University of Minho
School of Engineering

Guilherme Marques Andrade

Emotion Recognition: Recognition of emotions through voice

February 2022

University of Minho
School of Engineering

Guilherme Marques Andrade

Emotion Recognition: Recognition of emotions through voice

Master's Dissertation
Integrated Master's in Informatics Engineering

Work conducted under the orientation of
Professor Paulo Novais
Professor Manuel Rodrigues

February 2022

COPYRIGHT AND CONDITIONS OF USE OF THE WORK BY THIRD PARTIES

This is an academic work that may be used by third parties provided that the rules and good practices accepted internationally, regarding copyright and related rights are respected. Thus, this work may be used under the terms of the license below.

If the user needs permission to make a use of the work under conditions not provided in the license indicated, you should contact the author through RepositóriUM of the University of Minho.

Attribution-NonCommercial
CC BY-NC

https://creativecommons.org/licenses/by-nc/4.0/

ACKNOWLEDGMENTS

Firstly, I would like to thank the University as whole, for the chance provided to produce my work. It has been a long journey and I would not have it any other way.

I would like to thank my supervisors, Manuel Rodrigues and Paulo Novais. Since the start that they showed themselves available to help and solve any situation, any problem that would arise. It would be impossible without their collaboration and I would not be here writing this today.

I would like to thank my family, who's been there behind every step, believing and pushing, giving strength, living my life with me in the best way possibly imaginable.

To my friends, to those who stuck around after all this time, we've had amazing moments that I will cherish all of my life. And even through this whole year, mostly apart, these moments kept coming somehow.

To my community, who's everyday presence made life easy and fresh and kept me going day after day.

STATE OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

Assinado por: **Guilherme Marques E Andrade**
Num. de Identificação: 14458152
Data: 2022.02.17 12:10:26+00'00'

CHAVE MÓVEL

University of Minho, 28 / 12 / 2021

Signature: ____________________

ABSTRACT

As the years go by, the interaction between humans and machines seems to gain more and more importance for many different reasons, whether it's taken into consideration personal or commercial use. On a time where technology is reaching many parts of our lives, it's important to keep thriving for a healthy progress and help not only to improve but also to maintain the benefits that everyone gets from it. This relationship can be tackled through many points, but here the focus will be on the mind.

Emotions are still a mystery. The concept itself brings up serious questions because of its complex nature. Till the date, scientists still struggle to understand it, so it's crucial to pave the right path for the growth on technology on the aid of such topic. There is some consensus on a few indicators that provide important insights on mental state, like words used, facial expressions, voice.

The context of this work is on the use of voice and, based on the field of Automatic Speech Emotion Recognition, it is proposed a full pipeline of work with a wide scope by resorting to sound capture and signal processing software, to learning and classifying through algorithms belonging on the Semi Supervised Learning paradigm and visualization techniques for interpretation of results. For the classification of the samples,using a semi-supervised approach with Neural Networks represents an important setting to try alleviating the dependency of human labelling of emotions, a task that has proven to be challenging and, in many cases, highly subjective, not to mention expensive. It is intended to rely mostly on empiric results more than theoretical concepts due to the complexity of the human emotions concept and its inherent uncertainty, but never to disregard prior knowledge on the matter.

Keywords: Automatic Speech Emotion Recognition, Semi Supervised learning, Human emotion, Unlabeled dataset

RESUMO

À medida que os anos passam, a interacção entre indivíduos e máquinas tem vindo a ganhar maior importância por várias razões, quer seja para uso pessoal ou comercial. Numa altura onde a tecnologia está a chegar a várias partes das nossas vidas, é importante continuar a perseguir um progresso saudável e ajudar não só a melhorar mas também manter os benefícios que todos recebem. Esta relação pode ser abordada por vários pontos, neste trabalho o foco está na mente.

Emoções são um mistério. O próprio conceito levanta questões sobre a sua natureza complexa. Até aos dias de hoje, muitos cientistas debatem-se para a compreender, e é crucial que um caminho apropriado seja criado para o crescimento de tecnologia na ajuda da compreensão deste assunto. Existe algum consenso sobre indicadores que demonstram pistas importantes sobre o estado mental de um sujeito, como palavras, expressões faciais, voz.

O conteúdo deste trabalho foca-se na voz e, com base no campo de Automatic Speech Emotion Recognition, é proposto uma sequência de procedimentos diversificados, ao optar por software de captura de som e processamento de sinais, aprendizagem e classificação através de algoritmos de Aprendizagem Semi Supervisionada e técnicas de visualização para interpretar resultados. Para a classificação de amostras, o uso de uma abordagem Semi Supervisionada com redes neuronais representam um procedimentos importante para tentar combater a alta dependência da anotação de amostras de emoções humanas, uma tarefa que se demonstra ser árdua e, em muitos casos, altamente subjectiva, para não dizer cara. A intenção é estabelecer raciocínios baseados em factores experimentais, mais que teóricos, devido à complexidade do conceito de emoções humanas e à sua incerteza associada, mas tendo sempre em conta conhecimento já estabelecido no assunto.

Palavras-chave: Automatic Speech Emotion Recognition, Aprendizagem Semi Supervisionada, Emoção Humana, Dados sem anotação

LIST OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AE Autoencoder. xviii, 13, 14

AI Artificial Intelligence. xviii, 2, 7

ANN Artificial Neural Network. xviii, 10, 11, 13, 14

ASA Acoustic Society of America. xviii, 15

ASER Automatic Speech Emotion Recognition. xviii, 3–5, 12, 13, 16, 17, 19, 21, 24, 29, 41, 42, 46

ASR Automatic Speech Recognition. xviii, 12, 13

BCE Binary Cross Entropy. xviii, 27

BET Basic Emotion Theory. xviii, 42

BLSTMATT Bi Long Short Term Memory with attention. xviii, 17

CCC Concordance Correlation Coefficient. xviii

CNN Convolutional Neural Network. xviii, 12, 13, 21, 23, 24, 28, 33

CSA Convolutional Self-Attention. xviii, 17

DAE Denoising Autoencoder. xviii, 14

dB Decibel. xviii, 25

DCGAN Deep Convolutional Generative Adversarial Network. xviii, 18

DFT Discrete Fourier Transform. xviii, 24

DGM Deep Generative Model. xviii, 26

DL Deep Learning. xviii, 11, 21, 23, 26, 30, 32

FFT Fast Fourier Transform. xviii, 24

GAN Generative Adversarial Network. xii, xv, xviii, 14–16, 18, 23, 25–32, 34, 36–38, 45, 46

GAP Global Average Pooling. xviii, 30

GMM Gaussian Mixture Model. xviii, 4

GPU Graphics Processing Unit. xviii, 12

HLD High Level Descriptor. xviii, 23

HMM Hidden Markov Model. xviii, 4

IEMOCAP Interactive Emotional Dyadic Motion Capture Database. xv, xviii, 17, 18, 21, 25, 33, 34

LLD Low Level Descriptor. xviii, 23

LSTM Long Short Term Memory. xviii, 13

MEI Mestrado em Engenharia Informática. xviii

MFCC Mel Frequency Cepstral Coefficient. xviii, 23

ML Machine Learning. xi, xviii, 3, 5, 8–13, 15, 19, 21, 23, 29, 41, 43, 46

NLP Natural Language Processing. xviii, 12, 13

PCA Principal Component Analysis. xviii, 14

RL Reinforcement Learning. xviii, 9

RNN Recurrent Neural Network. xviii, 13, 21

SAE Sparse Autoencoder. xviii, 14

SGAN Semi supervised Generative Adversarial Network. xii, xviii, 16, 18, 27–29, 33, 34, 36, 38, 42, 43, 45, 46

SL Supervised Learning. xiii, xviii, 8, 37, 38, 46

SR Speech Recognition. xviii

SSL Semi Supervised Learning. xiii, xviii, 3, 4, 8, 17, 18, 26, 27, 30, 32, 37, 38, 42, 43, 46

SSSGAN Smooth Semi Supervised Generative Adversarial Network. xviii, 18

STFT Short-Time Fourier Transform. xviii, 24, 29

SVM Support Vector Machine. xviii, 4, 10

UA Unweighted Accuracy. xiii, xviii, 17, 18, 29, 32, 33, 35–37

UAR Unweighted Average Recall. xiii, xviii, 18, 29, 32–34, 36, 37

UL Unsupervised Learning. xviii, 8, 42

UM Universidade do Minho. xviii

VAE Variational Autoencoder. xviii, 14, 15

VLTP Vocal Tract Length Perturbation. xviii, 17

VSSSGAN Virtual Smooth Semi Supervised Generative Adversarial Network. xviii, 18

WA Weighed Accuracy. xviii, 17, 18, 29

1

INTRODUCTION

In this chapter, the main purpose and goals of this dissertation are described. It begins by referring the motivation and context behind such project. It then proceeds to an expectation's measurement regarding results, followed by the chosen methodologies to approach the subject. Finally both the planning for the project and this document itself are explained. It is intended to give the reader important concepts regarding two components to approach emotions: artificial intelligence and psychology concepts. It will help justifying decisions made regarding methodologies adopted.

1.1 Context and Motivation

Emotions are a big part of the human essence. They have the potential to completely drive our actions and portrait behaviors that model the human society. Its complex nature is somewhat uncertain as multiple mental states can overlap, originating different perspectives regarding each individual (Frijda, 2004). For a relatively long time, a lot of researchers built theories attempting to discretize these mental states. Paul Ekman initially focused on six basic emotions: anger, disgust, fear, happiness, sadness and surprise. He based his conclusions on empirically universally recognized emotions, independent of culture (Shiota and null, 2016). Robert Plutchik proposed a psychoevolutionary classification approach for emotional responses (Plutchik, 2000). He began from the point where he took into account a few basic, primary emotions, anger, fear, sadness, disgust, surprise, anticipation, trust, and joy. From these, different combinations would arise, giving origin to more complex sets of emotions, much like the basic colors and their derivatives. Note that it was not possible to combine just any of the basic emotions as some were proposed to be mutually exclusive.

Even today, many experts might sometimes show difficulties when it comes to identifying what emotion one might be expressing.

Right now, we are witnessing the revolution of deep learning (Sejnowski, 2018). Deep learning keeps proving itself on solving problems of many sorts as long as it is supported by careful planning. With the advancement of hardware (Pan et al., 2018), it's been possible to train bigger and more complex models of deep learning as well as training simple ones much faster. This alone buys a great margin of empirical nature to experiment with many different variants of the algorithm, potentially allowing to achieve more efficient and more accurate results. Note that the hardware was not the only improvement. Many different architectures are used for many different problems in the most diverse areas: finances (Sezer et al., 2019), computer vision (Lecun et al., 1998), medicine (Tilve et al., 2020), stocks (Nabipour et al., 2020), among others.

1.2 Motivation

There is still no solid ground when it comes to describing a concept as complex as an emotion (Landowska, 2019). Many times, we, as humans, find it hard to understand what are we feeling at a certain time during a certain event. There are so many underlying signs on every gesture and action, voluntary or involuntary, that gives up clues to what is going through our minds, even if we are not completely aware of it. It is not the purpose of this dissertation to clear the fog completely of what this mystery is, but to provide us with some concrete signals about our own mental state, as a race. The evolution of Artificial Intelligence (AI) might just provide us that chance (Martins et al., 2018) (Teixeira et al., 2020). The abstraction capability of machine learning algorithms might prove itself as a very important tool to understand specific constructs of the mind, making one of the main features of this project to rely as little as possible on human intervention and as much as possible on the math behind AI. This statement does not intend to diminish work done throughout the years by emotion psychology experts, but to reassure their results and try to complement the knowledge on such a complex field.

Many important applications can arise from this research (Rodrigues et al., 2020) (Gonçalves et al., 2015), making the target viable or at least more affordable, as specialist expenses on this sort of task usually come with high cost, especially if it is meant as an every day use on tasks like Medical diagnostics, recommendation system management, social anomalies, fraud detection, human resources management on emotionally high end tasks (Rodrigues et al., 2012) (Carneiro et al., 2019).

There can be many solutions, where a generic or custom approach can be taken, without any relevant intrusion on a subject's life, but the purpose of this research is to develop a solid methodology towards any of the solutions mentioned, and from there, allow many possible paths towards specific tasks that might require small tweaks to the logistics of the project. The main goals can be summarized to a few points on interest:

- Adoption of a benchmark audio speech dataset
- Development of an adaptive methodology, non specific to any field
- Construction of abstract representations of multiple instances of data
- Classification in relation to predefined different mental states

1.3 Problems and challenges

On the full of pipeline of this work, several challenges can be identified. On the past years, multiple studies have been pointing out primarily to the design of databases or limitations of existing ones (Douglas-Cowie et al., 2003) (Ayadi et al., 2011), as there are several aspects to be taken into account, raising important questions: what emotions will the database cover? Will the emotions be acted or taken out of natural speech? Under what social context will the speech be recorded? Is the sampling quality good enough? Are the emotions expressed clear enough? They are all common sense questions but somewhat crucial decisions with high impact on the final outcome.

When it comes to training and algorithms, the list is much shorter for issues, but can be equally persistent, as optimization, depending on what sort of learning is being taken into account, can be extremely sensitive and hard to master. Exhaustive and systematic analysis can be needed in order to achieve relevant results in a matter, which can be demanding on both theoretical and practical point of view. Combining this with the need of capable hardware to process the data at high speed, a big resource management issue appears, since time is not an unlimited luxury we can afford to have. It is important to tackle this issue as efficiently as possible in order to be able to walk through many experimental cenarios, which is only a natural thing in the field of ML.

Note that problems and limitations explained here do not apply only to the problem of Automatic Speech Emotion Recognition (ASER), both these and the ones related to the methodology will be exposed further on this document, withing a more specific context.

1.4 Research methods

The line of work where emotions become discrete values allowed for discriminative models to appear, and the recognition of emotions by a machine found its base to grow (Rodrigues et al., 2021). The traditional ASER framework divides itself in three components, data collection, feature extraction and classification (Aeluri and Vijayarajan, 2017) (Pathak and Kolhe, 2016).

Data collection on speech has had multiple approaches as well as categories where they fit, regarding the context and restrictions it has (Drakopoulos et al., 2019). But the issue focused here revolves around the expensive task of labelling necessary large amounts of data to be fed into state of the art algorithms. It does not scale to the required proportions of instances to make a reliable ASER system (Singh et al., 2008). So something like SSL could have a big impact on the design of such collection, where costs are severely reduced, making big scale projects viable, and so, more likely to perform ASER in a more naturalistic way.

With today's advances in deep learning, the feature extraction on ASER tasks require less and less human intervention as the neural networks designed can build complex functions that identify important features on data (Khalil et al., 2019). So the need for hand crafted features has been decreasing in favour of the automatically extracted ones. It opens up a bigger margin for training as the time wasted on feature crafting and selection might grow quite large. With automatic feature extraction, it might be possible to try out the most diverse scenarios (Shaheen et al., 2016).

The classification schemes have already been multiple. The more traditional ones consisted on Hidden Markov Model (HMM) (Rabiner and Juang, 1986), Gaussian Mixture Model (GMM) (Reynolds, 2008), Bayesian Networks (Heckerman, 2008) and Support Vector Machine (SVM) (Ben-Hur and Weston, 2010). However, neural networks have been the best choice for the past few years due to their discriminant ability and efficiency.

In this dissertation, mainly Deep Neural Network theory will be exploited for the purpose of SSL (Ouali et al., 2020) (Goodfellow et al., 2016). An explanation on important notions will be provided in order to justify the choices on different crossroads throughout the project.

1.5 Expected results

In the section dedicated to motivations and goals, there were numerated a few milestones crucial for the success of this work, but these mentioned points represent long term objectives (Méndez-Villas, 2005). When it comes to results actually expected from the research, the picture thickens and more abstract ideas come into play. The experiment will revolve around existing structured speech data. It needs to present certain characteristics regarding naturalness and diversity that are expected to provide important traces to the research, leading to viability on real world applications on many different scenarios. In terms of actual emotional content, it can be debated that, depending on the variety and quantity of samples, not all emotions will be captured. In fact Ayadi et al. (2011) refers that something described as a neutral emotion will compose most of the speech corpora around. Nevertheless, it is expected to find some sort of complex structure extrapolated from features of the data. From such a structure, a SSL framework can follow in the pipeline. With this project path, the purpose is to look to reduce the reliability on human labeling to the least as possible, considering that one of the base points is that there is no ground-truth when it comes to assemble classified instances on ASER. Ultimately, the purpose lies around providing clues to a subject's mental state.

1.6 Bibliographic Review Process

The elaboration of the conceptual framework needs to follow certain review approaches in order to achieve the overall quality and viability demanded. To do so, it is necessary to research the core topics of this dissertation as well as the works related to its subject.

