Pedro Marques Alegre de Sá

MERGE Audio: Audio Classification with Deep Learning

MSc Report

Pedro Marques Alegre de Sá

UNIVERSIDADE D COIMBRA

1 2 9 0

UNIVERSIDADE D COIMBRA

1 2 9 0

MERGE Audio: Music Emotion Recognition next Generation - Audio Classification with Deep Learning

MSc Thesis

Dissertation in the context of the Master in Informatics Engineering, Specialization in Intelligent System advised by Professor Rui Pedro Paiva and Professor Renato Panda and presented to the Faculty of Sciences and Technology / Department of Informatics Engineering.

October 2021

Faculty of Sciences and Technology

Department of Informatics Engineering

MERGE Audio: Music Emotion Recognition next Generation – Audio Classification with Deep Learning

MSc Thesis

Pedro Marques Alegre de Sá

Dissertation in the context of the Master in Informatics Engineering, Specialization in Intelligent Systems advised by Prof. Rui Pedro Paiva and Prof. Renato Panda and presented to the Faculty of Sciences and Technology / Department of Informatics Engineering.

October 2021

UNIVERSIDADE D
COIMBRA

This page is intentionally left blank.

Acknowledgments

As I come to conclude this journey, I reflect on the past year of extensive reading, countless hours researching and hitting the *run* button an endless amount of times, just to see myself having to hit it again due to an error I could swear I had addressed. I take a step back to try and see all the work put into this project and all I can remember is the immense amount of support throughout the past years.

I would like to show my gratitude to my advisors, Prof. Rui Pedro Paiva and Prof. Renato Panda, for always being available, for the constant care and incentive, for persistently giving me the right guidance, the most sincere critics and suggestions and for truly making this process as enjoyable as it could possibly be.

I wholeheartedly say thank you to my family, my mother Conceição, my father João and my brothers João and Miguel, for keeping me focused, motivated and for their patience, love and willingness to be there, by my side in every step of the way. I can not stress enough the ungodly amount of support and how essential it was for me to start, to keep on going and to finish this project.

Immense gratitude to Salomé, who day after day kept me grounded, who always supported me and made sure I never lost sight of the goal. I candidly thank you for the endless rants you had to sit through, for constantly being there for me, for the love and unequivocal support.

I am forever indebted to my friends, for all the long conversations, for all their aid and encouragement, for their advice and for the memories I will forever cherish.

To the University of Coimbra and primarily its professors, a special thank you for the utmost guidance and help along the way. I would also like to express my appreciation to CISUC, for the funding for this investigation project as a part of the Cognitive and Media Systems group, through the Foundation for Science and Technology.

I verily believe that the amount of support and thoughtfulness I received along this journey was the backbone to its success and with the last written sentence, of the last of countless iterations of this document, I want to express:

My deepest and most heartfelt thanks to all the people who directly and indirectly impacted this journey, as it made me a wiser student, a better professional and a greater person.

This page is intentionally left blank.

Abstract

The growing Music Emotion Recognition research field is evolving accompanied by an already massive and expanding library of digital music, which raises the need for it to be segmented and organized. Traditional Machine Learning approaches to identify perceived emotion in music are based on carefully crafted features that have dominated this field and brought state of the art results.

Our goal was to approach this field with Deep Learning (DL), as it can skip this expensive feature design by automatically extracting features. We propose a Deep Learning approach to the existing static 4QAED dataset, which achieved a state-of-the-art F1-Score of 88.45%. This model consisted in a hybrid approach with a Dense Neural Network (DNN) and a Convolutional Neural Network (CNN) for the features and melspectrograms (converted from audio samples), respectively.

Additionally, different methods of data augmentation were experimented with for the static MER problem, using a Generative Adversarial Network (GAN) and classical audio augmentation, which improved the overall performance of the model. Other pre-trained models were also tested (i.e. VGG19 and a CNN trained for music genre recognition). The Music Emotion Variation Detection field was explored as well, with (Bidirectional) Long Short Term Memory layers in combination with pre-trained CNN models, as we consider that the perceived emotion can change throughout the song.

This research gave us a good insight into several distinct deep learning approaches resulting in a new state-of-the-art result with the 4QAED dataset, in addition to getting to know the limitations of both datasets.

Keywords

deep learning, audio augmentation, music emotion recognition, music emotion variation detection

This page is intentionally left blank.

Resumo

A investigação do Reconhecimento da Emoção na Música (MER) está evoluir, acompanhado por uma biblioteca de música digital já maciça e em contínua expansão, o que levanta a necessidade de ser segmentada e organizada. As abordagens tradicionais de Aprendizagem Computacional para identificar a emoção percebida na música baseiam-se em *features* cuidadosamente trabalhadas que dominam este campo e são acompanhadas de resultados de estado da arte. O nosso objectivo foi abordar este campo com Aprendizagem Profunda, uma vez que pode saltar o dispendioso processo de criação de *features*, extraindo automaticamente as *features*. Propomos uma abordagem de Aprendizagem Profunda com a base de dados estática - 4QAED - já existente, que alcançou um F1-Score de 88,45%, superior ao estado da arte. Este modelo consistiu numa abordagem híbrida, com uma Dense Neural Network (DNN) e uma Convolutional Neural Network (CNN), para as features e mel-spectrogramas (convertidos a partir de amostras de áudio), respectivamente. Além disso, foram experimentados diferentes métodos de aumento de dados para o problema de MER estático, utilizando uma Generative Adversarial Network (GAN) e estratégias clássicas de modificação de áudio, o que melhorou o desempenho global do modelo. Outros modelos pré-treinados foram também testados (ou seja, VGG19 e uma CNN treinada para o reconhecimento do género musical). O campo de Detecção da Variação da Emoção Musical (MEVD) também foi explorado, com camadas de (Bidireccional) Long Short Term Memory em combinação com modelos CNN pré-treinados, considerando que a emoção percebida pode mudar ao longo de uma música. Esta investigação deu-nos uma visão aprofundada de várias abordagens distintas de Aprendizagem Profunda, contribuindo com num novo resultado de ponta com o conjunto de dados 4QAED, para além de conhecer as limitações de ambos os conjuntos de dados.

