–10357.

[65] Yun Liu et al. “Transformer in convolutional neural networks”. In: *arXiv preprint arXiv:2106.03180* (2021).

[66] Pavel Karpov, Guillaume Godin, and Igor V Tetko. “Transformer-CNN: Swiss knife for QSAR modeling and interpretation”. In: *Journal of Cheminformatics* 12.1 (2020), pp. 1–12.

[67] Jiang Zhang et al. “A CNN-transformer hybrid approach for decoding visual neural activity into text”. In: *Computer Methods and Programs in Biomedicine* 214 (2022), p. 106586.

[68] Xiaohan Xia, Dongmei Jiang, and Hichem Sahli. “Learning Salient Segments for Speech Emotion Recognition Using Attentive Temporal Pooling”. In: *IEEE Access* 8 (2020), pp. 151740–151752.

[69] Mustaqeem and Soonil Kwon. “CLSTM: Deep feature-based speech emotion recognition using the hierarchical ConvLSTM network”. In: *Mathematics* 8.12 (2020), p. 2133.

[70] Daniel S Park et al. “Specaugment: A simple data augmentation method for automatic speech recognition”. In: *arXiv preprint arXiv:1904.08779* (2019).

[71] Florian Schroff, Dmitry Kalenichenko, and James Philbin. “FaceNet: A Unified Embedding for Face Recognition and Clustering”. In: *CVPR*. 2015, pp. 815–823.

[72] Weihua Chen et al. “Beyond Triplet Loss: A Deep Quadruplet Network for Person Re-identification”. In: *CVPR*. 2017, pp. 1320–1329.

[73] James Bergstra et al. “Algorithms for Hyper-Parameter Optimization”. In: *NIPS*. 2011, pp. 2546–2554.

[74] Ye Yuan et al. “In Defense of the Triplet Loss Again: Learning Robust Person Re-Identification with Fast Approximated Triplet Loss and Label Distillation”. In: *INTERSPEECH*. 2020, pp. 1454–1463.

[75] Ting K.M. “Confusion Matrix”. In: *Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning and Data Mining* (2017).

[76] Puneet Kumar et al. “End-to-end Triplet Loss based Emotion Embedding System for Speech Emotion Recognition”. In: arXiv, 2020. DOI: 10.48550/ARXIV.2010.06200. URL: https://arxiv.org/abs/2010.06200.

[77] Deguo Mu et al. “Japanese Pronunciation Evaluation Based on DDNN”. In: *IEEE Access* 8 (2020), pp. 218644–218657. DOI: 10.1109/ACCESS.2020.3041901.

[78] Yann Soullard, Cyprien Ruffino, and Thierry Paquet. *A Keras Model for Connectionist Temporal Classification*. université de Rouen Normandie, 2019.

[79] Park Youngja et al. “An empirical analysis of word error rate and keyword error rate”. In: *INTERSPEECH*. 2008, pp. 2070–2073.

[80] Hao Tang et al. “Partially Supervised Speaker Clustering”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.5 (2012), pp. 959–971. DOI: 10.1109/TPAMI.2011.174.

[81] Quan Wang et al. *Speaker Diarization with LSTM*. 2017. URL: https://arxiv.org/abs/1710.10468.

[82] Y. Fujita et al. “EMOVO Corpus: an Italian Emotional Speech Database”. In: *Proceedings of the Annual Conference of the International Speech Communication Association*. 2019, pp. 4300–4304.

# Appendices

# Appendix A

# Fourier Analysis[1]

Discrete-time representation In order to process audio files with computers, we need to shift from the analog world to digital world where we can represent our signal with a series of number (also known as a discrete signal).