Some of the viability surging on this dissertations matter rests on the quality of the resources consulted. This puts a lot of weight into the research part of the project, where a lot of information, concepts and hypothesis are formulated in order to actually move forward. It's crucial that correct and well founded notations are acquired during the research phase as it will translate a major part of the the whole working framework itself.

For a rising field such as ASER, relatively recent papers, meaning around five to seven years old, are given higher importance, as over the more recent years, a lot of breakthroughs have been made in the area. As the context of this work gets more specific, the number of citations is often not given that much weight compared to somewhat standard circumstances.

Keywords like “Semi Supervised Learning", "Speech Emotion Recognition", “Convolutional Neural Networks", and "Audio Signal Features” were mostly use during the background acquiring phase, across multiple platforms:

- Google Scholar;
- IEEE Xplore;
- ResearchGate;
- RepositóriUM;
- Science Direct;

Between audio processing and neural networks, the range of papers is fairly big, until a more specific task is developed.

1.7 Document Structure

To provide a better understanding of this dissertation’s subject and benchmarking approach , the document was organised into six chapters:

- **Chapter 1:** Brief contextualization and description of base line for the research, as well as motivation and expected results;
- **Chapter 2:** Description and explanation of important concepts and notions in the context of the dissertation as well as some overall contextualization regarding the topic of ML. A view of possible approaches, as well as a conceptual framework description and a literature review on the state of the art results.
- **Chapter 3:** Review of the setup to be utilized to design the experiments, follow by its specifications. The procedure is described and the results are exposed to understand the impact of the concepts applied along the research.

- **Chapter 4:** A final analysis of the results is done in order to reflect upon utilized measures and the possible evolution/improvements to the current methodology and frameworks;
- **Chapter 5:** Identification of main problems and limitations to establish a concrete improvement point for future reference. Description of the landscape on the technological framework utilized.

2

STATE OF THE ART

For quite sometime, AI has been related to aspects regarding the function of the human mind. Since the 1940s, names like McCulloch and Pitts (McCulloch and Pitts, 1988) and Turing (Turing, 1950) were important on the design of theories on similarities between computers and the human brain. The term "Artifical Intelligence" was coined at the Dartmouth meeting (Mccarthy et al., 2006). Since then, a lot of different areas have surged from the field, merging from branches all around. Machine learning can be seen as one of these branches. Beverly Park Woolf (Woolf, 2008) utters that Machine learning refers to a system's ability to acquire and integrate knowledge through large-scale observations. and to improve and extend itself by learning new knowledge rather than by being programmed with that knowledge. Most instances of any type of information can convey patterns and features that can be hard for the human mind to perceive. Our huge capacity of abstraction is put at use on our everyday routine, but for intrinsically complex problems involving many numerical dimensions, the human mind generally falls short. Thus, there was a need to develop ways to solve these kind of problems where non explicit features would be displayed on relatively big amounts of data.

An important step came with the design of neural network theory (Rosenblatt, 1958), the perceptron, capable of creating proper boundaries on a linearly separable dataset. The concept ended up being dropped for some time due to its lack of capacity for solving relatively simple problems like the infamous XOR problem, where non-linearity was imposed in a very straightforward way. Later on, at 1986, back propagation was introduced by Rumelhart et al. (1988), but the at the time computational resources did not allow for too many experiments or conclusions, so no systematic way of training was defined. But then, around 2006, Hinton et al. (2006) showed that it was viable to train neural networks to learn certain patterns on non-linear data.

Many algorithms were made on treatment and comparison of data features, based on what we can describe as Paradigms of Learning, where four common ones will be distinguished: Supervised Learning, Unsupervised Learning, Semi Supervised Learning and Reinforcement Learning.

2.0.1 Supervised Learning

A lot of ML algorithms are built around SL, as it looks for direct correlations between input and desired label and builds a function on the network around it. Liu and Wu (2012) defines it as a machine learning paradigm for acquiring the input-output relationship information of a system based on a given set of paired input-output training samples. Thus, there is a discriminative component to it, that allows important tasks such as classification between several discrete labels and multiple features. Note that regression is also possible as continuous variables are supported for a lot of algorithms that follow such paradigm. However, the necessity for predefined labels raise an important issue related to costs of labelling. It might not be viable at all to classify thousands of data instances, especially as in many tasks, the aid of a paid specialist might be required, as the precision of labelling is something that can have a huge impact on the reliability of the model designed. Reckless labelling may lead to drop in performance. Other important issues go around ambiguity in instances and/or labels, as the boundaries created can get affected depending on the number of perturbed instances in a dataset.

2.0.2 Unsupervised Learning

On same cases, where labels might not be present or perhaps necessary, Unsupervised Learning (UL) makes itself as a feature detector, completely intrinsic to the data in hand. Unsupervised learning algorithms are used to group cases based on similar attributes, or naturally occurring trends, patterns, or relationships in the data (Larose and Larose, 2015). It is interesting to note that these kind of algorithms are not building mapping functions from input to output, at least not as directly as SL does. This can mean that when humans can not identify similarities or dissimilarity on certain types of data, there is potential to give it some discriminant structure with UL. However, the efficiency and actual practical applications regarding this learning paradigm can be rather limited overall.

2.0.3 Semi Supervised Learning

SSL represents a special case, where it tries to combine the best out of the two previously described paradigms. It is meant as a way of dealing with a context where there is a big amount of unlabelled data available and labelled data presents itself as a costly, expensive and time consuming task (Ouali et al., 2020). The idea is to use the unlabelled data to provide valuable information on training of models. This can open up possibilities for large scale projects, where big amounts of data are collected and stored in bulk. With an SSL approach, the purpose is to capitalize on the abundance of unlabelled data, as well as dimming down the dependence on labelled instances. In recent years, many SSL approaches have been showing up towards deep learning (Yang et al., 2021). This factor can however present itself as a two edge situation: its novelty brings up many issues on standardizing training techniques, meaning there are still a lot of choices and steps that need further testing/development.

2.0.4 Reinforcement Learning

Reinforcement Learning (RL) represents a problem much more generalized than Supervised Learning. An entity described as an agent interacts with an imposed environment where it need to find an optimal behavioral strategy based on limited feedback from the environment itself. This feedback comes in the form of a reward or a punishment which will affect on how the agent will try and adapt to the circumstances. The way the adaptation is made is ruled and modelled by the RL algorithms, where the goal is to find a policy that maximizes the long-term reward (Heidrich-Meisner et al., 2007). Functionality wise, the concepts of agent and environment face limitations. Both employment of physical or virtual agent and space can be expensive and it highly depends on the task in hand. The nature of RL can be linked to neuronal and behavioral sciences on incorporating biological concepts on the learning. It is still an area in development and there are very high expectations as researchers have been achieving considerable results on such paradigm (Mousavi et al., 2018). The ability to learn on almost non-existent predefined data can open many doors on behavioral adaptation research.

2.1 important ML bases and notions

In this section, important definitions and mechanics on technical notions will be displayed and fit on the context of this paper. It's important to note that not every aspect of a certain field or area of Machine Learning will be explored deeply. The goal is to point out crucial knowledge on understanding and justifying the choices made throughout the project.

2.1.1 Classification problems

Classification problems in the field of ML occupy a big portion of the utility of models designed on such guidelines. There are several applications for ML algorithms but one of the most relevant ones is predictive data mining (Kotsiantis et al., 2006). Datasets consisting of features, categorical, continuous or binary, can make a supervised problem if there are known labels to those instances or an unsupervised one if there aren't any. What's incredibly interesting is that classification problems can be formulated on the most diverse situations, towards the most specific tasks. Its versatility is what brought it to the highlight on many humanly intractable mathematical problems. Even more important is that many results obtained can already surpass human performance (He et al., 2015).

2.1.2 Neural Network theory

Many classifiers have been developed on very different contexts and tasks. Linear and logistic regression (Gabriel, 2018) (Maalouf, 2011) are baselines ones to solve linearly separable data problems with efficiency. SVM (Ben-Hur and Weston, 2010) introduce more complex operations on data, making them capable of solving problems on non linearly separable data with mostly good results.

Today, the major highlight on ML is the Artificial Neural Network (ANN) theory (Grossi and Buscema, 2008). ANN are graph composed systems, inspired on processes of the human brain (McCulloch and Pitts, 1988). They have a characteristic adaptive capacity to change their internal configuration to fit certain aspects of data towards a label. When the data associated with a problem show great degrees of complexity, ANN tend to dominate over any other algorithm (Çoban, 2016). This is due to many factors such as flexibility of models and dimension scaling of data (LeCun et al., 2015).

The base unit of an ANN is a neuron, represented on the graph notation as a node. Each neuron receives an input from the environment or other neurons affected by the strength or weakness of the connection (weighted sum). It then performs an operation involving an activation function on the input received. These specific functions exists mostly on each neuron. Their use confers a non linearity factor to the network, creating non linear relationships between the weights and data (Olgac and Karlik, 2011), which is very important given the complex high dimensional structure that data coming from every other scenario presents.

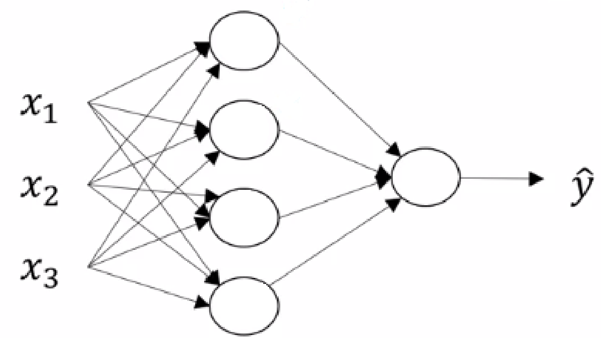

Figure 1: Example on schematics of a simple shallow (one hidden layer only) neural network.

These components work on a feedforward manner where they learn to map some input reproduced on some source to a certain output class, depending on context. The network can be divided on three different areas: input layer, group of neurons that take the data directly without change; one or more hidden layers, where each layer is composed by a group of neurons; output layer, where the final calculation of the function learnt is done, the number of neurons will vary with the task in hand (like classification (Kotsiantis et al., 2006), multitask learning (Caruana, 1997)). It's the changes on the weights of each neuron that will look to make the network learn the structure on the data. These changes are managed through the algorithm of backpropagation (Rumelhart et al., 1988). Backpropagation looks to compute the gradients of a predefined objective function (Wang et al., 2020) with respect to the weights.

The key insight is that the derivative (or gradient) of the objective with respect to the input of a module can be computed by working backwards from the gradient with respect to the output of that module (or the input of the subsequent module). The backpropagation equation can be applied repeatedly to propagate gradients through all modules, starting from the output at the top (where the network produces its prediction) all the way to the bottom (where the external input is fed). Once these gradients have been computed, it is straightforward to compute the changes with respect to the weights of each module.

From this explanation on the concept of backpropagation cited from LeCun et al. (2015), it's important to understand that this chain rule modifies each layer according to their error signal as a whole. So the process can be seen as build up of properties, formed in a hierarchy, growing on complexity and abstraction, where the association between raw data and a label is learned on the function expressed from the combination of the neurons. Backpropagation is used in an iterative way for proper tuning of weights on the network. This concept of training came in very importantly for the development of Neural Networks, as it launched important concepts on forming what Deep Learning (DL) is today (LeCun et al., 2015).

2.1.3 Deep learning

Among the ML guidelines, there is something that goes by Representation Learning, where multiple processes and computations fit each other in order to learn relevant representations of raw data, allowing for detection and classification (LeCun et al., 2015). DL builds on this concept, involving the ANN theory by creating multiple representations on a characteristic hierarchy set up where the level of abstraction increases gradually up to a defined goal. It can be seen as the stacking of multiple modules, composed of layers containing non linear units (neurons), where, with enough density, complex functions can be learnt. On the past years, it has been verified that deep learning based machines are able to find structure on high dimensional data on many different contexts. Automatic feature selection/extraction combined with considerable results provides valuable advantages over every other ML algorithm. Today, there are still many variants over models, architectures and algorithms being developed so many improvements can be expected on a near future.

2.1.4 Convolutional Neural Networks

Convolutional neural networks have been the primary choice for most tasks on computer vision (Shamsaldin et al., 2019). They are an architecture derived directly from the cerebral visual system and look to emulate some aspects of it. The inspiration source can be dated back till 1962, with the work of Hubel and Wiesel where they concluded on the sensitiveness of the cells in the cortex towards small sub-regions of the visual field, called receptive fields. LeCun et al. (2015) attributes the roots on Convolutional Neural Network (CNN) to the Neocognitron, a model proposed by Fukushima in the early 80's. Its first practical application came with Cun et al. (1990) to recognize hand-written digits. In the past years, the use of Graphics Processing Unit (GPU) on ML promoted CNNs as this hardware technology allowed for a relatively efficient way of training this type of networks.

The CNN architecture is usually composed around convolutional layers, pooling layers, flattening layers and finally fully connected units.

Convolutional layers are the core aspect of the network when it comes to feature extraction. Each of these layers are made of groups of filters/kernels with a fixed area of effect that move through the input data, creating one feature map per filter. Each of these filters have adaptable parameters that will be shared across the whole space of an instance, providing efficiency to the computations done. This way, different filters capture different properties of data, learning structure for complex datasets. One important advantage relatively to Fully Connected layers is that CNNs have the ability to obtain shift invariance automatically (Lee et al., 2020), providing some unique robustness to these systems. Fully Connected layers on the other hand, on tasks like Automatic Speech Recognition (ASR), where a lot of parameters and calculations are involved, tend to overfit somewhat easily. To the output of the filters, on each section of data, an activation function is still employed to confer the non linearity factor to the net.

Pooling layers address some important issues regarding efficiency as well as performance in training. They have the ability to down sample data in a more drastic way, where each section of an instance, depending on the pooling kernel size, will be measured and contracted into one single value, depending on type of pooling (for example Max Pooling or Average Pooling). This way, less parameters will be needed for posterior processing.

The flattening layer complements the transition from Convolutional layers to Fully Connected layers. The output of the CNN components are usually a three dimensional matrix, composed of multiple feature maps produced by previous kernels. So there is a need to adapt the structure of these to fit the input shape required by Fully Connected layers. This change of shape is conducted by the Flattening layer that prepares the input to the final transformations leading to the output of the network.

As said previously, CNNs have been applied extensively on Computer Vision. Given the way it's birth circumstances, it would only be a natural progression step. Lately though, their usage has expanded to further areas like the immense Natural Language Processing (NLP) field, involving tasks like ASR, ASER, sentiment analysis, text processing, among others. Given the context on which this dissertation paper revolves around, the extension of such models towards ASER provides a platform of exploration on multiple approaches.

2.1.5 Recurrent Neural Networks

Recurrent Neural Network (RNN) is a characteristic type of architecture capable of learning features on long term dependencies, augmented by the inclusion of recurrent edges acting over consecutive time steps, adding the time dimension to the model (Lipton, 2015). Properly tuned models following the RNN path are capable of handling any dynamic system (Salehinejad et al., 2017).

RNN alike algorithms can be dated back till 1982 where Hopfield (Hopfield, 1988) introduced a family of recurrent nets with some capacity of pattern recognition over time. However, they held no clear way of being trained on a supervised manner. After some refinement and progress with backpropagation and ANN theory during the early 90's, Bengio et al. (1994) introduced the notion on issues regarding training RNNs when the complexity of data, namely degree of long term dependencies, increased and were described as vanishing gradients. This phenom happened when the propagation of error signal among modules would die out and produce very little change on the weights to the point where earlier modules would not be able to learn any features on data causing a cascading decline effect throughout the network. Later, in 1997, with the introduction of Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) it was possible to successfully train RNNs in a more systematic way. The concept of memory along time steps is introduced as a way of keeping track of important features in sequential data, helping to deal with the problem created with the gradient training on long term dependencies. Today, LSTMs are widely used on many areas where sequence translated or time dependent data is used. The main application revolves around NLP areas, such as text translation, ASR, sentiment analysis, ASER, among others.

For quite some time now, a debate has surged on the usage of RNNs or CNNs on ASER tasks. The lack of proper benchmarking makes it confusing when it comes to determine a valid conclusion, as multiple different approaches on different set ups (datasets, features, network configuration) are being conducted and obtaining different results. The use of LSTMs seems like the natural choice to process data in format of audio, where time is implicit on such volatile concepts as emotions (Ayadi et al., 2011). But CNNs have been picked up from the Computer Vision area because the frame level features created for this type of network don't require as much preprocessing and the algorithm itself is computationally efficient, note to mention that competitive results against previous RNN approaches are being obtained (Abdul Qayyum et al., 2019). This creates room for further diversified approaches that will require proper documenting for future researchers.

2.1.6 Autoencoders

A very interesting variation of the ANN theory is the concept of Autoencoder (AE): graph systems specifically built to encode some input into a compressed relevant representation and the ability to decode it back to a state as similar as possible as the original input. The way the system is designed allows fitting an unsupervised learning paradigm, conveying multiple advantages when it comes to usability of data (Bank et al., 2020) (Hinton and Salakhutdinov, 2006).