Palavras-Chave

aprendizagem profunda, aumento de dados de audio, reconhecimento de emoção na música, reconhecimento da variação da emoção na música

This page is intentionally left blank.

Contents

Glossary

Adam Adaptive Momentum Estimation. 17

AUC Area Under the Curve. 28

CNN Convolutional Neural Network. xiii, 3, 5, 18, 19, 21, 26, 27, 32, 34, 60

CRNN Convolutional Recurrent Neural Network. 26, 27, 32, 33

DBLSTM Deep Bi-directional Long Short-Term Memory. 33

DL Deep Learning. 2, 3, 4, 5, 17, 19, 26, 28, 35, 60

DNN Dense Neural Network. 18, 28, 29, 59

FC Fully Connected. 34

GAN Generative Adversarial Network. xiii, 3, 5, 19, 40, 60

GEMS Geneva Emotion Music Scale. 10

GP Gaussian Process. 25, 32

GRU Gated Recurrent Units. 27

ISMIR International Society for Music Information Retrieval. 2

LSTM Long Short Term Memory. 3, 21, 33, 34, 58

MER Music Emotion Recognition. v, 1, 2, 3, 4, 5, 10, 12, 20, 24, 25, 26, 27, 28, 30, 32, 33, 34, 35, 37, 61

MEVD Music Emotion Variation Detection. 1, 2, 3, 4, 21, 25, 30, 32, 33, 35, 56, 57, 61

MIR Music Information Retrieval. 1, 2, 14, 24, 26

ML Machine Learning. 2, 3, 4, 5, 24, 25, 26, 28, 35, 53

RMSE root mean square error. 33

RNN Recurrent Neural Network. 3, 21, 27, 59

SGD Stochastic Gradient Descent. 17, 41

SOA State of the Art. 3, 4, 5

SVM Support Vector Machine. 24, 25, 28, 29, 32

SVR Support Vector Regression. 32, 33

This page is intentionally left blank.

List of Figures

List of Tables

This page is intentionally left blank.

Chapter 1

Introduction

Never before has music played such a persistent role in our day to day life. The technological evolution within the mobile devices powered the presence of music in our daily routine and this digital era provided a way for music to be stored and consumed like never before.

Music Information Retrieval (MIR) came as a need to compartmentalize the growing library of digital music and provide more advanced, flexible and user-friendly search mechanisms, adapted to the requirements of individual users. In recent years, with the introduction of streaming services, this already massive database of songs registered a significant growth as technology enabled more artists to publish their work to a larger audience. For example, Spotify, the major audio streaming platform at the moment, registered 40 thousand songs being uploaded daily and, as of 2018, 140 million daily active users [2]. This ever growing database propels the attention given to this research field, as its use is immense, from playing or queuing a song based on the mood of the user for entertainment purposes or even from a health standpoint [3].

Music in general has the capability to trigger a memory buried deep in our brain and bring out the most overwhelming and powerful emotions. This is, for many artists, the core intent of producing this form of art. Emotion is for many of them the integral part of a song, the reason behind it. For that reason, it only makes sense to be able to distinguish and categorize music in that way, enabling users to select what to hear based on the feeling that it tries to convey.

There is no agreement among music psychology researchers on a standard approach or definition for emotion in music. It can be said that Music Emotion Recognition (MER) is a growing and exciting new field with a very difficult task and part of this is due to not being able to exactly pinpoint the components that affect the emotional part of it. MER focuses on the emotion present in the song itself, what is perceived, independent of the listeners mood, in other words, it stays as objective as possible by analyzing the song itself, on what it is trying to convey to the listener rather on what the listener feels when listening to it (see Section 2.1).

In music, one key aspect that can set apart a song is its evolution as time progresses. It is not uncommon for a song to change its emotional content after some time and the emotion it conveys varies, for example: with a sudden or progressive change of pace, a breakdown, a key change or an emerging solo. Music Emotion Variation Detection (MEVD) tackles this issue by, instead of targeting an average perceived emotion across an excerpt or a full song, recognizing that some dependency between time and musical information exists.

1.1 Problem and Motivation

MER has a wide range of uses, for example: it can be applied in the daily use of music recommendation systems, in the advertising and movie industries, and in the improvement of our well-being in humans as it can be also linked to health purposes [4] (e.g. psychological monitoring [5]).