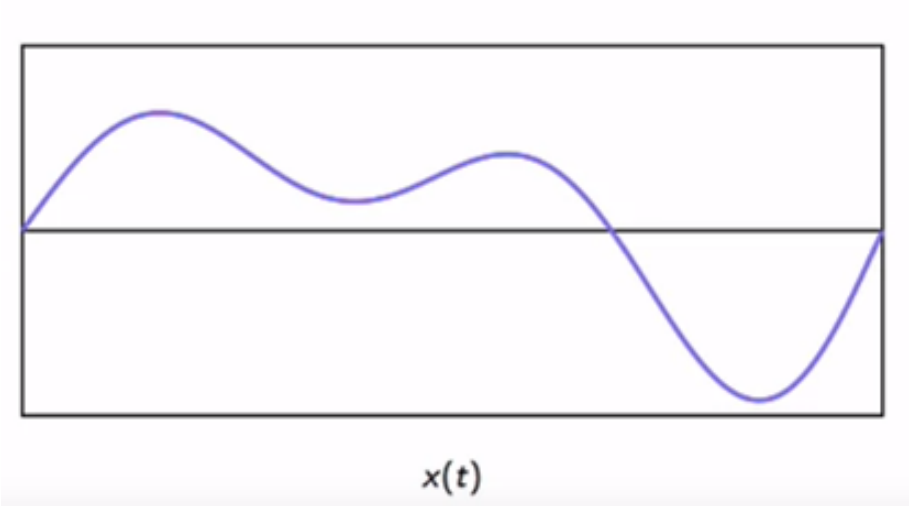


Figure A.1: Continuous signal

Before getting into details, there is some notions we need to be familiar with:

- The sampling period **Ts** which is the time interval between samples (in seconds).
- Periodicity of **M** samples ⇔ Periodicity of **M*Ts** seconds.
- Every audio file has a sampling rate **fs**, where $Ts = \frac{1}{fs}$

[1]Most information and figures gathered in this annex, are taken from a course provided by EPFL (École polytechnique fédérale de Lausanne), Switzerland.

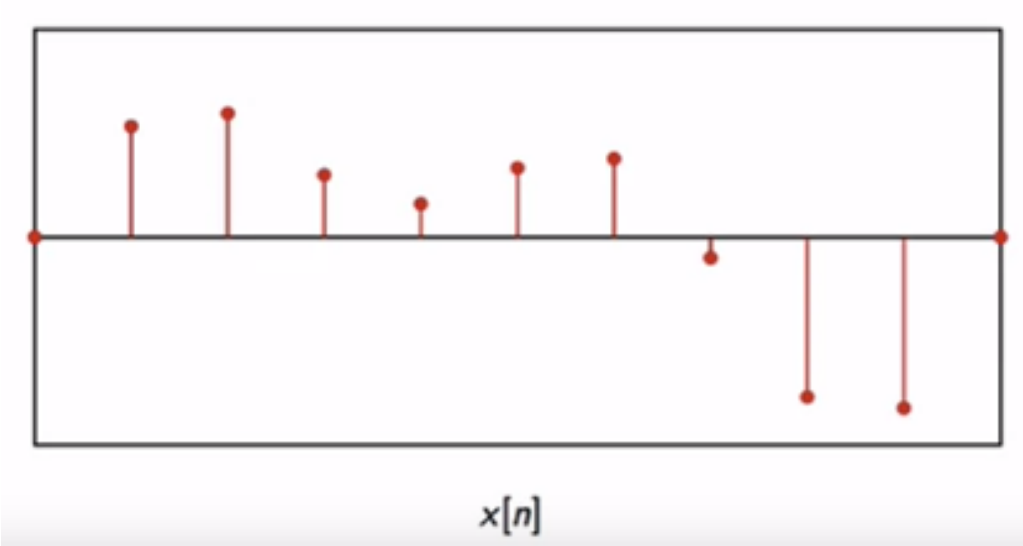


Figure A.2: Discrete signal

- Real world frequency is $f = \frac{1}{M*Ts}$
- The number of samples per second is $Fs = \frac{1}{Ts}$ hz