Standard AEs can be decomposed on three parts, encoder, code and decoder (Maheshwari, 2020): the encoder is responsible for receiving the input and conducting a series of operations depending on the number of hidden layers, down to a lower dimension than the input itself (generally), where this dimension is defined by the number of hidden units at the end of the encoder; the code corresponds to the compilation of features comprised on the input, transformed by the encoder, where they can be described as a latent space and are used as input to the decoder; the decoder will take the coded data and revert it back to the initial dimensions, as similar as possible to the original input.

Mathematically, they are usually composed by activation functions to confer the non linear factor to the system. Linear AEs are also a concept where they have the tendency to behave just like the dimensionality reduction algorithm of Principal Component Analysis (PCA) (Bro and Smilde, 2014). Since the formation of the concept of AE at Rumelhart and McClelland (1987), many different variations were developed. Denoising Autoencoder (DAE) is a type of AE, designed to be able to reestablish an input from a corrupted version of itself, by capturing statistical dependencies between input/output. This allows for the model to have considerable resistance to perturbations on instances of data. Another example is the notion of Variational Autoencoder (VAE), fitting the Generative Models category (Guzel Turhan and Bilge, 2018), where models attempt to describe data generation through a probabilistic distribution, allowing for the use of adaptive parameters representing some latent space to generate data. Another variation is the Sparse Autoencoder (SAE) where the dimensionality of data is increased by using more hidden units per layer than input features, where sparsity is introduced by constraints in the loss function, allowing the autoencoder to find distinctive patterns within the dataset.

AEs unsupervised nature allow for many creative uses. They have a powerful ability to extract relevant features on the most complex data structure.

2.1.7 Generative models

Recently, within the Deep Learning community, there has been a trend towards the notion of Generative Model, mainly GAN (Goodfellow et al., 2014) (Salimans et al., 2016) (Ruthotto and Haber, 2021) . There have been other generative models before but they came with a lot of issues regarding complexity and computational cost, causing them to stay out of the spotlight (Guzel Turhan and Bilge, 2018). GANs offer the ability to generate data realistic adversarial examples as well as a good ANN based classifier. The potential on content creation out of many different data distributions, depending on context, presents itself as a window for many high end tasks, like data augmentation, object generation, image manipulation, among others.

Usually, GANs are composed of two different networks, denominated as Generator and Discriminator, trained separately but simultaneously. The Generator takes a randomized input and tries to map it to data space, representing the training process on the generator's side. The Discriminator takes some input and classifies it by outputting a probability close to 1 if it belongs to the real data distribution or close o 0 if it belongs to the generator's distribution. This means that training on both networks will create a Generator capable of understanding the data distribution and creating new instances, and a Discriminator specialized on feature extraction/selection on the data. From this vanilla set up, many variations have been designed. Radford et al. (2016) talks about the use of Convolutional layers in the GAN setup and leverages the Discriminator to classify instances.

GANs show promising results and the margin of improvement is stretching far away into the horizon (Odena, 2016) , but its complexity and computational cost might be a breaking factor when it comes to choice of algorithm, as there are many factors about GAN that aren't quite standardized yet as well as other intrinsic problems regarding its performance, training and design (Saxena and Cao, 2020). VAEs are also in the category of Generative Models, as previously mentioned. Because of the way the model is constructed, it is possible to sample from the latent space created by the encoder and delivering newly generated data directly from the learnt distribution. The use of VAEs, compared to that of GANs, show a lower degree of complexity as the amount of hyper parameters to test from is smaller and training itself can be more stable, although mostly showing lower quality results (Munjal et al., 2019). This might be important on research where the focus is not on the algorithm itself but on the pipeline designed for the task, although it would be essential to keep this kind of trade off in mind.

2.2 Signal Processing

According to the Acoustic Society of America (ASA), sound is defined as (a) Oscillation in pressure, stress, particle displacement, particle velocity, etc., propagated in a medium with internal forces (e.g., elastic or viscous), or the superposition of such propagated oscillation. (b) Auditory sensation evoked by the oscillation described in (a). Sound can be described by a waveform, which is characterized by factors such as frequency, amplitude. Taking periodic sound as an example, frequency keeps track of the number of cycles a wave has per second which translate to the height of the sound: higher frequency means higher sound. Amplitude refers the magnitude of the wave, the measure of its change in a single period: larger amplitude means louder sound.

Within the context of ML, the concern lies around the conversion of the analog signal to a digital one, followed by some sort of preprocessing to get it ready to be used to train some algorithm on a determined task. An analog signal has a continuous nature where amplitude and frequency values present themselves as continuous values. This causes issues with digital oriented hardware, where discrete values are stored. Theoretically, we would need an infinite amount of digital memory to save any continuous piece of data.

In order to fit in the data, current technology resorts to the process of sampling. Sampling goes around discretizing the values of a waveform, where two parameters are highlighted, the sampling rate and bit depth. The sampling rate models itself around period (inverse of frequency), where given a period T, one sample of the waveform will be saved every time T has passed. The higher the sampling rate, the lower the error rate. For example, common CDs are sampled at a rate of 44100Hz. Why? The Nyquist-Shannon theory (Weik, 2001) states that an analog signal waveform may be uniquely and precisely reconstructed from samples taken of the waveform at equal time intervals, provided the sampling rate is equal to, or greater than, twice the highest significant frequency in the analog signal. Otherwise, the phenom of aliasing might occur, where if the signal was to be reproduced from its digital conversion, artifacts would surge corrupting the original data. Humans can capture sounds in the range of 20-20000 Hz based on the pressure applied on eardrums (Darji, 2017). So half of the Sampling rate on CDs equals 22050 Hz, standing just above the range of human ear range. Another important aspect of sampling is bit depth, where the number of bits used for sampling the amplitude of a signal is defined. A predefined number of bits will be equally spaced across the amplitude dimension, forming levels. At a sampling point, the amplitude value will be set on the closest bit value. This implies that the bigger the bit depth, the more precise will be the value set on the sampling process. For reference, CDs bit depth goes around 16 bits, resulting up to the value of 65536.

The aspects described previously can be important for this research as the quality of digital signal can and will have an impact on any algorithm's performance. On the other hand, better signal quality will ask for more competent hardware, imposing restrictions on affordable usage. It´s important to keep in mind this quality trade off, where for higher portability (for common users with no task-specific hardware), lower quality signal will most likely follow. This adds another issue where, on such stance, a more robust and efficient model is required, potentially creating more challenges on the development of models. For the use of high quality hardware, where portability usually falls short in terms of viability, the signal quality will be higher. This raises extra questions, namely regarding the actual performance of the model: will the model achieve the expected performance with high level quality of samples? The field of ASER is still evolving and walking towards a point where it can be used reliably on social interactions but still rather far away from high end/high risk scenarios.

2.3 Conceptual Framework

For this dissertation's focus, a generative model will be adapted to a classification task of emotion recognition on audio speech data. More concretely, by consuming data from multiple sources, close to an end-to-end approach, a GAN variant will be employed and trained from scratch, which can be more commonly known as SGAN (Odena, 2016), on the attempt of achieving good levels of generalization based on the combination of labelled and unlabelled data. The goal here is to investigate on an efficient an scalable approach to the problem of ASER, in order to push the paradigm forward and encourage further testing, since SSL does show a lot of potential on the strength of being able to handle partially raw data.

In an era where information is everywhere, it can be seen as an absolute priority to learn to leverage data that requires little to no human intervention on processing as it presents a huge fraction of the resources in the world for almost any task at hand. Deep learning methods are being developed on that sense and the author believes it presents itself as a unique opportunity to combine the power of Neural Networks with Big Data.

2.4 Literature Review

Emotion Recognition, as a concept, has been around for quite some time, which by itself implies many different approaches of distinct nature. From a generalized perspective, it's relevant to make the distinction between multi modal and single modal emotion recognition. Multi modal usually has more than one source of information: voice, facial expression, body language, among other auxiliary types data. Single modal refers to only one type of data, which applies on this dissertation. As speech data only is used, it should be only fair to make a comparison with other single modal focused work, mainly speech.

As fully supervised approaches are still the most relevant, compared to SSL, a simplified overview of supervised approaches is also necessary to understand the standards of classification. However, the comparison of actual results can be tricky, as different forms of sampling from the dataset, different types of classification and different evaluation metrics arise from a purpose established by each researcher, which leads to a variety of parameters, making a direct comparison not as a reliable as it normally should be.

In Xu, Zhang, Cui and Zhang (2021), a relatively small Convolutional neural network, parameter wise, is employed with feature fusion and multiscale area attention, extending on Li et al. (2020), using Log Mel Spectrograms, achieving an incredible 79.34% Weighed Accuracy (WA) and 77.54% UA on the IEMOCAP. Only a specific portion of the IEMOCAP is used, meaning, four classes, sad, neutral, angry and happiness+excitement. Posterior filtering is done to the data: the dataset is divided in two parts, an acted portion and an improvised portion; only the improvised portion is used. An ablation test is also conducted to verify the impact of each enhancing technique applied on the model, like the attention variation or Vocal Tract Length Perturbation (VLTP) for data augmentation, allowing to verify the improvement made by the implementation choices.

In Jalal et al. (2020), two groups of experiments are conducted, which can be mainly distinguished by the neural network type, Bi Long Short Term Memory with attention (BLSTMATT), described by Milner et al. (2019), and Convolutional Self-Attention (CSA), described by Jalal et al. (2019), both with higher complexity that the previous work mentioned. On the standard benchmark IEMOCAP, with the BLSTMATT approach, a value of 80.1% UA and 73.5% WA are obtained, while with the CSA approach, the best values go up to 76.3% UA and 69.4% WA. Finally, by combining both approaches into one, where a manipulation over the outputs of each network is conducted, a state of the art value of 80.5% UA combined with 74% WA is obtained. Just like previously mentioned on other work, four classes are used, sad, neutral, angry and happiness+excitement, but on this setup, both improvised and acted speech samples are used. The dataset is separated into train/test in a speaker-independent manner, where the actors on one set are different from the actors on the other, making it even more impressive than it already is.

Fully supervised approaches are hitting higher and higher degrees of performance and slowly evolving towards what could be highly reliable systems. Of course that the reliability of such system depends on the final goal of the task.

On some specific tasks, the gap between a fully supervised paradigm and semi supervised paradigm is not black and white anymore (Andrade et al., 2022). So SSL is presenting itself as a more viable solution to the future, as the amount of data collected is larger and larger, and its processing is more and more expensive. To take advantage of the abundant unlabelled audio data, many researchers found original approaches on the SSL paradigm. In Zhao et al. (2020), a GAN system is employed along a classifier to add some meaningful information from unlabelled and generated data to the labelled set. To complement the system, the researches chose to implement some techniques originated on Adversarial Training (Goodfellow et al., 2015) and Virtual Adversarial Training (Miyato et al., 2018), implementing a Smooth Semi Supervised Generative Adversarial Network (SSSGAN) and a Virtual Smooth Semi Supervised Generative Adversarial Network (VSSSGAN), respectively. Note that the features used to represent speech are not all image-like, which might drive the algorithm implementation away from stable architectures based on Deep Convolutional Generative Adversarial Network (DCGAN) (Radford et al., 2016). However, considerably good results are still achieved on the IEMOCAP (also four class set up, improvised and acted speech included), with 59.3% and 58.7% UAR, at 2400 labelled data, on the proposed methods of SSSGAN and VSSSGAN, respectively, claiming to outperform the state of the art under this context. This is most likely one of the few pieces of work done on SGANs, on a single modal baseline with speech-like only data, which translates the lack of actual practical resources for the task proposed in this dissertation.

3

TECHNOLOGICAL FRAMEWORKS

The development of any virtual system requires technical support from one or multiple frameworks. On this section, the whole conceptual design of the pipeline will be described.

A very important part of the model development phase starts around the data preparation and processing. In this pipeline, the python library Librosa (McFee et al., 2021), a comprehensive python library for audio processing and analysis, is a perfect match to fit on the task. With the assistance of other important core data processing packages like Pandas (pandas development team, 2020) (Wes McKinney, 2010) and Numpy (Harris et al., 2020), the data is prepared from its raw state into features ready to be fed onto the Neural Network system.

For the model development itself, Tensorflow (Abadi et al., 2015) will be selected. Tensorflow is a machine learning system that enables many different kinds of experiments and novel training procedures and has been widely used on many Artificial Intelligence tasks. Its flexibility provides an ideal environment to customize and control any experiment from end to end. With Keras back end included from Tensorflow 2.0 and up, a lot of simplicity is added to the whole framework, allowing for a somewhat straightforward build of a system.

3.1 Dataset

Nowadays, audio data is rather abundant in the digital world, whether it is music, conversational speech or any other kind. Classification tasks on ML still need to get some processed, labelled data to make it reliable. For ASER, a lot of different collections of datasets have been made throughout the years (Wani et al., 2021) (Abbaschian et al., 2021). Three different types of collected datasets can be distinguished: Spontaneous Speech, Acted Speech and Elicited Speech.

Spontaneous speech is characterized by its naturalness and expression of unbiased emotions towards emotion caption. There is usually no direct intervention with the speaker, as the data is collected from external platforms like interviews, podcasts and many other sources. This type of natural speech might prove itself to be very important when it comes to real world scenarios due to its genuineness, as it might approximate the abstract representation of emotions on neural networks better than any other type. As such, these characteristics can bring subtle, more complex, harder to learn features that hinder or refine learning, depending on the system's capacity. There is also the issue of distribution and annotation of data, where both of these factors show itself to be very volatile, meaning that, depending on the source, there is a high chance that the amount of data collected from that source, under some predetermined set of labels, is not going to be balanced. Even worse, the annotation process itself can be rather subjective and prone to error, even though many efforts are made to preserve the viability of both discrete and continuous labels on emotional data (Lotfian and Busso, 2019).

Acted speech has the big advantage that the experiment's factors can be fully controlled. Aspects like the speaker, the emotion to be expressed, the wording, can all be predetermined beforehand, allowing to create the perfect environment for restricted events. The distribution of instances per emotion can be fully accounted for, which tackles a big issue on emotional databases. On the other hand, the nature of acted speech can interfere with the expression of emotions itself. On many instances, professional actors are employed to follow scripts in order to provide some degree of authenticity to the desired emotion. Also, the annotation process is much simpler, as the conditions are all previously set up, leaving the issue to the quality of the performance of the actor, instead of the quality of the annotation as in Spontaneous Speech.

Elicited speech is a particular strategy of speech data generation that aims to reunite a few advantages from both spontaneous and acted speech. Certain situations are manufactured in order to induce an emotional response from a subject. There are some formalities that present themselves as issues, like the fact that if a subject knows that the reaction is being recorded, the authenticity of the emotional response might be compromised. There is still a lot of control on the experiment, allowing for some variation on the emotion pretended, but it still presents a big overhead of set up, which by itself, restricts the distribution of the data.

There are other important aspects on the design of such databases. Language, for example, can dictate the amount of data that can be obtainable from a certain source, i.e. if the criteria shows english language, then there is a big chance that a scientist will be able to collect huge amounts of data on the internet without many costs associated. If the language would be portuguese, then it could be more difficult to find appropriate data for the task. Next, the annotation scheme combined with the label set. The labels on an emotional dataset need extra attention. As stated before, there is no solid ground for emotions, especially when it comes to describing it in a discriminating way. The boundaries between them are not clear and should not be treated as such. The dimensional values of valence, arousal and dominance can counter this issue, but it brings something else along: there is no precise way of measuring the values of these dimensional labels, which might bias the whole dataset towards the annotators' point of view. The problem that both of these approaches have in common is that they can both compromise the quality of a database from the point of the absolute ground truth that labels represent (on general case, there are no ambiguities on object recognition, for example), originating from the conveniences made. They have their value towards improving the machine-human interaction, but the knowledge available right now is not enough to employ such models, trained on such datasets, to high end or high risk scenarios, like medical applications.

However, the work done on providing audio speech data has allowed for efforts to be made on improving the conditioning of an ML problem, which by itself can allow scientists to establish proper frameworks that will grow along the research on human emotions.

For the purpose of this dissertation, the IEMOCAP (Busso et al., 2008) will be the main source of data. This dataset is characterized by its high quality, high amount of acted and elicited speech. From the perspective of real world applications, this can actually come up as an issue, due to the actual diversity on how the audio data reaches the model.