In recent times, both studies on static MER and MEVD converged to an analysis through Deep Learning (DL) models, although many researchers still strive to improve and investigate relations between emotion tags and acoustic features. Aligned with the current research trends, our aim is to explore multiple DL approaches and to evaluate them on the existing 4QAED dataset [1]. This is a quality dataset (in terms of its diversity of styles, genres and care of annotation), despite being somewhat small for DL experiments.

Currently, MER faces two main research problems: static MER and MEVD. Static MER is focused on the classification of short audio clips (i.e. 30 seconds) with an uniform and unique emotion tag. On the other hand, MEVD addresses the problem of detecting emotion variation typically in complete songs, finding segments with uniform emotion and automatically classifying each of them.

MEVD is not as well studied as the static MER research field, having some key problems such as the absence of large quality datasets. It requires much more time detecting and labeling continuous changes in a song maintaining a high quality level, as quantity seems to prevail in this field.

The use of DL in this research field is being rapidly adopted as traditional Machine Learning (ML) supports itself on feature extraction, which is an expensive and difficult process with certain levels of uncertainty of whether the extracted features are related to musical emotions. The trend to use this popular method is present in the majority of research fields after its massive success in Computer Vision and MER is no stranger to it. In 2010, at the ISMIR conference, there were only two deep learning models present; six years later this number grew to sixteen and is fair to say that the role of DL is of high regard in MER and MIR [6].

As previously stated, it is very hard to design and compute features to retrieve some information from music although, as seen in Chapter 2, a huge effort is being put into it and the results are there to prove it. We can use DL in our advantage by automatically retrieving features from music itself, skipping the high effort process of feature design. This is our main objective, to build upon already existing extensive work with the intent to outperform classical ML models.

Another important issue with MER is the copyright restrictions as the datasets that use copyrighted music can not be passed along to other researchers that easily. The annotation process tends to fall on the researchers and it is a very time consuming and burdensome task. Later in the document we will show that the somewhat large databases also have their disadvantages (see Section 2.2), as they normally result from a combination of unofficial online annotations and tags.

1.2 Objectives and Approaches

With the initial plan and various progresses through the first and second semesters, our goal became clearer and these are the proposed objectives (see Table 1.1), ranked based on

their priority - *High*, *Medium* or *Low*, with regard to their importance and contribution, given the existing time constraints.

Table 1.1: Objectives for the second semester

Objective	**Priority**
Static MER - Evaluate existing and new DL approaches against existing traditional ML models in controlled datasets	High
Static MER -Transfer Learning with State of the Art (SOA) MIR models (i.e. genre)	Medium
MEVD - Dynamic Database creation	Low
Static MER - Continuous AV (using 4QEAD tags) traditional ML and DL approach	High
MEVD - Evaluate existing and new DL approaches against SOA models in controlled datasets	High
MEVD - GAN model for Dataset Augmentation	Medium
Static MER - End to End Raw Audio Approaches	Low

The main purpose with this study, as explained before, was to exploit several different DL approaches on the employed dataset (Section 3.1.1), as a quality reference, with structures such as CNNs and LSTMs, being the static analysis the primary focus. We also explored a hybrid solution, therefore including both carefully extracted features used in traditional ML (Section 2.4.1) and features extracted from the CNN models. As stated in the following Section 2.1, we will be using the Russell emotion taxonomy, as it allows for the use of both discrete (4 quadrants) and continuous (Arousal and Valence values) models. Regarding MEVD, a 29-song dataset previously used (Section 4.1), annotated along the entire length of the song, will be used with models such as CNNs and RNNs, more precisely LSTMs. In addition, a comparable study will be made in order to evaluate the contribution of augmented data with Generative Adversarial Network (GAN) models as well as classical audio augmentation (e.g. pitch shifting, time stretching).

Note that the previously established priorities suffered some changes with the evolution of the project: the transfer learning, did no perform as well as we hoped and not many authors made their models publicly available in order for us to experiment with them; the database creation proved to be much more complex and time consuming than expected and our focus shifted towards the GAN approach and different DL models for the static MER dataset.

Overall, the goal was to assess if a DL approach provides the same or a better result and get a sense of its advantages as well as its disadvantages over the traditional ML approaches, given their heavy computation and difficulty, in specific to the 4QAED dataset (Section 3.1.1).

1.3 Results, Contributions and Limitations of the thesis

The key results to retrieve from this thesis are:

- an F1-Score of **88.45%**, which is above the current state-of-the-art approach for this dataset;
- the same F1-Score and accuracy results as the state-of-the-art for the MEVD problem on the current dataset, which will be a primary focus in future work;
- improved F1-Score with audio augmentation and transfer learning (i.e. VGG19) approaches.

The essential contributions to come out of this project are:

- the analysis of the impact of several different architectures for the static MER and MEVD fields;
- a proposal of a DL solution which outperformed the state-of-the-art classical ML approach;
- the analysis of the impact of data augmentation on a small quality dataset regarding the static MER field.

The limitations encountered through out the project:

- a reduced amount of samples for the static MER and MEVD datasets which, as referenced in Chapter 5, are currently being expanded and will be explored in the future.

1.4 Outline

The following Chapter 2 presents a review of the state-of-the-art approaches to MER and MEVD. The different emotion taxonomies are described in Section 2.1 in addition to a critical review of the existing datasets in Section 2.2. Recent and significant progresses with respect to MER, from classical ML to DL approaches are also exposed, in both static (Section 2.4) and dynamic datasets (Section 2.5).