According to Nyquist and Shannon, the continuous and discrete-time representations are equivalent (under very mild conditions). To convert a continuous signal to a discrete signal (or vice versa), a technology called Fourier analysis is required. The Fourier transform will provide us with a quantifiable estimate of the speed at which a signal travels, and once we have that value, we can determine a sampling interval (time interval between measurements). Many natural and man-made phenomena exhibit an oscillatory behavior. After a certain amount of time, called the period, these phenomena come back to the same position. It thus makes sense to use sines and cosines as basic building blocks to represent these oscillatory signals. This is the basic goal of Fourier analysis: to decompose a signal in terms of sines and cosines. We distinguish two kinds of Fourier tools, Fourier analysis and Fourier synthesis. Fourier analysis allows moving from the time to the frequency domain and Fourier synthesis allows moving from the frequency domain to the time domain. The sampling theorem is:

$$x(t) = \sum_{n=-\infty}^{+\infty} x[n] sin(\frac{t - n * Ts}{Ts}) \tag{A.1}$$

The following is an example of the samples of an audio file:

x[n]= [ 22 101 83 ... 97 108 99]

So, an audio file is introduced to the computer as one dimensional array of integers.

The DFT can be seen as simply a change of basis, which is a change of perspective, since changing the basis can reveal things. Figure 18 shows that the signal has a different

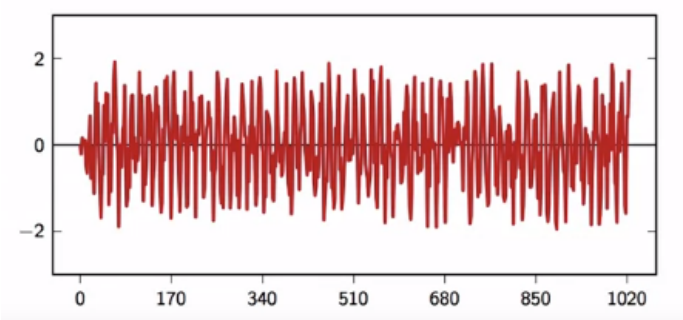


Figure A.3: Signal in time domain

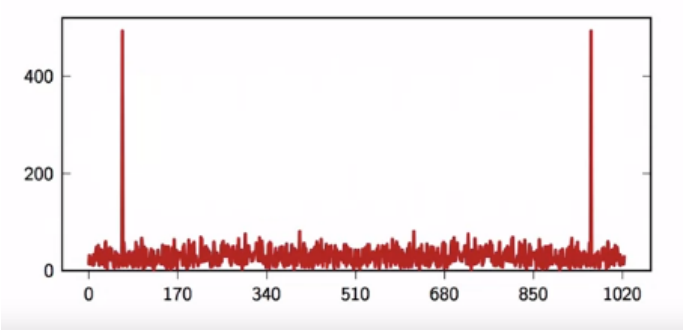


Figure A.4: Signal in Fourier-basis

behavior at the beginning and at the end that can be noticed only if we change the basis (to Fourier basis). **So how to change a signal into a Fourier-basis?**

A basis is a set of vectors ($w^{(k)}$). Let **N** be the size of the vector x[n]. Let **k** be the index of different vectors (k = 0, ... , N-1). Let **n** be the index of each element within each vector (n = 0, ... , N-1).

So, $\{w^{(k)}\}$ where $W_n^{(k)} = e^{j\frac{2\pi}{N}nk}; n, k = 0, ..., N-1$ , is an orthogonal basis in $C^N$. Now if we have a signal x[n] in time-domain, we can transform it to the frequency-domain using the Analysis formula as follows:

$$X[k] =< w^k, x[n] >= \sum_{n=0}^{N-1} x[n] e^{j\frac{2\pi}{N}nk} \tag{A.2}$$

And to go back from the frequency -domain to time-domain we use the Synthesis formula

as follows:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] w^{(k)} \tag{A.3}$$

(we use a scaling factor in the synthesis formula because the basis in not orthonormal).

In simple words, The STFT is applying a DFT on small pieces instead of applying a DFT on the whole signal.