3.1.1 Data exploration

The IEMOCAP, announced in 2008, is a standard benchmark for ASER tasks. It comprehends around 12 hours worth of speech, recorded at 16KHz (sampling rate), divided in five parts, where each part holds two different actors, gathered at the Drama Department of University of Southern California. Along with this, there is another division between scripted and improvised data. The dataset comprises nine different types of emotions, anger, excitement, fear, frustration, happiness, neutral, sadness, surprise and other. The labelling process is built on a set of annotators where it relies on the agreement of more than half of them in order to decide on the emotion expressed. This type of annotation procedure led to an imbalance in the number of labels per class, as the situations where a certain emotion would be expressed weren't designed by label but annotated case by case. For this reason, researchers chose to proceed with four emotions out of the set, neutral, anger, sad and happy, where due to the low number of "happy" labelled instances and the similarity of it compared to "excitement", these two would get merged (Xu, Zhang and Zhang, 2021). For the sake of proper comparison, both the scripted and improvised part of the dataset will be utilized. In the end, the effectively used dataset is composed by 5531 utterances: 1103 angry labelled instances, 1636 happy labelled instances, 1708 neutral labelled instances and 1084 sad labelled instances, with the percentages on the data illustrated at Figure 2.

This truncated dataset presents a fairly high variety of utterance length, ranging from 0.58 seconds to 34.14 seconds, which can and does present itself as a challenge from the perspective of a due pipeline for the processing of the data. This sample duration range also implies other problems like the presence of one or more emotions throughout the same sentence on one side and the lack of of enough information over small segments on the other. Depending on the approach chosen, it can force researchers into non optimal paths, especially when the task in hand involves DL, as the treatment of data is a crucial step on the makings of a good DL system. To deal with variable length input such as audio, RNNs alike algorithms come to mind where they could be combined with CNNs to take advantage of its efficiency on hardware, which has been a frequent approach in the past (. and Kwon, 2020) (Kurpukdee et al., 2017). From Fig.3,

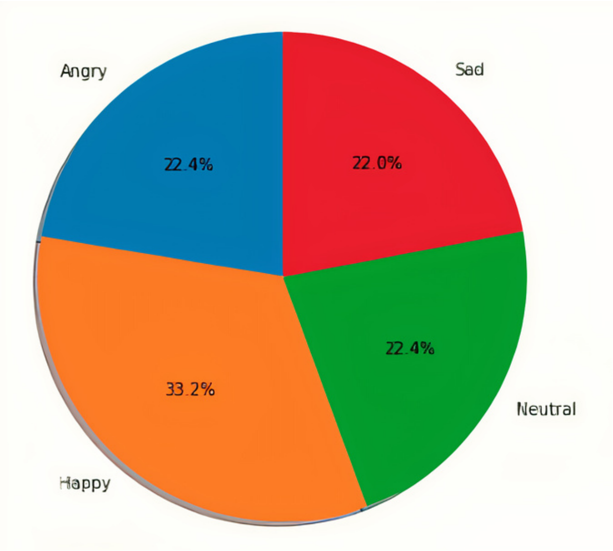

Figure 2: Distribution of the four selected labels over the dataset. .

it can be verified that the distribution of the duration of each samples is rather skewed positively, which means that relatively low duration samples overtake most of the data.

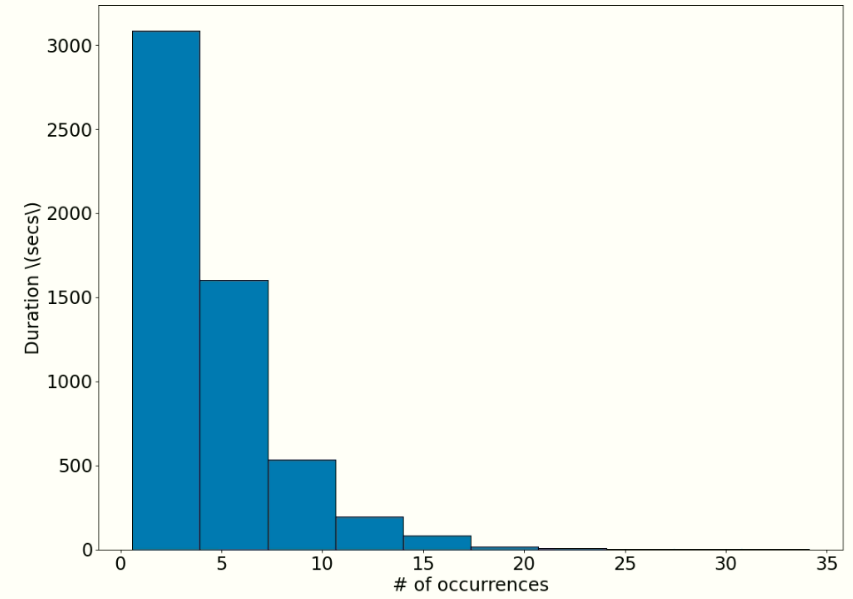

Figure 3: Distribution of the four selected labels over the dataset. .

However, the problem would potentially shift towards the stability of a GAN system that could handle such a model. GANs present a fine line on the optimization process (Lucic et al., 2018), where small changes can lead to drastic performance changes, meaning that the more complex a model is, the harder to control the experiment will be. Each choice of data processing heavily influences the model algorithm itself (and vice-versa), which means that on this set up, special consideration needs to be taken for the final goal, because the viability of the project itself might get compromised.

3.2 Feature settings

The process of audio data takes a fundamental spot on the performance of any type of audio processing system overall. The type of features extracted, whether handcrafted or automatically extracted, will have a huge impact on the viability of the whole process.

As input to Neural Networks, the possibilities range from the raw wave form to some high level transformations of the data, according to certain principles (Latif et al., 2021). A standard approach consists of starting by extracting features known as Low Level Descriptors (LLDs), comprehending loudness, jitter, shimmer, spectral flux, spectral slope, Mel Frequency Cepstral Coefficients (MFCCs), Log Mel Spectrograms, fundamental frequency and others. These can be referred to also as frame-level features, as they are extracted from fixed size frames of audio data. They are then processed again into High Level Descriptors (HLDs) comprehending arithmetic and geometric means, standard deviations, peak to peak distances and others. These are considered segment-level features as they compressed variable sized feature vectors into fixed size ones, relieving and important issue on the processing of variable length data as volatile as audio (Parthasarathy and Busso, 2019). With the recent development of DL though, Log Mel Spectrograms are becoming one of the most popular features in the research community (Meng et al., 2019). Its image-like nature is allowing for the usage of CNNs as models for efficient representation learning and overall relatively good performance.

3.2.1 Feature Selection

For this instance of work, Log Mel Spectrograms will be selected as an only feature, providing a somewhat end-to-end like approach (Glasmachers, 2017) to the issue, which is only made possible under these circumstances by the development of DL. End-to-end systems present many advantages over traditional ML systems. It has also been verified that end-to-end systems can generate very powerful predictors by sheer capacity of a model (Glasmachers, 2017). More importantly, the need for feature engineering fades away, saving considerable empirical time on handcrafting and selecting features which, until the development of DL, was very prominent and important on the development of ML systems. However, there could be some problems in the context of the work done arising from the limitations of stochastic gradient descent and the learning signals provided during the training phase towards the milestones set. Also, very importantly, the problem decomposition phase, which can include/coincide with the feature engineering phase, is almost skipped entirely, ignoring a lot of possible data/model enhancements and steps towards a well conditioned problem. On the context of ASER, it is a choice to take this kind of approach due to the efficiency of CNNs on these kind of tasks as well as the good representation capacity associated to Log Mel Spectrograms. To understand why, its necessary to know what Log Mel Spectrograms represent when one refers it to audio speech data and why they fit with CNN architecture.

Audio signal's properties dictate that they vary through time. There are representations of such in time domain or frequency domain, however, our interpretation on sound arises from spectral or temporal aspects of the sound itself, which creates a need for time-frequency representations (Darji, 2017). For that matter, the use of spectrograms (French and Handy, 2007) comes up.

Spectrograms are time-frequency representations that provide ways of discerning important features of many different types of audio signals and can be very helpful when the signal doesn't change rapidly in time, although, on situations where it actually does, the context on how the frequencies change can also contribute to the understanding of the underlying information.

In order, to create such representations, one refers to the Short-Time Fourier Transform (STFT) (Allen, 1982). STFT is a sequence of Discrete Fourier Transforms (DFTs) (Bracewell and Bracewell, 1986) of a windowed signal, where each value corresponds to the energy in the dataframe.

The calculation of the STFT can be then described as follows through the STFT pair on Equation 1.

$$\begin{cases} X_{STFT}[m,n] = \sum_{k=0}^{L-1} x[k]w[k-m]e^{-j2\pi nk/L} \\ x[k] = \sum_m \sum_n X_{STFT}[m,n]w[k-m]e^{j2\pi nk/L} \end{cases} \tag{1}$$

where x[k] denotes a discrete signal and w[n] denotes a windowing function. The first term refers to the transformation of the signal to a spectrogram through STFT, whereas the second term refers to the inverse operation (Kehtarnavaz, 2008). The computational algorithm is built by the Fast Fourier Transform (FFT) (Brigham and Morrow, 1967), a fast, efficient way of computing the DFT on a computational environment. The window function is present to address the issues related to spectral leakage, originated by the assumption that the signal on each data block is periodic: when the FFT is calculated over non periodic signals, the frequency spectrum suffers from leakage, originating artifacts or noise in the representation. To fight this, the window function, composed by zeros on both ends and a special defining shape in the middle, is introduced and multiplied by every data frame in order to address the characteristic discontinuities on the edges of a signal (Ramirez, 1985). A common window function used in the calculation of STFT on the context of audio is the Hann Function (Harris, 1978), a bell shapped function, which will also be used on this dissertation, by default.

Following the calculations for the spectrogram, the Mel scale (Umesh et al., 1999) (Stevens et al., 1937) is applied. The Mel scale is a fundamental result of psychoacoustics that looks to establish a relationship between the real frequency and the perceived frequency. This representation looks to somewhat emulate how humans actually perceive sound, by affecting how frequency scales up to higher values. One of the most popular formulas for converting signals to the Mel scale is given by Equation 2.

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{2}$$

After this, the Mel spectrogram is put through a transformation that converts the power values to Decibel (dB) units, creating a more visible representation of the spectrogram, commonly known as Log Mel Spectrogram. Fig.4 shows a comparison between representations, audioform and Log Mel Spectrogram from each of the selected classes, with arbitrarily selected instances, on the IEMOCAP. On the left, the raw audioform is presented. The speech parts are relatively easy to distinguish as they can be identified by the sudden changes along the X-axis. On the right, the Log Mel Spectrogram presents itself as a rich image-like representation of sound. The speech parts are identified by a higher dB values on multiple frequencies along the X-axis. These two representations are very different but both viable as an input to a neural network system. A pure end-to-end system would rely on actual raw waveforms without any further processing. This won't be the case here though due to the size of a raw waveform, meaning, number of samples per data instance, as it could be problematic from the point of view of a generator in a GAN system to create instances of such dimensions.

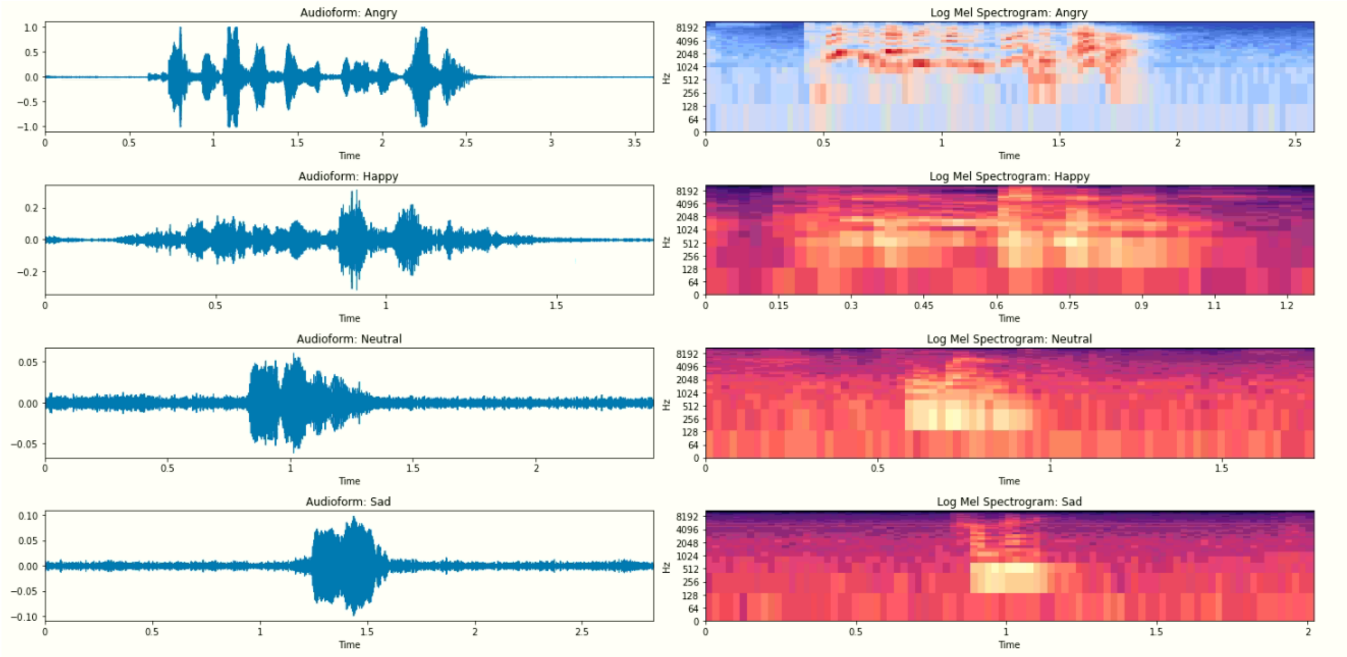

Figure 4: Distribution of the four selected labels over the dataset. .

3.3 Model Contextualization

Regarding the nature of the neural network system used in this work, GANs will be used to employ a solution for this task. GANs are part of a family of models denominated by Deep Generative Models (DGMs) (Ruthotto and Haber, 2021). DGMs can be understood as neural networks that reach a certain level of complexity and aim to approximate probability distributions over high dimensional parameters, on the expense of a lot of data samples. With the astonishing success of DL, these type of models became a prime topic of research, as their usage can serve the most diverse purposes, a lot of times only bounded by imagination (theoretically). The potential of such models, specifically of GANs, is what proved attractive to dive into the matter of the adaptation of such to SSL, which, until today, remains as a topic with very few resources for real world applications. This can be attributed to the high difficulty of training on these types of systems, both computational resource and complexity wise, to the lack of practical measures and testing on other than standard benchmark datasets like MNIST (Lecun et al., 1998) or CIFAR (Krizhevsky, 2009), and to the many implications that SSL has on the GAN system, regarding relationships between modules, that are still not completely understood or even catalogued at all.

3.3.1 GANs

Usually, the final goal of a GAN is to train a Generator network G where it learns a distribution p_g over data x by defining a prior on input noise variables $p_z(z)$, and linking it to data space by $G\left(z;\theta_g\right)$, where

G represents a neural network parameterized by θ_g. In order to provide gradients for G to learn on, a second network, Discriminator D, parameterized by θ_d, will generally output a single scalar value $D\left(x;\theta_d\right)$, corresponding to the probability over the origin of the data x from real data distribution p_{data} or from the G's data distribution p_g. D is trained over a standard Binary Cross Entropy (BCE), where 0 corresponds to a fake example and 1 correspond to a real one, G is trained to minimize the probability of D classifying given instances as fake, as shown in Equation 3.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{dat}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \qquad (3)$$

The GAN formulation showed empirical issues as the theoretical assumptions made on the convergence analysis did not hold (Manisha and Gujar, 2019). One of the most common issues was associated with mode collapse, where the generator learns a single mode of the data distribution. More concretely, the same or extremely similar images are produced every time, usually because the generator "figures out" that the discriminator is fooled by that instance. The images tend to look good but lack variation. Another common problem is that the instability of GANs leads to loss values that are not consistent with converging, which created a lot of issues regarding interpretation of results from task to task. There have been instances of losses indicating divergence, but still realistic images being produced, without explanation whatsoever (Manisha and Gujar, 2019).

Regarding the training of G, the initial formulation also showed itself prone to vanishing gradients issues, so in practice, it got replaced by maximizing the probability of D classifying given instances as real (Goodfellow et al., 2014), which allowed some alleviation of this issue.

From this original formulation, multiple GAN variants appeared, among them, a SSL inspired approach (Salimans et al., 2016). used in this dissertation, where a supervised objective is added to the Discriminator in order the map data to a label and still take advantage of the feature extraction process associated with the GAN system.