A summary of the performed experiments regarding the static MER problem are present in Chapter 3, accompanied by an introduction to the used datasets (Section 3.1), as well as a critical analysis of the results (Section 3.3). The same is done for the MEVD problem in the following Chapter 4.

An overall conclusion and future work guidelines are given in Chapter 5.

1.5 Organization, Resources and Planning

The purpose of this section is to: showcase the progress over time and the distribution of the time invested in each task. An high level analysis is given regarding the deviations from the expected plan, followed by the team description and an overview of the computing resources available to carry out this project.

1.5.1 Planning

The following figures represent the expected plan and the real effort put into each task over the first semester - Fig 1.1 - and over the second semester - Fig 1.2.

First Semester

The first 8 weeks of the semester were set exclusively to deal with the literature review, to educate myself and gather the core information from an extensive research in order to create a comprehensive SOA. The rest of the semester was dedicated to getting familiar with the existing MER dataset, assessing established models, experimenting with basic

DL models and transfer learning with the aim to build a strong theoretical and practical baseline.

Given some initial difficulty getting into the DL approaches in MER, and having already invested some time into researching the traditional ML aspect of the SOA, most of the time was invested in experimenting. This effort was made as a way to get a better grasp on the practical side of the project. This explains the shift of the initial plan regarding the last weeks of October.

Regarding the extended period with the DL approaches, it involved a greater amount of time than I expected and some misjudgment from my part when it came to the transfer learning model. Also, key errors on creating initial CNNs models took a significant toll on the expected plan.

Revisiting and complementing the SOA became, therefore, a necessity and a consequently easier task given the familiarity with the different ML and DL models used.

It is important to address that some weeks, primarily the sixth, seventh, thirteenth and fourteenth, were heavily affected by other side projects from the remaining classes, which were not expected.

Second Semester

The decision to extend the final deadline was the culmination of a couple of factors. One of them being the several inconsistencies with the server used for the experiments (Section 1.5.3), which delayed the project significantly. The other major factor was the opportunity to dive deeper into different approaches (i.e. GAN model for dataset augmentation, audio augmentation, separate voice from audio), which, given the exploratory nature of the project, made the most sense.

The end-to-end task did not worth the time, as early experiments demonstrated a poor performance. On the other hand, transfer learning had very interesting results so a larger chunk of time was given to it and given the lack of public models, a late strive was made. The GAN expected time period (1 week) gives a good example on how far from reality (5 weeks) the estimated time for some tasks was, as does the expected time period for the DL approaches to the 4QAED dataset.

Additionally, it is important to bear in mind that the added time period made it possible to experiment with models that achieved the best results, because the gained experience posed as a huge advantage, being able to improve on early designed models and strategies.

Overall, it can be said that the effort required to perform all experiments was extremely underestimated, as seen in the plan (see Figure 1.2).