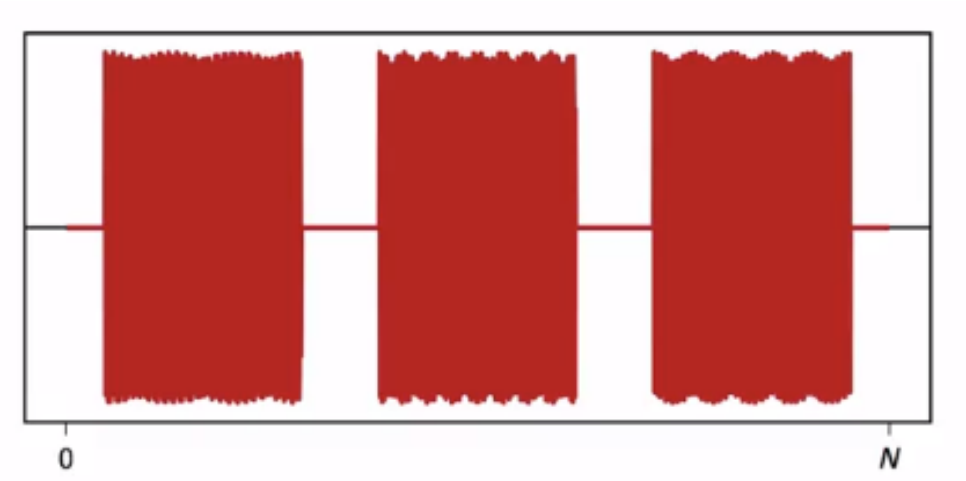


Figure A.5: Audio in time domain

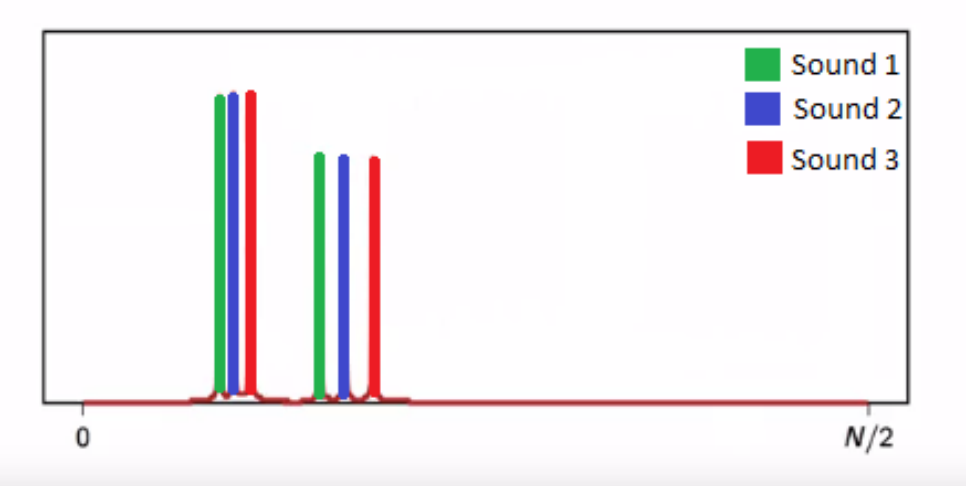


Figure A.6: Audio in frequency domain

Every presentation carries a different information (depending on the representation domain) so we need to search for a way that allows us to combine both presentations. And this is the idea behind the STFT. Let's remember that the equation behind STFT is:

$$X[m;k] = \sum_{n=0}^{L-1} x[m+n]e^{-j\frac{2\pi}{L}nk} \tag{A.4}$$

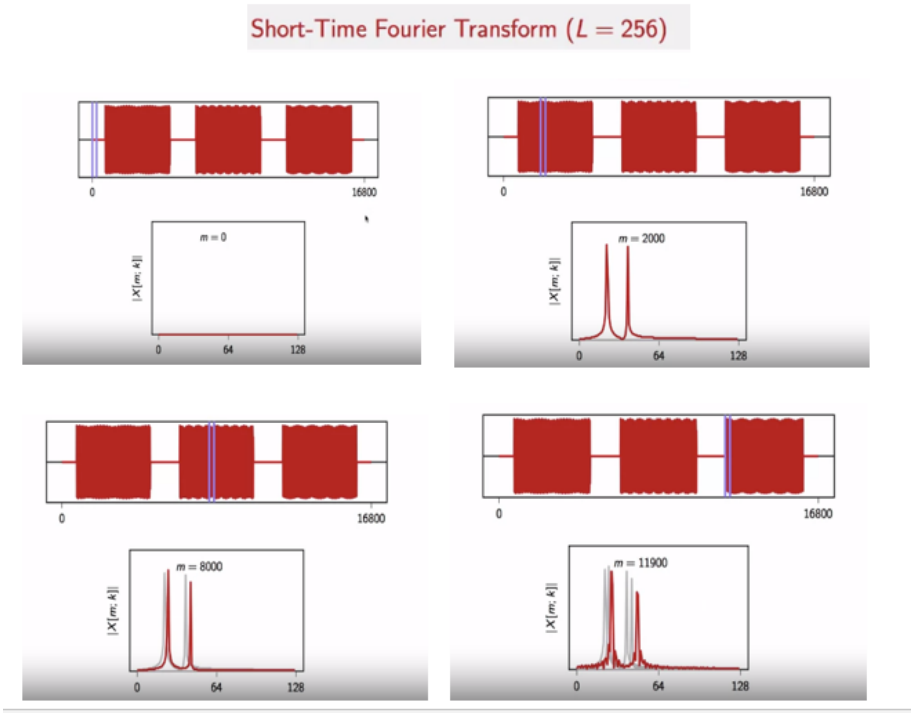


Figure A.7: Applying the STFT with L=256

Figure A.7 shows the result obtained by applying a STFT on an audio signal. The lower right picture shows the representation of the signal (once the STFT is done) in which both the time information and the frequency information are present.

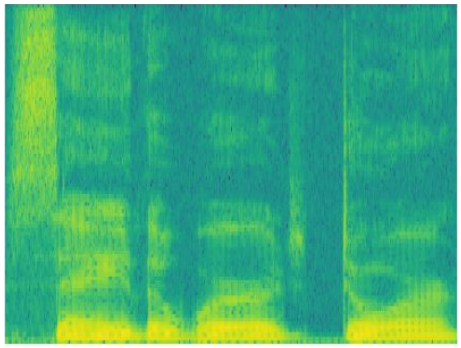

Figure A.8: Spectrogram with overlap= 120 ; window size =180

For the experiments, various spectrograms were used in our SER system:

- For the first experiment, we have used a spectrogram with low overlap and low window size (Figure A.8)

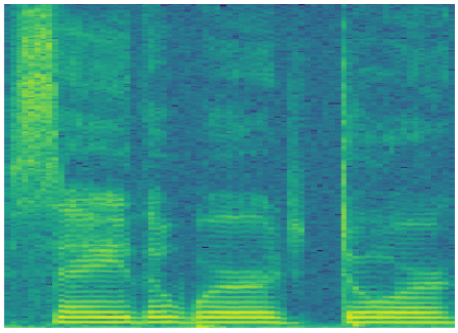

Figure A.9: Spectrogram with overlap= 120 ; window size =1048

- For the second experiment, we have used a spectrogram with low overlap and high window size (Figure A.9)
- For the third experiment, we have used a spectrogram with medium overlap and medium window size (Figure A.10)
- And for the fourth experiment, we have used a spectrogram with high overlap and high window size (Figure A.11)

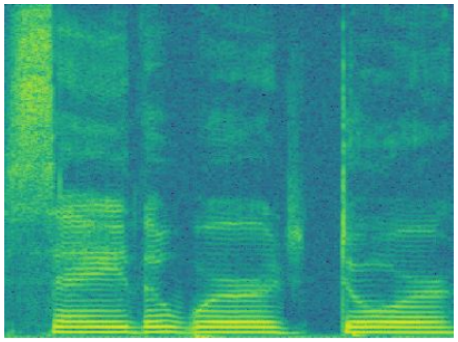

Figure A.10: Spectrogram with overlap=600 ; window size =750

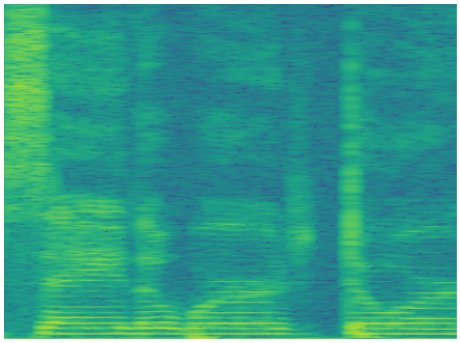

Figure A.11: Spectrogram with overlap= 3000 ; window size= 1048