3.3.2 SGAN

The standard Classifier receives a data point x as input and outputs a vector of size corresponding to K classes to which a Softmax function is applied, allowing to generate class probabilities. The model is trained on a cross entropy loss. By adding samples created by a Generator and assigning it a new pseudo-class, which will represent all generated samples, the loss function of the model can be extended to include a unsupervised objective, incorporating the GAN system into the classifier, shown in Equations 4 and 5.

$$L_{\text{supervised}} = -\mathbb{E}_{x,y\sim p_{\text{data}}(x,y)} \log p_{\text{model}}(y \mid x, y < K+1) \tag{4}$$

$$L_{\text{unsupervised}} = -\left\{\mathbb{E}_{x\sim p_{\text{data}}(x)} \log\left[1 - p_{\text{model}}(y = K+1 \mid x)\right] + \mathbb{E}_{x\sim G} \log\left[p_{\text{model}}(y = K+1 \mid x)\right]\right\} \tag{5}$$

By introducing the GAN system into the the classifier, the Generator module also requires a loss function in order to learn an approximation of the real data distribution. Instead of the standard initial approach of maximizing the probability of the Discriminator classifying an instance as real, the Feature Matching loss (Salimans et al., 2016) is used, where the objective of the Generator is to match the statistics of the actual data, corresponding to some predetermined intermediate layer of the Discriminator, as in Equation 6.

$$L_{\text{feature matching}} = \left\|\mathbb{E}_{x\sim p_{\text{data}}} \mathbf{f}(x) - \mathbb{E}_{z\sim p_z(z)} \mathbf{f}(G(z))\right\|_2^2 \tag{6}$$

, where $f(x)$ represents the activations of the intermediate Discriminator layer. This loss function has shown itself effective on SGAN training as it also tackles the instability of training during the procedure.

3.4 Experiment Design

3.4.1 Data Preparation

As a start, the dataset is divided into train set and test set, containing 80% and 20% of the data respectively, for a 5-fold cross validation. After this, each of the instances themselves are divided in 2 second chunks, with 1 seconds overlap on training and 1.6 seconds overlap on test, where each chunk inherits the label associated to its original utterance. This methodology, used recently in Xu, Zhang, Cui and Zhang (2021), allows to tackle the problem of variable length on samples. Although, it is important to note that CNNs have the capacity of processing variable length inputs due to the sliding window implementation, as long as the correct form of pooling is used when it comes to get an output class. For the context of a GAN, a variable output length might be troublesome to the generator and affect the stability of training to some degree, which is why the choice of fixing the length of samples was taken. While during the training process, each chunk represents an isolated instance, on testing time, an aggregation of all chunks per utterance is made by averaging the prediction output by the network. It was shown that higher level of overlapping between chunks led to a more robust classification and better performance overall (Xu, Zhang and Zhang, 2021). A very interesting chunk wise approach was made in Lin and Busso (2021), where a more dynamic solution presents itself by having chunks that are variable in length from instance to instance, leading to different overlap levels over each utterance and creating an adaptive behaviour to the processing of speech.

Each of these chunks are then converted to a Log Mel Spectrogram, where 40 bands are used, a common parameter used in this kind of transformation for ML purposes (Parthasarathy and Busso, 2019), with an STFT window of 2048 and hop length of 512, originating a 1-channel representation of dimensions 40x59.

Note that his chunk-like approach extends the number of training samples. Depending on the utterances (randomly) selected, the dataset, from the point of view of the Neural Network, gets bigger, usually approaching the value of 13000 training samples. The same does not happen for the test set as the aggregation step on classification fixes its length.

3.4.2 Evaluation Metric

For the purpose of evaluating and comparing the results obtained, three metrics were saved, the crossentropy loss, the UAR (also known as WA) and the UA. Among these three, UAR should be the more common one, as it allows for a better understanding of the model's performance towards an unbalanced dataset. It's important to have corresponding metrics when it comes to comparing results, which can vary quite a bit throughout all papers on ASER research, as the dataset conditions and the final goal for the experiment might change from researcher to researcher. Sometimes this is not possible as the labels for the dataset change: researchers opt out of categorical emotional labels in favour of continuous measures of valence, arousal and dominance, as training models with these has shown advantages over categorical labels (Khorrami et al., 2016). The human readability might be somewhat affected as the interpretation of continuous values towards an associated emotion can be trickier.

3.4.3 Model Definition

As a GAN system is employed, the architecture and hyperparameters defined will have a huge impact on the outcome, as these type of models are sensitive to changes in structure like depth of the network, learning rate, batch size (Lucic et al., 2018). The SGAN algorithm is no exception. The search for optimal hyperparameters on a task can be extremely expensive, even somewhat not viable, depending on the data in hand, the computational resources, time resources, personnel resources, among others. As a result, it is often advantageous to employ certain conditions verified by other researchers to ensure that there is some logic to the baseline created. The neural network architecture will then follow closely the work of Lecouat et al. (2018). The model starts by receiving an input Log Mel Spectrogram with dimensions 40x59x1. Then, three 2D convolutional layers, with 32 filters each, with a LeakyRElu (Xu et al., 2015) activation function after each layer are applied to the input. This is then followed by a Spatial Dropout Tompson et al. (2015) layer with a ratio of 0,3. Next, another identical block of convolutional layers with LeakyREly activation follows, but this time with 64 filters each, and once again a Spatial Dropout layer with the same ratio. The last convolutional block is composed by a 3x3 layer, then two 1x1 layers, with 64 filters each and LeakyRElu after, followed by a Global Average Pooling into the Dense units that will output the logits for the classifier and for the discriminator. On the classifier end, the general approach is used, where a softmax is applied to get an output probability of each class. For the discriminator, the kernel trick mentioned in Salimans et al. (2016) is used allowing for all the layers and units to be shared between Classifier and Discriminator.The unsupervised model takes the logits prior to the application of the softmax and makes a normalized sum of the exponential ouputs. In practice, an aggregation function Lambda layer is used on the logits output by the Dense units defined by:

$$D(x) = \frac{Z(x)}{Z(x)+1}, \text{ where } Z(x) = \sum_{k=1}^{K} \exp\left[l_k(x)\right] \tag{7}$$

The model has a small number of parameters due to the size of the labelled dataset: on SSL, it is common to use a very small amount of labelled data, which can be rather counter-intuitive when it comes to DL, as huge amounts of data are usually necessary to reliably train a DL system (Alom et al., 2019). The usage of Spatial Dropout and the last block of the model, consisting of Convolutional layers with 1x1 kernel sizes followed by a Global Average Pooling (GAP) (Lin et al., 2014), are also introduced to help the susceptibility of the model to overfit to the small portion of labelled instances. Note that all the layers are applied with Weight Normalization (Salimans and Kingma, 2016), a reparameterization of the weights to improve the conditioning of the optimization problem, shown to have promising results on generative models like GANs. The model schematics, for the Discriminator and Classifier shared architecture, up until the final logits Dense layer are shown in Table 1.

To complete the GAN system, the Generator has to be defined as well. As standard procedure, a latent dimension of 100 sampled from gaussian distribution is used. Then, to firstly make the shape of an image, 3x4x256 Dense units are used in the start of the processing of the noise input. Then, four similar blocks are employed, where each block consists of a Strided Convolution layers of 128 filters, Batch Normalization (Ioffe and Szegedy, 2015) with 0.8 on momentum, and LeakyRElu activation function. Following, two blocks, each composed by a Convolutional layer with 128 filters, Batch Normalization and LeakyRElu activation function for a finer processing of latent features. Finally, a Convolutional layer with 1 filter and Weight Normalization

Model Architecture
40 x 59 x 1 image (Spectrogram) 32 3x3 conv2D Padding Stride=(1,1) weightnorm lReLU 32 3x3 conv2D Padding Stride=(1,1) weightnorm lReLU 32 3x3 conv2D Padding Stride=(2,2) weightnorm lReLU
SpatialDropout(0.3)
64 3x3 conv2D Padding Stride=(1,1) weightnorm lReLU 64 3x3 conv2D Padding Stride=(1,1) weightnorm lReLU 64 3x3 conv2D Padding Stride=(2,2) weightnorm lReLU
SpatialDropout(0.3)
64 3x3 conv2D Stride=(2,2) weightnorm lReLU 64 1x1 conv2D Stride=(1,1) Padding weightnorm lReLU 64 1x1 conv2D Stride=(1,1) Padding weightnorm lReLU
Global Average Pooling Dense weightnorm

Table 1: Discriminator/Classifier architecture for the GAN system..

applied. The kernel shapes are not mentioned as they are adapted towards achieving the final shape of 40x59x1. The model is shown in Table 2.

Model Architecture
100 x 1 image (Noise) 3x4x256 Dense batchnorm lReLU 128 4x4 conv2DTranspose Stride=(2,2) Padding batchtnorm lReLU 128 4x4 conv2DTranspose Stride=(2,2) Padding batchtnorm lReLU 128 4x4 conv2DTranspose Stride=(2,2) Padding batchtnorm lReLU 128 4x4 conv2DTranspose Stride=(2,2) Padding batchtnorm lReLU
128 4x4 conv2D Stride=(1,1) batchtnorm lReLU 128 4x4 conv2D Stride=(1,1) batchtnorm lReLU
1 3x3 conv2D Stride=(1,1) weightnorm

Table 2: Generator architecture for the GAN system..

To train each of the models, two different optimizer objects are used. Both are Adam optimizers with 0.0001 and beta=0.5, combined with stochastic weight averaging (Izmailov et al., 2019) with 0.999 on the average decay parameter. The batch size for the discriminator is 128, where samples that have the same nature are grouped up and fed into the model, i.e. supervised samples, unsupervised real samples and unsupervised fake samples are separated into their own batches, then fed to the Classifier and Discriminator in the correct manner. For the Generator, the batch size is 256. Over all instances of testing, the number of epochs is set to 300, where each epoch is defined by the size of the whole dataset (labelled + unlabelled) divided by batch size. All the modules are initialized with a random Normal distribution, of mean 0 and standard deviation 0,05.

3.4.4 Test Specification and Results

Each testing instance is defined by a 5-fold cross validation and the number of labelled instances, which will vary between the values of 300, 600, 1200, 2400 and fully labelled dataset. The parcial values represent roughly 2%, 4%, 9% and 18% of the labelled data on the whole dataset. The results are then averaged among the tests to present a fair performance value with same weight on every run of the procedure. A simple ablation test is also ran, where the generator and the discriminator are turned down, turning the task into a fully supervised matter in order to assess the value that the GAN system brings to the model.

The first round of experiments starts with the lowest number of labelled samples to establish a baseline to progress on, as this is expected to present the lowest performance out of all of the SSL portion of the tests. By representing only 2% of the dataset as labelled samples, this is arguably the most insightful and important test because it illustrates situations where the amount of processed data is extremely scarce. So, the first test over 300 supervised samples reaches a value of 53.059% UAR and 49.391% UA, with a cross entropy loss of 1.5880238. From this initial result, it is expected that every test following will have a considerably higher level of performance.

In the second round of tests, with 600 labelled samples, the model hits 55.597% UAR and 52.967% UA, with a cross entropy loss of 1.3667043. Going from around 2% of labelled data to around 4% has resulted in the improvement of the evaluation metrics, UAR and UA, by at least 2.5%. The main loop code is shown at Appendix B, for an example test over 300 labelled samples.

With 1200 labels, the model hits 58.329% UAR and 56.067% UA, with a cross entropy loss of 1.1982944. At this point, almost 10% of labelled data is used which sets a mark on SSL paradigm based, as it can bee seen a reference for testing such algorithms. As the number of labelled instances goes higher, it's expected that the results of it converges slower and slower to the results that would've been obtained if all the samples were labelled. So 10% would be a good average indicator for low sample training performance on DL models. As expected, the improvement is still considerably high.

With 2400 labels, the model hits 62.855% UAR with 61.089% UA with a cross entropy loss of 1.0183306. There is a steady improvement on every metric, where doubling the amount of labelled data is resulting in a more or less consistent growth of the model's effectiveness. Naturally, the test instance with higher number of labelled samples achieves the best results. The compilation of all test results described are shown in Figures 5 and 6.

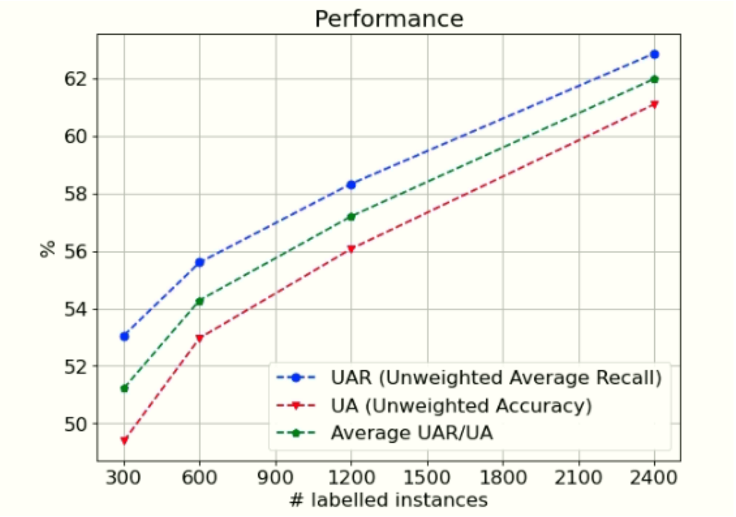

Figure 5: Performance metrics of UAR and UA along with an average.

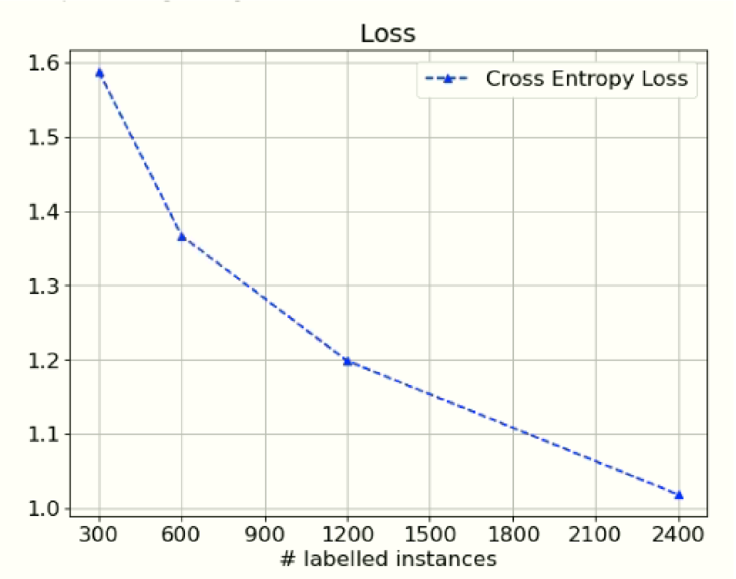

Figure 6: Cross entropy loss across tests.

By looking at these tests alone, the slope of performance shows some indices of slowing down, meaning that for a certain dataset, on the context of this procedure, there might be an optimal number of labelled instances that allows to reach some acceptable performance. The values obtained with this approach are competitive and even slightly surpass the state of the art shown by Zhao et al. (2020), which is, to the best of our knowledge, the best results obtained on an SGAN system, on the IEMOCAP dataset. This might be due to several factors: firstly, the usage of Log Mel Spectrograms combined with Convolutional layers might show a better capacity of feature extraction which combined with the efficiency on training CNNs makes this approach somewhat favorable; secondly, the aggregation function used, where the average of the predictions is taken as the final prediction for a collection of chunks of the same utterance, changes the classification scheme a bit, which can turn it into an invalid comparison, since both models are not compared on the same conditions; same can apply to training, where the chunk division inflates the dataset up to four or five times its original size; finally, the generator's impact might be greater, as no ablation tests are shown on the article, due to the naturalness of a convolutional generator to actually generate one channel representations, i.e. the generator might be able to pick features more easily when convolutional layers are employed; as an extra, the Feature Matching loss is used to trained the generator, a loss function that Salimans et al. (2016) shows to improve the performance on an SGAN system. Table 3 shows a comparison with the values of UAR obtained in Zhao et al. (2020), to understand how algorithm and procedures can change the output performance.

UAR (%)				
	300	600	1200	2400
Zhao et al. (2020)	52.3	55.4	57.8	59.3
This work	53.1	55.6	58.3	62.9

Table 3: Comparison of the results obtained on this work, compared to the state of the results for GAN on the IEMOCAP.

Along the experiments, as the number of labelled instances changes, the behaviour of the GAN system is more or less consistent, where they tend to converge to the same values of loss. This would be expected since the unsupervised task associated doesn't really change, data wise, as every single sample is utilized. Varying the number of unlabelled samples fed to the GAN system would be an interesting testing point to verify the impact of number of samples, which could potentially translate to changes in the supervised vertent as well. The plots of losses for 300 labelled instances and full dataset as labelled are shown on Figures 7 and 8, respectively.