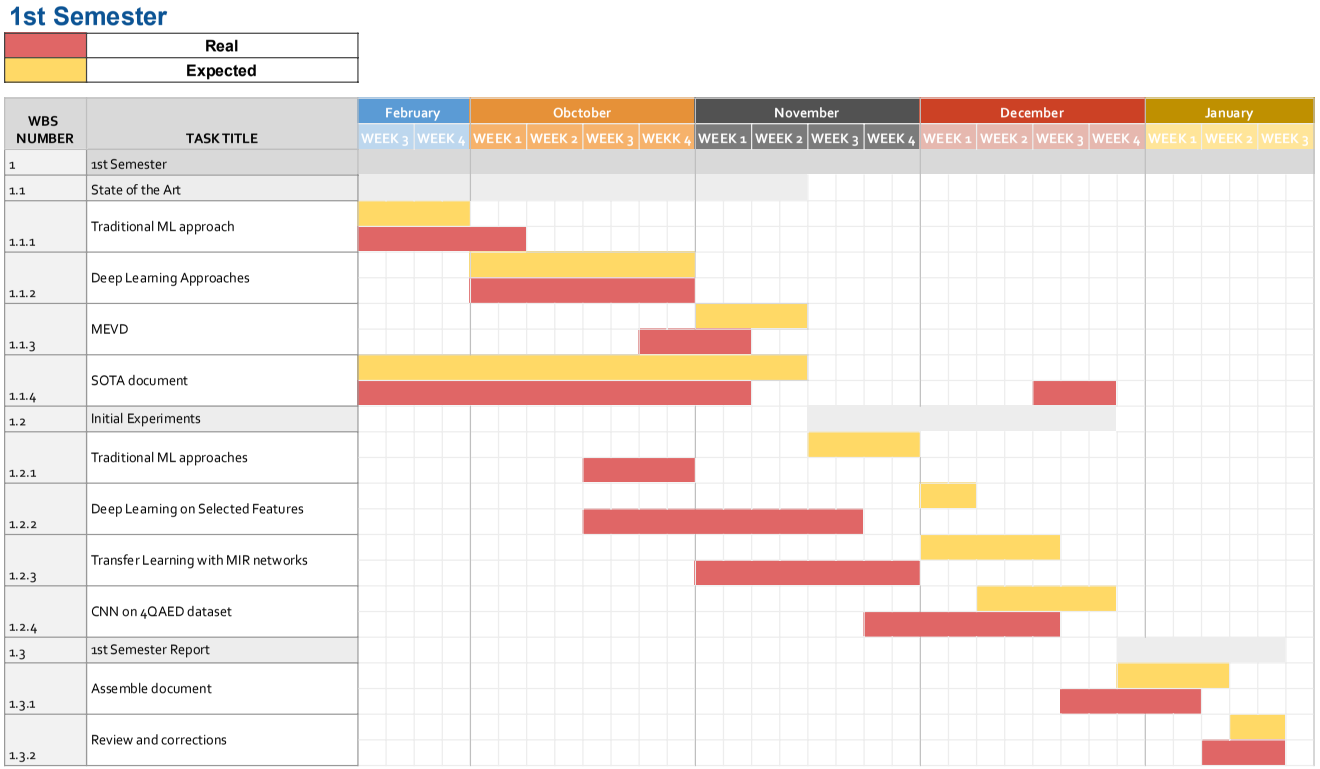


Figure 1.1: Gantt Diagram - First Semester

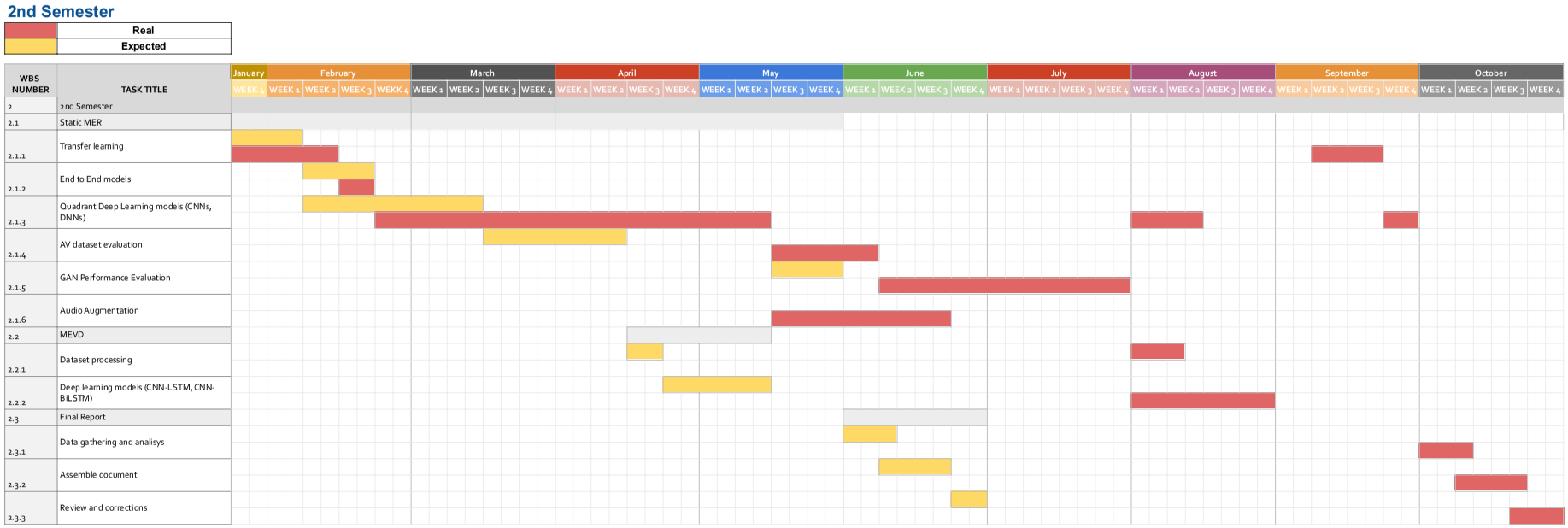


Figure 1.2: Gantt Diagram - Second Semester

1.5.2 Team

The team responsible for this project is composed by 4 elements:

- **Pedro Sá** is an MSc student from the University of Coimbra, where he concluded his Bachelor's degree in Computer Engineering and is currently pursuing a MSc degree in Intelligent Systems with interest in Music Emotion Recognition.
- **Renato Panda** as the thesis co-advisor, is a PhD from the University of Coimbra, where he also concluded his Master and Bachelor degrees. He currently is an Auxiliary Researcher at the Smart Cities Research Center, from the Polytechnic Institute of Tomar. He is a member of the Cognitive and Media Systems group at the Center for Informatics and Systems of the University of Coimbra (CISUC). His main research interests are related with Music Emotion Recognition (MER) and Music Information Retrieval (MIR).
- **Ricardo Malheiro** is a PhD from the University of Coimbra, where he also concluded his Master and Bachelor degrees, respectively in Informatics Engineering and Mathematics. He is a former Professor at Miguel Torga Higher Institute, Coimbra. He is also a member of the CMS research group at CISUC. His main research interests are in the areas of Natural Language Processing, Detection of Emotions in Music Lyrics and Text and Text/Data Mining.
- **Rui Pedro Paiva** as the main thesis advisor, is a Professor at the Department of Informatics Engineering of the University of Coimbra, where he concluded his Doctoral, Master and Bachelor degrees in 2007, 1999 and 1996, respectively. He is also a member of the CMS group at CISUC. His main research interests are in the areas of MIR and Health Informatics. The common research hat is the study of feature engineering, machine learning and signal processing to the analysis of musical and bio signals.

1.5.3 Server and Environment

All models were trained and tested in the server shared by the team that has:

- Intel Xeon Silver 4214 CPU @ 2.20GHz × 48
- 3 x NVIDIA Quadro P500 16GB

All models ran exclusively on the GPUs (Graphics Processing Unit) which saved a great amount of time as compared to the CPU performance. The primary language used was Python. With the main libraries being *tensorflow*, *keras* and *scikit-learn* and Matlab was also used, especially when dealing with large matrices and operations regarding them in order to save time (i.e. getting the average and covariance matrix for the GAN input) and to perform the statistical tests on the results.

It is important to note that, although very capable, the server is not always entirely available. The CUDA (Compute Unified Device Architecture) library was utilized to take advantage of the GPUs but it does not always know how to properly streamline the process. In other words, we have to manually dictate which and how many GPUs we will use and during the training process (which some of the tests took up to a few days) they remain 'locked' and unusable by any other user. This obviously can present itself as a big issue as sometimes there are no GPUs available and some training processes even got corrupted.

Chapter 2

State of the Art

This section serves as a literature review from a critical standpoint referencing the landmarks and significant research in the field.

2.1 Emotion models

As stated by Kleinginna et al. [7]: "Emotion is a complex set of interactions among subjective and objective factors, mediated by neural/hormonal systems, which can give rise to affective experiences such as feelings of arousal, pleasure/displeasure; generate cognitive processes such as perceptually relevant effects, appraisals, labeling processes; activate widespread physiological adjustments to the arousing conditions; and lead to behavior that is often, but not always, expressive, goal-oriented, and adaptive". This definition is a result of 92 compiled and analyzed ones, being it such an ambiguous topic. A more broad perspective, as the Merriam-Webster dictionary describes it: "a conscious mental reaction subjectively experienced as a strong feeling usually directed towards a specific object and typically accompanied by physiological and behavioral changes in the body".

Overall, emotions can be seen in three major categories: **perceived**, **induced** and **expressed**. We tackle this problem focusing ourselves in the perceived form as it is what is present in the song itself. Induced emotion is the emotion felt by the listener which is highly subjective, depending on his state of mind. Bear in mind that this is not a straightforward topic as one artist can try to convey a certain emotion (expressed emotion) such as sadness and the song might come out as calm and serene (perceived emotion) possibly inducing the listener to a state of happiness (induced emotion) [8].

Therefore, it is very important for the listener to be able to make this assessment. In other words, be capable of distinguishing his emotional response to the music (induced emotion) from the presence of emotional content in the music (perceived emotion) as it is a separate perceptual-cognitive process [9].

As exposed before, there is no real standard model to evaluate emotion in music as there is no consensus in the psychology field. There is no uniform agreement on how many categories should be considered or even if it should be discrete or continuous [10]. Even when discussing simple terms such as emotion, mood or affect, there is no unanimous agreement [9].

We can summarize the existing models as discrete (categorical) and dimensional (continuous).

2.1.1 Discrete

In 1992, Ekman et al. stated that an emotion could be expressed by a limited amount of basic emotions as a discrete basic model composed of labels like anger, sadness and happiness [11], arguing that every emotion is perceived by an independent neural system. This model soon failed to be supported in MER, as it was also based on facial expressions.

A standout model was presented in Scherer et al. [12]: the nine-factor Geneva Emotion Music Scale - GEMS. This resulted from several studies and in the experiments it outperformed the existing discrete and dimensional models. However, this can be refuted as they support its performance on classical music only [13] and were focused on induced emotions.

Hevner's model [14] consists of eight clusters of adjectives, having a close meaning, layed out in a circular form with their distance apart addressing their similarity. This conclusions came from an experiment, involving, for the most part, classical music which poses as an issue not being diverse. Constructing this clusters restricts the entire universe of emotions to this eighth categories, and from a semantic standpoint, it is possible to not fully understanding its separation as some emotions tend to overlap (i.e. joyous, playful).