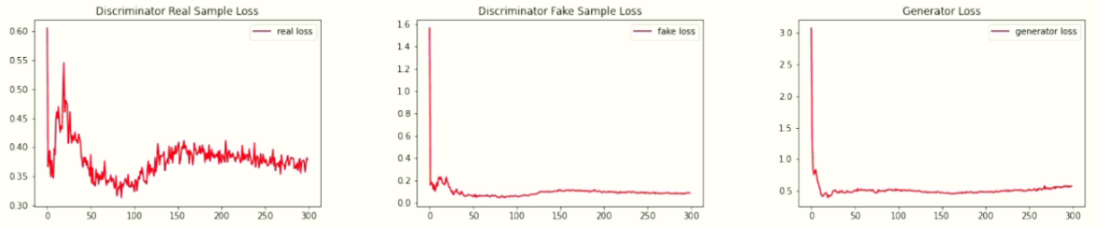

Figure 7: Cross entropy loss on Discriminator and Feature Matching loss on Generator, over 300 labelled instances.

The spike at the initial epochs represents the shock that the system suffers on the initialization of the procedure. The plots eventually get smoother as the training progresses.

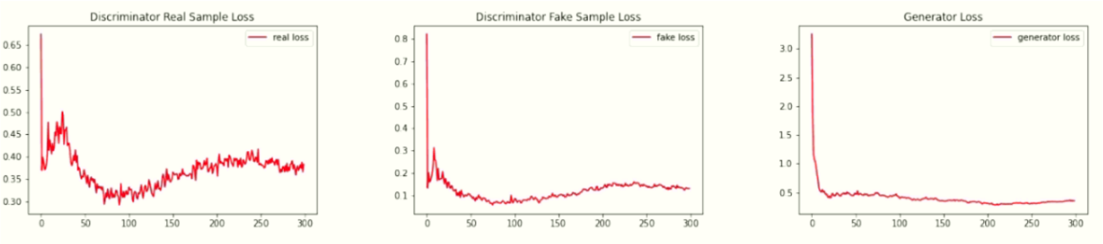

Figure 8: Cross entropy loss on Discriminator and Feature Matching loss on Generator, over full dataset as labelled instances.

The accuracy charts on the supervised component, the classifier, are a bit different. It would only be natural that, by changing the amount of labelled data fed into the Classifier, the behaviour of classification would change. For smaller amounts of data, the difference between training accuracy and validation accuracy would be bigger, because the model would be able to overfit to the small set, especially since 300 epochs is selected for every testing instance as a way of keep training uniform over all tests. The results are shown in Figure 9.

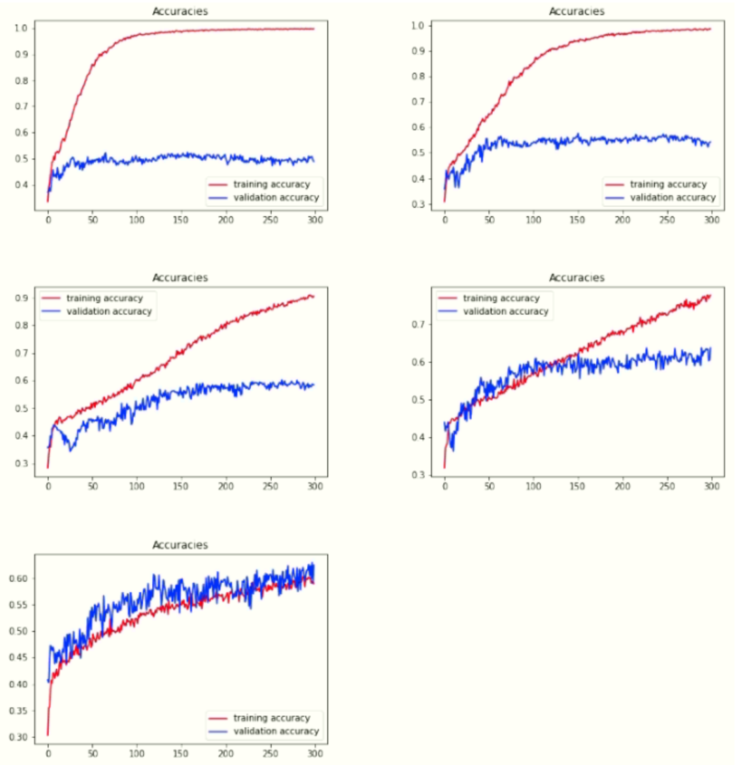

Figure 9: Evolution of accuracy charts, ranging through 300, 600 , 1200, 2400 and full dataset labelled instances, with 300 epochs, left to right, top to bottom, with UA metric.

As expected, a lower ammount a of labelled instances means that the model can reach close to 100% UA on the training set, as it is able to overfit. Inversely, the higher amount of labelled instances, the longer it takes for the model to reach its peak. In fact, when the model is trained with full dataset, we can verify that the accuracy values of training and validation set go hand to hand, meaning that further training would most likely produce a better model. An interesting observation is that, with 300 and 600 labelled instances, even though training reaches close to 100%, the validation metric doesnt really dip down drastically as it sometimes happens on supervised objectives. This could be due to the unsupervised task imposed by the GAN that doesn't allow the system to completely overfit to the training set. For insigh on the losses, they are displayed on Figure 10.

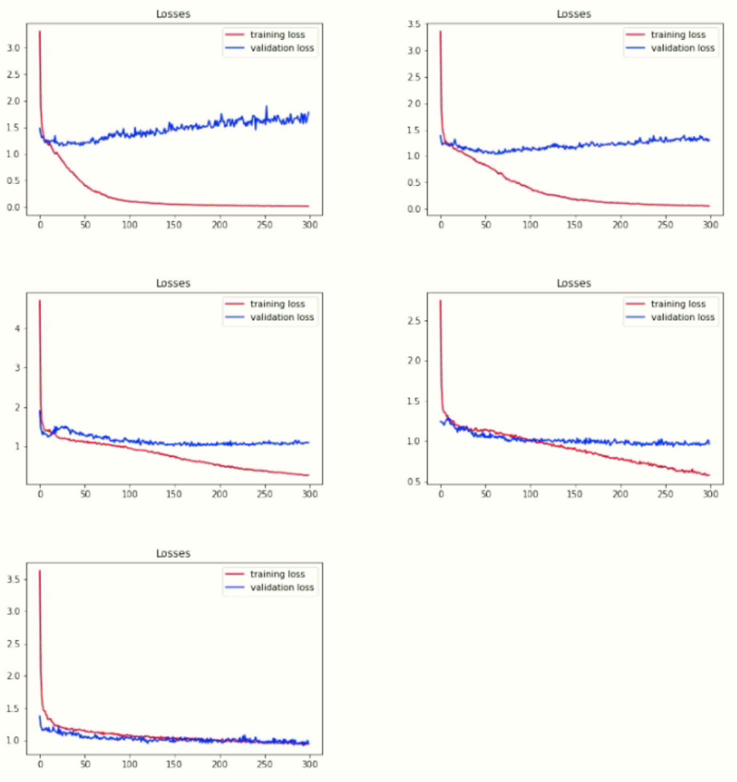

Figure 10: Evolution of cross entropy loss charts, ranging through 300, 600 , 1200, 2400 and full dataset labelled instances, with 300 epochs, left to right, top to bottom.

As mentioned before, to verify the performance changes that the GAN system brings to the Classifier, tests were made where the Generator and Discriminator task were disabled, for the same range of labelled instances of 300, 600, 1200, 2400 and full dataset length. For each of these instances, respectively, the results were the following: 48.296% UAR and 47.892% UA with a cross entropy loss of 2.0307658 for 300 labelled instances; 53.690% UAR and 50.810% UA with a cross entropy loss of 1.6905901 with 600 labelled instances; 54.632% UAR and 54.054% UA with a cross entropy loss of 1.3356034 for 1200 labelled instances; 58.284% UAR and 56.648% UA with a cross entropy loss of 1.2401884 for 2400 labelled instances; 64.003% UAR and 63.567% UA with a cross entropy loss of 0.9275583 for fully labelled dataset. Figures 11, 12 and 13 illustrate the performance difference between the SGAN and the same Classifier network without intervention of the GAN system, under the same conditions (learning rate, batch size, initialization).

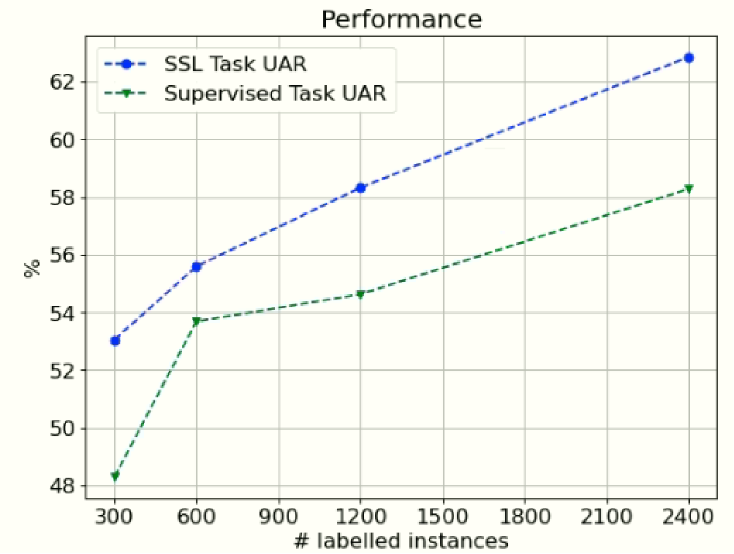

Figure 11: Evolution of UAR charts, ranging through 300, 600 , 1200, 2400, with a comparison between an SSL and an SL approach.

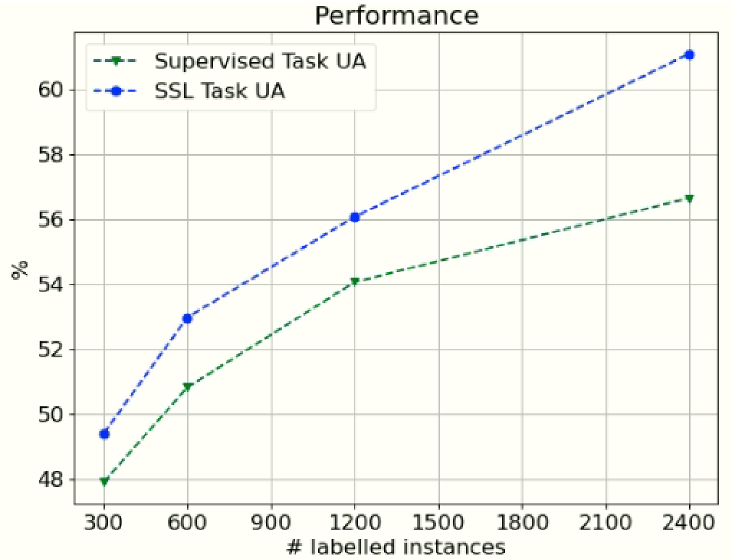

Figure 12: Evolution of UA charts, ranging through 300, 600 , 1200, 2400, with a comparison between an SSL and an SL approach.

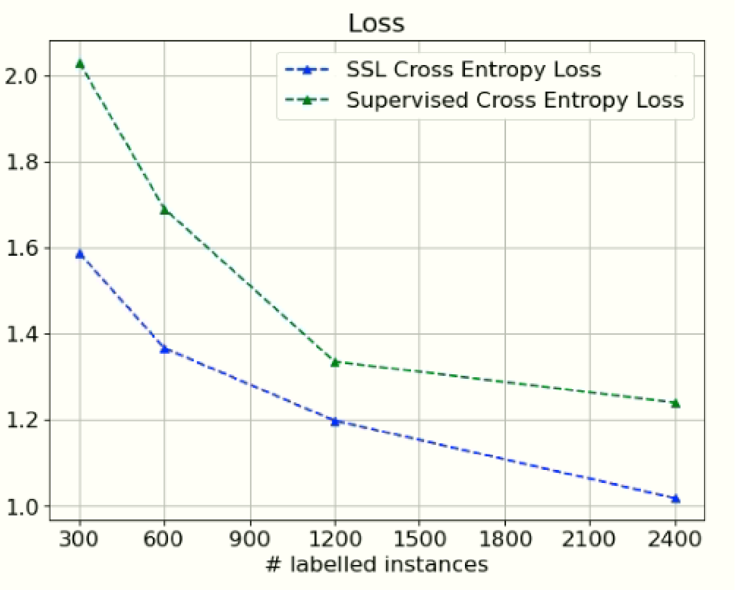

Figure 13: Evolution of the cross entropy loss charts, ranging through 300, 600 , 1200, 2400, with a comparison between an SSL and an SL approach.

As observed, there is a clear gain of performance from the SGAN when both frameworks are compared pair to pair. The performance gain might not be efficient enough though, as the optimization and tuning process of the SGAN takes a considerable amount of time. There are two practical flaws associated with this comparison though: firstly, the unlabelled data utilized is known to belong to the one of the four classes utilized on the system, which could influence the feature extraction process of the GAN towards these classes, and on a real world application, would not be a guarantee to have unlabelled data belonging to the same class as the labelled set. Secondly, the supervised test is not optimized towards the SL paradigm, as it takes on parameters and architectural techniques designed towards SSL.

It is, never the less, a start towards what could be the future, as semi supervised approaches start to grow over, performance wise.

Generally, when training an SGAN, the Generator stops being the focus on the task. This probably happens due to the lack of knowledge on the relationship between Generator and Classifier, even though scientists are looking for solid connections (Dai et al., 2017) (Liu and Xiang, 2020) (Lecouat et al., 2018). The problem is that most of these approaches do not show a empirical character or demonstrate a high level of complexity (execution wise), which makes them being overlooked when the focus in the whole task is not the algorithm itself. For this research, the generator was part of the focus until considerably sharp images were made, where no mode collapse would occur, by monitoring the losses of the GAN system. For a comparison, Figures 14 and 15 shows a direct look over fake (output of the generator) and real data that would be fed into the Discriminator.

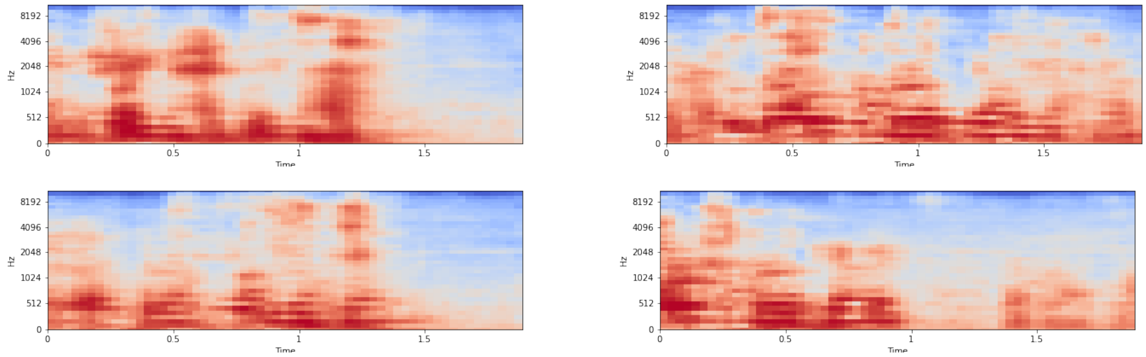

Figure 14: Images output out of the generator, ready to be fed to the Descriminator.

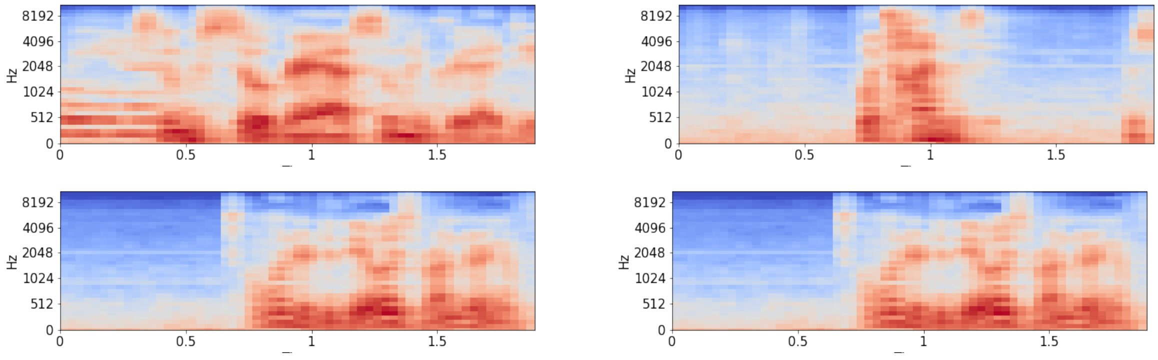

Figure 15: Real images post processing, ready to fed to the Discriminator.

4

DISCUSSION

In this chapter, the issues related to current frameworks are explored.