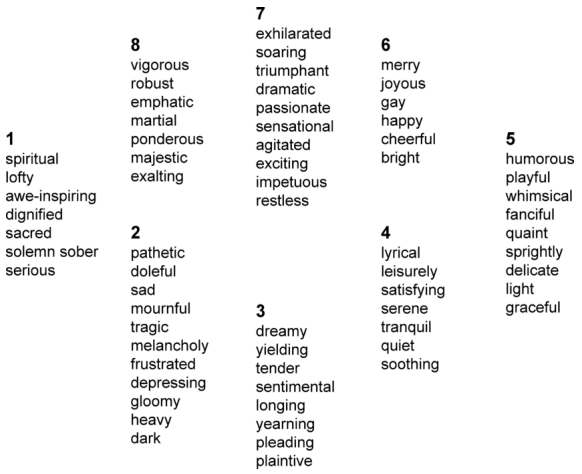


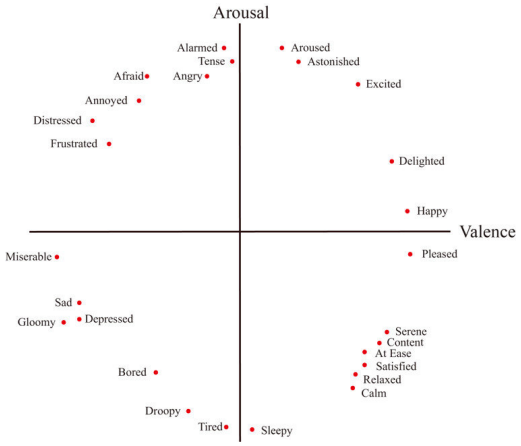
Figure 2.1: Hevner's emotion clusters

2.1.2 Dimensional

A dimensional approach can be preferred to a discrete one, as the later brings out ambiguities in the annotators interpretation of the discrete values. Many authors claim that a higher amount of annotators reduces the subjectivity in the annotation, however it can not reflect the proximity or disparity between discrete categories, as one cannot truly quantify it.

Russell [15][16] affirms that our emotion perception can be divided into two neurophysiological responses: **arousal** and **valence**, dividing the plan into 4 major quadrants (Fig. 2.2b). In cross-cultural emotion recognition, due to the difficulty of using equivalent emotion adjectives in all languages, less categories (i.e. quadrants) are more consistent [17].

Another important point is that, arousal can be related to tempo (fast/slow), pitch (high/low), loudness level (high/low), and timbre (bright/soft), valence is related to mode (major/minor) and harmony (consonant/dissonant) [9] and this is the main reason why, from a MER point of view, the arousal detection capability is by the most part, superior given it is based on pace and power. Using this same model, Meyers [19] divides the Russell circumplex plane



(a) Russell model and quadrants

Quadrants	Emotions
Q1 (A+V+)	Joyful activation
	Power
	Surprise
Q2 (A+V-)	Anger
	Fear
	Tension
Q3 (A-V-)	Bitterness
	Sadness
Q4 (A-V+)	Tenderness
	Peace
	Transcendence

(b) Quadrants emotions overview (adapted from [18])

Figure 2.2: Russell model quadrants

into 8 equal slices with the distinct tags: arousal, excitement, pleasure, relaxation, sleepiness, depression, displeasure and distress (see Figure 2.3).

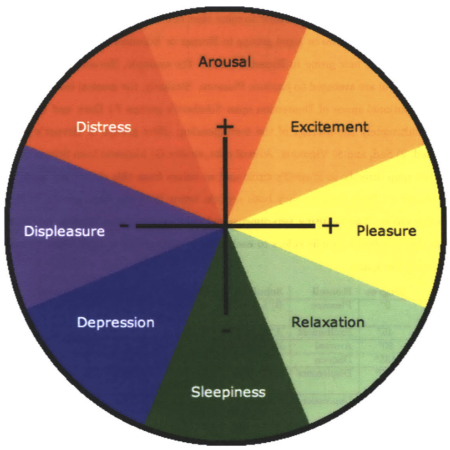


Figure 2.3: Russell model with 8 categories (adapted from [19])

Thayer [20] and Vieillard et al. [21] also propose a multidimensional approach having their primary dimensions as energetic arousal, tension arousal and energy and tension, respectively.

As of today, the major model and the one that gained more supporters is the one that this study focuses on, the Russell AV (Arousal/Valence) model, as it offers a higher consistency [13]. This model will be tackled from a continuous and discrete point of view, as we have two dimensions that can be separated into 4 separate quadrants.

It is pertinent to mention the difficulty that researchers face as so many different authors follow divergent emotion models that makes a comparison a somewhat complex task to perform, which is aggravated in the databases that each supports and uses (See Section 2.2).

2.2 Music Emotion Databases

Huq et al. [4] state that dataset quality is comprised by 3 factors: the number of ratings per clip, the number of clips in the dataset and the setting in which the data is collected. To these quality criteria we could add others, such as dataset balanced, coverage, diversity and annotator agreement rate.

It is also very important to address how the clip is classified and the criteria for class agreement, as being humanly annotated brings out discrepancies. Collecting data in this manner is a time-consuming and labor-intensive process, which limits the feasibility of growing the number of tracks and reflects our decision to emphasize quality over quantity when creating the dataset. This is opposed to huge datasets that generally focus on quantity over quality, being that the result of a narrow selection of music clips, small number of annotators or their low confidence and lack of music background. A so called artificial approach when creating a new dataset is the attempt to select music clips for particular emotions, manually selecting this or that clip for the perceived emotion in contains. This can be supported or refuted by many authors and has its advantages: being able to get a strong representation for each targeted emotion with the hope to extract strong features unique/related to them; and its disadvantages: not being a true representation of a real world scenario.

Several researchers opt to create a very focused dataset, for instance a creation of a Turkish only dataset or involving only one genre such as K-Pop. However, since the purpose of MER is to facilitate music retrieval and management in everyday music listening, analyzing the emotional content of popular everyday music is our main focus with this project. One key aspect to keep in mind is that a significant portion of the datasets used are labeled with tags that are not focused in emotions as it can portray a certain genre or other key aspect perceived by the annotator as it is a free choice (i.e. energetic, calm, exciting).

The following table encapsulates the primary aspects of these datasets and a more thorough analysis is made after.