4.1 Datasets

Datasets made of audio speech have been gathered for years now with the purpose of designing systems capable of handling audio towards a certain task, in this particular case, we address ASER. As final goal, the usage in real world applications should always be selected, whether on a personal or commercial use. From a personal use point of view, customized products can be challenging. Even though there is a unique robustness to proper developed ML systems, a topic as volatile as emotions, considering the amount of different cultures there are and the amount of different ways there are of expressing an emotion, will never show itself as uniform. This means that, as a researcher developing a dataset, either one is able to include every instance and every type of emotion manifestation on the set, which is virtually impossible, or we look for aggregating representations for the utterances we have, which is where the design of discrete labels and continuous attributes attempt to emulate. It was already mentioned before in this work that discrete labels impose strict boundaries on emotions. It's the design of the system itself that orders that only one type of emotion is displayed overall. This is not correct as mental states overlap with each other and interact in a rather dynamic way (Frijda, 2004). This leads to an observation: it is rather counter intuitive to treat emotions as labels. The usage of dimensional attributes like arousal, dominance and valence can tackle this restricting factor that categorical labels impose. They present a fluid approach to what emotions are, where a 3-dimensional space can better characterize the mental state, even if this 3-dimensional space is composed by conventions. For the success on many scales of speech emotion recognition, it's important to represent an emotion as something flexible. Now the issue that arises with dimensional attributes that they themselves are conventions created, is that the measurement is not precise. As psychology concepts, there is not a scientific verification of these values. If one or more annotators are labelling instances with such attributes, suddenly, there is a big risk that these concepts do not practically represent what reality really is. This lack of ground truth, shared with categorical labels in a away, compromises the work done towards authenticity of emotion recognition. It is indeed a valuable start on standardizing the concept of emotion and translating it to paper, but we can not rely on this methodology to build high end high risk systems that could be on crucial positions to affect people's lives. The point is that current labelling schemes are limited to the paradigm they are built on.

To tackle such a difficult issue, we talk about two suggestions. The research on the basic emotional framework is evolving further and further (Keltner et al., 2019), more and more studies try to clarify the expression of mental states and coordinate into unified opinions. So it is not impossible that in a few months, maybe a few years, the framework of emotions will be clear enough to build rich sets of data that allow for the building of reliable systems for ASER. Note that Basic Emotion Theory (BET) does not apply only to speech data, as other modalities might be included later on. So as science evolves, we as Computer Scientists will have more and more tools and concepts available on the matter, that will not raise the credibility of information as an issue. Asking to wait can be seen as an invalid solution, which is completely reasonable. So the other solution could be digging into unsupervised representations between data that show different emotional content. UL can be generally used on unlabelled data with its limitations. But instead of using labels to describe certain aspects of an individual's manifestation, maybe we can use features in representations that can be associated to an emotion by utilizing the model as a tool to create simplified versions of the human mind that could be readable with some skill. Keep in mind that the idea is still to improve Machine-Human interactions, but just like topics like autonomous driving, this is only possible in an independent way only when the machine "understands" the base grounds for the task and is able to apply, which is not the case for ASER right now due to the built framework around it.

Researchers have been doing an amazing work to make ASER frameworks available. We came to a point where we need to be more rigorous with information, which will allow us to take machines to a whole other level. As we require more sophistication out of technology, it also needs a proper base where it can grow from.

4.2 SGAN framework

With unprocessed audio data existing in bulk, the promise of SSL is being taken up to. The leveraging of unlabelled data seems to show a future when a lot of efficiency and effectiveness could be extracted for the digital era we find ourselves on. On this work, with the SGAN framework, there was an attempt to incentive to such. Generative models do show a lot of potential on many tasks (Andrade et al., 2022), but what this potential translates to is that amazing accomplishments have been made with the algorithm, but there is still a lack of standardization together with huge traces of instability that doesn't allow research to flow as fast as it could. These kind of systems are remarkably hard to train, by requiring a relatively high amount of experience and touch as well as discipline and time/computational resources. Problems like this together with the issues on replicability discourage the production of SGANs to real world applications. On top of this, it is known that complex modules interact on a SGAN system, namely, generator, discriminator and classifier. There is still a huge lack of empirical understanding over the effect of generators on classifiers (Salimans et al., 2016), even though a lot of theoretical work was done to understand these relationships (Liu and Xiang, 2020). Two important thought processes branch out from this.

Firstly, the importance of unlabelled data should be a high priority research. To have the capacity of extracting value out of unlabelled data is to increase the efficiency of virtually every existing ML system deployed in real world applications. Not only a cost reduction is associated with this but also a performance gain, which could skyrocket technology's sophistication levels. It is a research on the go and on the rise, with SSL scientists leading the efforts.

The other branch is the selection of proper labelled data. The labelled data will always be the main link to the final labels. These are instances that form the core connections inside a mapping function forged by ML algorithms.

The SGAN presents many complex issues that a standard supervised task usually does not, which is why SSL is still a growing area, with a relatively low empirical use, compared to other paradigms.

5

CONCLUSIONS AND FUTURE WORK

5.1 Difficulties and Limitations

A few of the limitations involved was on the coding and design of the whole framework. Tensorflow Keras offers multiples APIs to deal with many different scenarios. The Sequential API allows for a rather straightforward creation of a Neural Network. The Functional API offers higher degree of control, allowing to create custom algorithms like GANs. The cost that comes with having higher degree of control is that a coder can be more prone to error, as more aspects of the algorithm are subject to change. Even though many official implementations of SGAN models exist in repositories like Github, a lot of them do not meet requirements to use newer software versions, which would make its adaptation way harder: it was necessary to go through the whole code, from the processing and loading of the data to the final stage of results of the model, to understand exactly the implications that a particular implementation would have on itself. The solution to avoid this was to design from scratch the code of SGAN in python, with Tensorflow Keras Functional API. As SGAN can still port some complexity, a lot of resources were sunk into the verification, correction and tuning of the algorithm, custom metrics, dataset process. To develop this kind of methodology, a higher level of discipline is required, as custom algorithms may fail due to the smallest mistake. The same goes for the processing of the data: the amount of transformations done to the data to make it ready to be fed is considerable, as it is filtered, divided and transformed multiple times. As the amount of data would grow, as research progresses to include/involve other datasets of bigger dimension, the rigor of processing becomes more and more necessary as the resources to allocate are higher and so is the time consumed.

Another issue rises with the inherent aspects of training an SGAN. The slow training speed, as multiple modules are backpropagated through independently, can be pretty heavy on time. Combining this with the fact that the algorithm itself has a lot of instability associated, i.e. exploding/vanishing gradients, numeric overflow/underflow, poor random initialization of the model, the training is interrupted unexpectedly many times. For this work, 300 epochs are selected but important SGAN works, like in Lecouat et al. (2018), run up to 1200 epochs on simple datasets, which can be rather taxing depending on resources available.

At last, the interpretation of the losses associated mostly with the GAN part, inside the SGAN system, are very loose. The wildness of this algorithm makes the losses and other metrics be rather variable and volatile from task to task, from dataset to dataset, from model to model. More than on any other algorithm, experience is very important in understanding where the training procedure of a GAN stands. Recognizing failure of the algorithm early is a very important step, as a problem will have many model setups that are not viable. As a counterpart, GANs are known for sometimes being able to bounce back from a bad start, which usually translate moving from mode collapse on the generator to rich representations, from the discriminator being able to identify real and fakes perfectly to a more balanced overview and loss functions, allowing for some proper gradients to run through the model, providing it that important feature extraction focus. This lack of theoretical and practical understanding of GANs affects all types of research done on it.

5.2 Future Work

As an improving point, the priority would come from the theoretical and practical understanding of GANs on SSL. The expression "leveraging unlabelled data" needs to be translated into something more concrete, so the development towards solid systems on this paradigm, on the context, can start in a more systematic way. With more experimentation, more empirical cues will be found, making the task of developing such a model more consistent.

Next, the methodology development needs more discipline. The gathering of information and research status was pretty broad and complete, but the translation of it into practice was rather lackluster. The multiple different phases on the development of ML systems need to be more evident, which by itself allows its maintenance in a more simplified, efficient and effective way.

5.3 Work Done

The field of SSL is very young. Only in the last few years, with the growth of technology, software and hardware wise, resources were allocated towards SSL experiments as the margin of experimentation grew larger and larger. Competitive results with fully supervised approaches are being obtained on the most diverse tasks (Andrade et al., 2022), which is motivating young researchers to dig deeper on the usecases of this framework. However, there is a rather complex nature, intrinsic to semi supervised approaches, that makes its introduction harder. A lot of work is needed to refine it to levels of reliability of SL and we are slowly moving towards it.

After this initial stabilizing, the aim is to deploy real world uses for such models. Unlabelled data is abundant in many many scenarios, which could turn SSL into a very attractive approach in areas like health, education.

This work is done to incentive the investigation over the effects of data on the SGAN algorithm and provide some insight on the frameworks of ASER, as well as setting up a path towards applications revolving audio with potential for high levels of parallelization and streamable capacity to produce efficient programs on the processing of speech data.

REFERENCES

(n.d.).

., M. and Kwon, S. (2020), `Clstm: Deep feature-based speech emotion recognition using the hierarchical convlstm network', 8, 19mathema.

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y. and Zheng, X. (2015), `TensorFlow: Large-scale machine learning on heterogeneous systems'. Software available from tensorflow.org.
URL: https://www.tensorflow.org/

Abbaschian, B. J., Sierra-Sosa, D. and Elmaghraby, A. (2021), `Deep learning techniques for speech emotion recognition, from databases to models', Sensors 21(4).
URL: https://www.mdpi.com/1424-8220/21/4/1249

Abdul Qayyum, A. B., Arefeen, A. and Shahnaz, C. (2019), Convolutional neural network (cnn) based speech-emotion recognition, in `2019 IEEE International Conference on Signal Processing, Information, Communication Systems (SPICSCON)', pp. 122–125.

Aeluri, P. and Vijayarajan, V. (2017), `Extraction of emotions from speech-a survey', International Journal of Applied Engineering Research 12, 5760–5767.

Allen, J. (1982), Applications of the short time fourier transform to speech processing and spectral analysis, in `ICASSP '82. IEEE International Conference on Acoustics, Speech, and Signal Processing', Vol. 7, pp. 1012–1015.

Alom, M. Z., Taha, T., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M., Hasan, M., Essen, B., Awwal, A. and Asari, V. (2019), `A state-of-the-art survey on deep learning theory and architectures', Electronics 8, 292.

Andrade, G., Rodrigues, M. and Novais, P. (2022), A survey on the semi supervised learning paradigm in the context of speech emotion recognition, in K. Arai, ed., `Intelligent Systems and Applications', Springer International Publishing, Cham, pp. 771–792.

Ayadi, M., Kamel, M. S. and Karray, F. (2011), `Survey on speech emotion recognition: Features, classification schemes, and databases', Pattern Recognition 44, 572–587.

Bank, D., Koenigstein, N. and Giryes, R. (2020), `Autoencoders'.

Ben-Hur, A. and Weston, J. (2010), `A user's guide to support vector machines', Methods in molecular biology (Clifton, N.J.) 609, 223–39.

Bengio, Y., Simard, P. and Frasconi, P. (1994), `Learning long-term dependencies with gradient descent is difficult', IEEE Transactions on Neural Networks 5(2), 157–166.

Bracewell, R. N. and Bracewell, R. N. (1986), The Fourier transform and its applications, Vol. 31999, McGraw-Hill New York.

Brigham, E. O. and Morrow, R. E. (1967), `The fast fourier transform', IEEE Spectrum 4(12), 63–70.

Bro, R. and Smilde, A. (2014), `Principal component analysis', Analytical methods 6, 2812.

Busso, C., Bulut, M., Lee, C.-C., Kazemzadeh, A., Mower Provost, E., Kim, S., Chang, J., Lee, S. and Narayanan, S. (2008), `Iemocap: Interactive emotional dyadic motion capture database', Language Resources and Evaluation 42, 335–359.

Carneiro, D., Novais, P., Augusto, J. C. and Payne, N. (2019), `New methods for stress assessment and monitoring at the workplace', IEEE Transactions on Affective Computing 10(2), 237–254.

Caruana, R. (1997), `Multitask learning', Machine Learning 28.

Cun, Y. L., Boser, B., Denker, J. S., Howard, R. E., Habbard, W., Jackel, L. D. and Henderson, D. (1990), Handwritten Digit Recognition with a Back-Propagation Network, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, p. 396–404.

Dai, Z., Yang, Z., Yang, F., Cohen, W. W. and Salakhutdinov, R. (2017), `Good semi-supervised learning that requires a bad gan'.

Darji, M. (2017), `Audio signal processing: A review of audio signal classification features', International Journal of Scientific Research in Computer Science, Engineering and Information Technology 2, 227–230.

Deng, J., Xu, X., Zhang, Z., Frühholz, S. and Schuller, B. (2017), `Semi-supervised autoencoders for speech emotion recognition', IEEE/ACM Transactions on Audio, Speech, and Language Processing PP, 1–1.

Douglas-Cowie, E., Campbell, N., Cowie, R. and Roach, P. (2003), `Emotional speech: Towards a new generation of databases', Speech Communication 40(1), 33 – 60.
URL: http://www.sciencedirect.com/science/article/pii/S0167639302000705

Drakopoulos, G., Pikramenos, G., Spyrou, E. and Perantonis, S. (2019), Emotion recognition from speech: A survey.

French, M. and Handy, R. (2007), `Spectrograms: Turning signals into pictures', Journal of Engineering Technology 24, 32–35.

Frijda, N. (2004), `Emotions and action', Journal of Organic Chemistry - J ORG CHEM .

Gabriel, M. (2018), `Machine learning - linear regression - lecture note 1'.

Glasmachers, T. (2017), `Limits of end-to-end learning', CoRR abs/1704.08305.
URL: http://arxiv.org/abs/1704.08305

Gonçalves, S., Rodrigues, M., Carneiro, D., Fdez-Riverola, F. and Novais, P. (2015), Boosting Learning: Non-intrusive Monitoring of Student's Efficiency, pp. 73–80.

Goodfellow, I., Bengio, Y. and Courville, A. (2016), Deep Learning, MIT Press. http://www.deeplearningbook.org.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014), `Generative adversarial networks'.

Goodfellow, I. J., Shlens, J. and Szegedy, C. (2015), `Explaining and harnessing adversarial examples'.

Grossi, E. and Buscema, M. (2008), `Introduction to artificial neural networks', European journal of gastroenterology hepatology 19, 1046–54.

Guzel Turhan, C. and Bilge, H. (2018), Recent trends in deep generative models: a review.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C. and Oliphant, T. E. (2020), `Array programming with NumPy', Nature 585(7825), 357–362.
URL: https://doi.org/10.1038/s41586-020-2649-2

Harris, F. (1978), `On the use of windows for harmonic analysis with the discrete fourier transform', Proceedings of the IEEE 66(1), 51–83.

He, K., Zhang, X., Ren, S. and Sun, J. (2015), Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in `2015 IEEE International Conference on Computer Vision (ICCV)', pp. 1026–1034.

Heckerman, D. (2008), A Tutorial on Learning With Bayesian Networks, Vol. 156, pp. 33–82.

Heidrich-Meisner, V., Lauer, M., Igel, C. and Riedmiller, M. (2007), Reinforcement learning in a nutshell, pp. 277–288.

Hinton, G. E., Osindero, S. and Teh, Y.-W. (2006), `A fast learning algorithm for deep belief nets', Neural Comput. 18(7), 1527–1554.
URL: https://doi.org/10.1162/neco.2006.18.7.1527

Hinton, G. and Salakhutdinov, R. (2006), `Reducing the dimensionality of data with neural networks', Science (New York, N.Y.) 313, 504–7.

Hochreiter, S. and Schmidhuber, J. (1997), `Long short-term memory', Neural computation 9, 1735–80.

Hopfield, J. J. (1988), Neural Networks and Physical Systems with Emergent Collective Computational Abilities, MIT Press, Cambridge, MA, USA, p. 457–464.

Ioffe, S. and Szegedy, C. (2015), `Batch normalization: Accelerating deep network training by reducing internal covariate shift'.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D. and Wilson, A. G. (2019), `Averaging weights leads to wider optima and better generalization'.

Jalal, A., Milner, R. and Hain, T. (2020), Empirical interpretation of speech emotion perception with attention based model for speech emotion recognition.

Jalal, M. A., Moore, R. K. and Hain, T. (2019), Spatio-temporal context modelling for speech emotion classification, in `2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)', pp. 853--859.

Jordan, M. I. (1989), Serial order: A parallel, distributed processing approach, in J. L. Elman and D. E. Rumelhart, eds, `Advances in Connectionist Theory: Speech', Erlbaum, Hillsdale, NJ.

Kehtarnavaz, N. (2008), Chapter 7 - frequency domain processing, in N. Kehtarnavaz, ed., `Digital Signal Processing System Design (Second Edition)', second edition edn, Academic Press, Burlington, pp. 175–196.
URL: https://www.sciencedirect.com/science/article/pii/B9780123744906000076

Keltner, D., Sauter, D., Tracy, J. and Cowen, A. (2019), `Emotional expression: Advances in basic emotion theory', Journal of Nonverbal Behavior 43.

Khalil, R. A., Jones, E., Babar, M. I., Jan, T., Zafar, M. H. and Alhussain, T. (2019), `Speech emotion recognition using deep learning techniques: A review', IEEE Access 7, 117327–117345.

Khorrami, P., Le Paine, T., Brady, K., Dagli, C. and Huang, T. S. (2016), How deep neural networks can improve emotion recognition on video data, in `2016 IEEE International Conference on Image Processing (ICIP)', pp. 619–623.

Kotsiantis, S., Zaharakis, I. and Pintelas, P. (2006), `Machine learning: A review of classification and combining techniques', Artificial Intelligence Review 26, 159–190.

Krizhevsky, A. (2009), `Learning multiple layers of features from tiny images', pp. 32–33.
URL: https://www.cs.toronto.edu/ kriz/learning-features-2009-TR.pdf

Kurpukdee, N., Koriyama, T., Kobayashi, T., Kasuriya, S., Wutiwiwatchai, C. and Lamsrichan, P. (2017), Speech emotion recognition using convolutional long short-term memory neural network and support vector machines, in `2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)', pp. 1744–1749.

Landowska, A. (2019), `Uncertainty in emotion recognition', Journal of Information, Communication and Ethics in Society 17(3), 273–291.

Larose, D. T. and Larose, C. D. (2015), Data Mining and Predictive Analytics, 2nd edn, Wiley Publishing.

Latif, S., Rana, R., Khalifa, S., Jurdak, R., Qadir, J. and Schuller, B. W. (2021), `Deep representation learning in speech processing: Challenges, recent advances, and future trends'.

Lecouat, B., Foo, C.-S., Zenati, H. and Chandrasekhar, V. (2018), `Manifold regularization with gans for semi-supervised learning'.

LeCun, Y., Bengio, Y. and Hinton, G. (2015), `Deep learning', Nature 521, 436–44.

Lecun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998), `Gradient-based learning applied to document recognition', Proceedings of the IEEE 86(11), 2278–2324.

Lee, J., Yang, J. and Wang, Z. (2020), `What does cnn shift invariance look like? a visualization study'.

Li, Y., Kaiser, L., Bengio, S. and Si, S. (2020), `Area attention'.

Lin, M., Chen, Q. and Yan, S. (2014), `Network in network'.

Lin, W.-C. and Busso, C. (2021), `Chunk-level speech emotion recognition: A general framework of sequence-to-one dynamic temporal modeling', IEEE Transactions on Affective Computing pp. 1–1.

Lipton, Z. (2015), `A critical review of recurrent neural networks for sequence learning'.

Liu, Q. and Wu, Y. (2012), `Supervised learning'.

Liu, X. and Xiang, X. (2020), `How does gan-based semi-supervised learning work?'.

Lotfian, R. and Busso, C. (2019), `Building naturalistic emotionally balanced speech corpus by retrieving emotional speech from existing podcast recordings', IEEE Transactions on Affective Computing 10(4), 471–483.

Lucic, M., Kurach, K., Michalski, M., Gelly, S. and Bousquet, O. (2018), `Are gans created equal? a large-scale study'.

Maalouf, M. (2011), `Logistic regression in data analysis: An overview', International Journal of Data Analysis Techniques and Strategies 3, 281–299.

Maheshwari, A. (2020), Autoencoders.

Manisha, P. and Gujar, S. (2019), `Generative adversarial networks (gans): What it can generate and what it cannot?'.

Martins, R., Gomes, M., Almeida, J., Novais, P. and Henriques, P. (2018), Hate speech classification in social media using emotional analysis, pp. 61–66.

Mccarthy, J., Minsky, M., Rochester, N. and Shannon, C. (2006), `A proposal for the dartmouth summer research project on artificial intelligence', AI Magazine 27.

McCulloch, W. S. and Pitts, W. (1988), A Logical Calculus of the Ideas Immanent in Nervous Activity, MIT Press, Cambridge, MA, USA, p. 15–27.

McFee, B., Metsai, A., McVicar, M., Balke, S., Thomé, C., Raffel, C., Zalkow, F., Malek, A., Dana, Lee, K., Nieto, O., Ellis, D., Mason, J., Battenberg, E., Seyfarth, S., Yamamoto, R., viktorandreevichmorozov, Choi, K., Moore, J., Bittner, R., Hidaka, S., Wei, Z., nullmightybofo, Hereñú, D., Stöter, F.-R., Friesch, P., Weiss, A., Vollrath, M., Kim, T. and Thassilo (2021), `librosa/librosa: 0.8.1rc2'.
URL: https://doi.org/10.5281/zenodo.4792298

Meng, H., Yan, T., Yuan, F. and Wei, H. (2019), `Speech emotion recognition from 3d log-mel spectrograms with deep learning network', IEEE Access PP, 1–1.

Milner, R., Jalal, M. A., Ng, R. W. M. and Hain, T. (2019), A cross-corpus study on speech emotion recognition, in `2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)', pp. 304–311.

Miyato, T., ichi Maeda, S., Koyama, M. and Ishii, S. (2018), `Virtual adversarial training: A regularization method for supervised and semi-supervised learning'.

Mousavi, S., Schukat, M. and Howley, E. (2018), Deep reinforcement learning: An overview, pp. 426–440.

Muckenhirn, H., Abrol, V., Magimai.-Doss, M. and Marcel, S. (2019), Understanding and visualizing raw waveform-based cnns, in `INTERSPEECH'.

Munjal, P., Paul, A. and Krishnan, N. C. (2019), `Implicit discriminator in variational autoencoder'.

Méndez-Villas, A. (2005), Badajoz : Formatex 3, 929–934.

Nabipour, M., Nayyeri, P., Jabani, H., Mosavi, A., Salwana, E. and S., S. (2020), `Deep learning for stock market prediction', Entropy 22(8), 840.
URL: http://dx.doi.org/10.3390/e22080840

Nwankpa, C., Ijomah, W., Gachagan, A. and Marshall, S. (2020), `Activation functions: Comparison of trends in practice and research for deep learning'.

Odena, A. (2016), `Semi-supervised learning with generative adversarial networks'.

Olgac, A. and Karlik, B. (2011), `Performance analysis of various activation functions in generalized mlp architectures of neural networks', International Journal of Artificial Intelligence And Expert Systems 1, 111--122.

Ouali, Y., Hudelot, C. and Tami, M. (2020), `An overview of deep semi-supervised learning'.

Pan, W., Li, Z. and Zhang, Y. (2018), `The new hardware development trend and the challenges in data management and analysis', Data Sci. Eng 3, 263–276.

pandas development team, T. (2020), `pandas-dev/pandas: Pandas'.
URL: https://doi.org/10.5281/zenodo.3509134

Parthasarathy, S. and Busso, C. (2019), `Semi-supervised speech emotion recognition with ladder networks'.

Pathak, S. and Kolhe, V. L. (2016), `Emotion recognition from speech signals using deep learning methods', Imperial journal of interdisciplinary research 2.

Plutchik (2000), `Emotions in the practice of psychotherapy: clinical implications of affect theories'.

Rabiner, L. and Juang, B. (1986), `An introduction to hidden markov models', IEEE ASSP Magazine 3(1), 4--16.

Radford, A., Metz, L. and Chintala, S. (2016), `Unsupervised representation learning with deep convolutional generative adversarial networks'.

Ramirez, R. W. (1985), The FFT Fundamentals and Concepts, Prentice-Hall, Inc., USA.

Reynolds, D. (2008), `Gaussian mixture models', Encyclopedia of Biometrics .

Rodrigues, M., Durães, D., Santos, R. and Analide, C. (2021), Emotion detection throughout the speech, in K. Arai, S. Kapoor and R. Bhatia, eds, `Intelligent Systems and Applications', Springer International Publishing, Cham, pp. 304–314.

Rodrigues, M., Fdez-Riverola, F. and Novais, P. (2012), An approach to assessing stress in e-learning students.

Rodrigues, M., Monteiro, V., Fernandes, B., Silva, F., Analide, C. and Santos, R. (2020), `A gamification framework for getting residents closer to public institutions', Journal of Ambient Intelligence and Humanized Computing 11.

Rosenblatt, F. (1958), `The perceptron: a probabilistic model for information storage and organization in the brain.', Psychological review 65 6, 386–408.

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1988), Learning Representations by Back-Propagating Errors, MIT Press, Cambridge, MA, USA, p. 696–699.

Rumelhart, D. E. and McClelland, J. L. (1987), Learning Internal Representations by Error Propagation, pp. 318–362.

Ruthotto, L. and Haber, E. (2021), `An introduction to deep generative modeling'.

Salehinejad, H., Sankar, S., Barfett, J., Colak, E. and Valaee, S. (2017), `Recent advances in recurrent neural networks'.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A. and Chen, X. (2016), `Improved techniques for training gans'.

Salimans, T. and Kingma, D. P. (2016), `Weight normalization: A simple reparameterization to accelerate training of deep neural networks'.

Saxena, D. and Cao, J. (2020), `Generative adversarial networks (gans): Challenges, solutions, and future directions'.

Sejnowski, T. J. (2018), The Deep Learning Revolution, The MIT Press.

Sezer, O. B., Gudelek, M. U. and Özbayoglu, A. M. (2019), `Financial time series forecasting with deep learning : A systematic literature review: 2005-2019', CoRR $\mathrm{abs}/1911.13288$.
URL: http://arxiv.org/abs/1911.13288

Shaheen, F., Verma, B. and Asafuddoula, M. (2016), Impact of automatic feature extraction in deep learning architecture, pp. 1–8.

Shamsaldin, A., Fattah, P., Rashid, T. and Al-Salihi, N. (2019), `The study of the convolutional neural networks applications', UKH Journal of Science and Engineering 3, 31–40.

Shiota, . and null, M. (2016), `The sage encyclopedia of theory in psychology'.

Singh, A., Nowak, R. and Zhu, X. (2008), Unlabeled data: Now it helps, now it doesn't, pp. 1513–1520.

Somefun, O., Akingbade, K. and Dahunsi, F. (2020), `The nlogistic-sigmoid function'.

Stevens, S., Volkmann, J. and Newman, E. (1937), A scale for the measurement of the psychological magnitude pitch.
URL: https://books.google.pt/books?id=9SCWoAEACAAJ

Teixeira, A., Rodrigues, M., Carneiro, D. and Novais, P. (2020), HORUS: An Emotion Recognition Tool, pp. 126–140.

Tilve, A., Nayak, S., Vernekar, S., Turi, D., Shetgaonkar, P. R. and Aswale, S. (2020), Pneumonia detection using deep learning approaches, in `2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)', pp. 1–8.

Tompson, J., Goroshin, R., Jain, A., LeCun, Y. and Bregler, C. (2015), `Efficient object localization using convolutional networks'.

Turing, A. M. (1950), `Computing machinery and intelligence'. One of the most influential papers in the history of the cognitive sciences: http://cogsci.umn.edu/millennium/final.html.
URL: http://cogprints.org/499/

Umesh, S., Cohen, L. and Nelson, D. (1999), Fitting the mel scale, Vol. 1, pp. 217 – 220 vol.1.

Wang, Q., Ma, Y., Zhao, K. and Tian, Y. (2020), `A comprehensive survey of loss functions in machine learning', Annals of Data Science .

Wani, T. M., Gunawan, T. S., Qadri, S. A. A., Kartiwi, M. and Ambikairajah, E. (2021), `A comprehensive review of speech emotion recognition systems', IEEE Access 9, 47795–47814.

Weik, M. H. (2001), Nyquist theorem, Springer US, Boston, MA, pp. 1127–1127.
URL: https://doi.org/10.1007/1-4020-0613-6$_1$2654

Wes McKinney (2010), Data Structures for Statistical Computing in Python, in Stéfan van der Walt and Jarrod Millman, eds, `Proceedings of the 9th Python in Science Conference', pp. 56 – 61.

Woolf, B. (2008), Building Intelligent Interactive Tutors, Student-Centered Strategies for Revolutionizing E-Learning.

Xu, B., Wang, N., Chen, T. and Li, M. (2015), `Empirical evaluation of rectified activations in convolutional network'.

Xu, M., Zhang, F., Cui, X. and Zhang, W. (2021), `Speech emotion recognition with multiscale area attention and data augmentation'.

Xu, M., Zhang, F. and Zhang, W. (2021), `Head fusion: Improving the accuracy and robustness of speech emotion recognition on the iemocap and ravdess dataset', IEEE Access 9, 74539–74549.

Yang, X., Song, Z., King, I. and Xu, Z. (2021), `A survey on deep semi-supervised learning'.

Zhao, H., Yufeng, X. and Zhang, Z. (2020), `Robust semisupervised generative adversarial networks for speech emotion recognition via distribution smoothness', IEEE Access PP, 1–1.

Çoban, E. (2016), `Neural networks and their applications'.

A

PUBLICATIONS

Andrade G., Rodrigues M., Novais P. (2022) A Survey on the Semi Supervised Learning Paradigm in the Context of Speech Emotion Recognition. In: Arai K. (eds) Intelligent Systems and Applications. IntelliSys 2021. Lecture Notes in Networks and Systems, vol 295. Springer, Cham. https://doi.org/10.1007/978-3-030-82196-8_57

B

CODE

```
def train(g_model, d_model, c_model, gan_model, numpy_iemotrain, numpy_iemotest,
    latent_dim, iemotest, n_epochs=300, n_batch=256):
    # select supervised dataset
    global train_history_closs
    global train_history_dloss_r
    global train_history_dloss_f
    global train_history_gloss
    global val_history_acc
    global val_history_loss
    supervised_data = select_supervised_samples(numpy_iemotrain,n_samples=300)

    bat_per_epo = int(numpy_iemotrain.shape[0] / n_batch)

    n_steps = bat_per_epo * n_epochs

    half_batch = int(n_batch / 2)
    print('n_epochs=%d, n_batch=%d, 1/2=%d, b/e=%d, steps=%d' % (n_epochs, n_batch,
        half_batch, bat_per_epo, n_steps))

    train_history_acc_avg = []
    train_history_closs_avg = []
    train_history_dloss_r_avg = []
    train_history_dloss_f_avg = []
    train_history_gloss_avg = []

    for i in range(n_steps):
        # update classifier
        Xsup_real, ysup_real = generate_real_samples(supervised_data, half_batch,labelled=
            True)
        c_loss, c_acc = c_model.train_on_batch(Xsup_real, ysup_real)
        train_history_acc_avg.append(c_acc)
        train_history_closs_avg.append(c_loss)
        # update unsupervised discriminator
        X_real,y_real = generate_real_samples(numpy_iemotrain, half_batch,labelled=False,
            smooth=True)
        d_loss1 = d_model.train_on_batch(X_real, y_real)
        train_history_dloss_r_avg.append(d_loss1)

        X_fake, y_fake = generate_fake_samples(g_model, latent_dim, half_batch)
        d_loss2 = d_model.train_on_batch(X_fake, y_fake)
        train_history_dloss_f_avg.append(d_loss2)



        # update generator
        X_real,y_real = generate_real_samples(numpy_iemotrain, n_batch,labelled=False)
        X_gan, y_gan = generate_latent_points(latent_dim, n_batch), gan_activation_model.
            predict(X_real)

        g_loss = gan_model.train_on_batch(X_gan, y_gan)
        train_history_gloss_avg.append(g_loss)

        print('>%d, c[%.3f,%.0f], d[%.3f,%.3f], g[%.3f]' % (i+1, c_loss, c_acc*100,
            d_loss1, d_loss2, g_loss))

        # evaluate the model performance every epoch
        if (i+1) % (bat_per_epo * 1) == 0:
            clear_output()
            print(supervised_data.shape)
            train_history_acc.append(np.mean(train_history_acc_avg))
            train_history_acc_avg = []

            train_history_closs.append(np.mean(train_history_closs_avg))
            train_history_closs_avg = []

            train_history_dloss_r.append(np.mean(train_history_dloss_r_avg))
            train_history_dloss_r_avg = []

            train_history_dloss_f.append(np.mean(train_history_dloss_f_avg))
            train_history_dloss_f_avg = []

            train_history_gloss.append(np.mean(train_history_gloss_avg))
            train_history_gloss_avg = []
            summarize_performance(n_steps,i, bat_per_epo, g_model, c_model, d_model,
                latent_dim, supervised_data, numpy_iemotest,iemotest)
```

```
#size of the latent space
latent_dim = 100

# create the discriminator models
d_model, c_model, gan_activation_model = define_discriminator()

# create the generator
g_model = define_generator(latent_dim)

# create the gan
gan_model = define_gan(g_model, d_model,gan_activation_model)

# load image data
numpy_iemotrain = np.load("numpy_iemotrain_object.npy",allow_pickle=True)
numpy_iemotest = np.load("numpy_iemotest_object.npy",allow_pickle=True)

# train model
train(g_model, d_model, c_model, gan_model, numpy_iemotrain, numpy_iemotest, latent_dim,
    iemotest)
```

.