Table 2.1: Databases Overview

Name	**Approach**	**Features/Data**	**Duration**	**Emotion Taxonomy**	**Size**	**Possible limitations**
DEAM	Dynamic/ Static	Sound Signal, Metadata (Artist, Song name, Duration and Genre)	45 seconds song excerpts and 0.5 seconds	AV values	1802 clips	Random starting points, 45 seconds may be to large for one emotion, low agreement rate.
Million Song Dataset	Static	Sound Signal (majority), Extracted Features (i.e. tempo, energy, key), Metadata (Artist, Song name, Lyrics, Genre, ...)	30 seconds song excerpts	Tags (freely written)	1000000 clips	Unreliable annotation quality, combination of diferent datasets
CAL500exp	Dynamic	Sound Signal	3 and 16 seconds clips	67 Tags	500 songs (divided into 3 and 16 seconds clips)	Based on 67 tags from the previous CAL500 dataset
Soundtrack	Static	Sound Signal	15 seconds clips	7 emotion tags and bipolar values for valence, energetic arousal and tense arousal	360 clips	Clips were retrived from 110 films, the size is not desirable for the low diversity
Bi-Modal	Static	Sound Signal, Lyrics	20 seconds clips	4 AV quadrants	163 clips	Although carefully picked and filtered, it has a small size
4QEAD	Static	Sound Signal, Metadata (Genre, original emotion tags and respective AV values)	30 seconds clips	4 AV quadrants	900 clips	Random clips in song
1000 Songs for EAM	Dynamic/ Static	Sound Signal, Extracted Features (i.e. MFCC, Timbre Features)	45 seconds clips (divided into 0.4 seconds segments annotations)	AV values	1000 clips	Dynamic annotations to concentrated on the center of Russell's AV model

The DEAM[1] dataset consists of 2013 to 2015 'Emotion in Music at the MediaEval Multimedia Evaluation Campaign' datasets, to evaluate the models performance. It is composed of 1802 clips with both static 45 seconds annotations and dynamic ones, converted to a rate of 2Hz (labeled each 0.5 seconds). The emotion model used for annotations is the continuous numerical values representation of Russell's AV (Arousal/Valence) [16] model. The 45 seconds excerpts are all re-encoded to have the same sampling frequency (44100 Hz) and are extracted from random starting points in a given song. This raises a potential issue as, for example, selecting the intro of a song that can be complete silence. The averaged annotations and their respective standard deviations are provided so that one can have an idea on the margin of error. As stated by Vale et al. [8] 45 seconds is too long for static annotations as longer samples tend to not encapsulate a single emotion. The dataset's mean of the agreement rate is 0.47 which shows that most songs had problems with disagreement between annotators. The annotations in question were a result of crowdsourcing using Amazon Mechanical Turk. The authors state that a procedure was developed to filter out poor quality workers but the agreement rate, as pointed out by Vale et al. [8], is not the best representation of that. Therefore it is recommended a sample selection based on the agreement rate prior to its utilization.

A known database in MIR is the Million Song dataset[2], with nearly a million song entries is the largest database in MIR and its main goal is to encourage research on MIR solutions. It is a collection of audio features and metadata but the majority of the audio files can also be retrieved as it is a cluster of complementary datasets contributed by the community. Annotations are therefore retrieved from tags assigned by a large community of music listeners as an option, instead of professionals or even listeners educated to do so. This can raise an important problem as common listeners do not take their time to form a well thought out response. This is why many researchers tend to to follow the AllMusic tags instead, being them made by experts [1].

CAL500exp[3] is an enriched version of the CAL500[4] dataset, as it segments the original music clips into 3 to 16 seconds, This was done based on a audio-based segmentation that allowed to identify acoustically homogeneous segments by clustering them in order to select representative segments for annotation, making the connection between tags and music better-defined. They also recruited eleven annotators with strong musical backgrounds for better annotation quality [22]. The final dataset uses 67 of the original CAL500 labels instead of offering the option to users to freely create a tag which inevitably adds consistency however, it makes it dependent on the previous dataset limiting the annotators options.

The Soundtrack database[5] (used by Er et al. [23]) is composed of 110, 15 seconds clips, extracted from 60 movie soundtracks from 1958 to 2006. These excerpts were annotated in a listening experiment by 116 non-musicians but the segments were handpicked by experts with the intention to represent discrete emotions such as happiness, sadness, fear, anger, surprise and tenderness and 3 categories defined by valence, energetic arousal, tense arousal. Attention was given so that no lyrics or sound effects are present. The original 360 clips suffered a selection process and 110 clips were chosen. Although being non-expert students, the experiment itself was very controlled with each subject rating clips from 1 to 7 in terms of discrete emotions, as well as using a bipolar scale for the dimension ones (valence, energetic arousal, tense arousal). All the volunteers were subjected to similar

[1] http://cvml.unige.ch/databases/DEAM/,

[2] https://labrosa.ee.columbia.edu/millionsong/,

[3] http://slam.iis.sinica.edu.tw/demo/CAL500exp/,

[4] http://calab1.ucsd.edu/~datasets/cal500/,

[5] https://www.jyu.fi/hytk/fi/laitokset/mutku/en/research/projects2/past-projects/coe/materials/emotion/soundtracks/Index,

conditions, using studio quality headphones in a soundproof room. The two issues with this dataset are its small size however, it is the result of a very thought out experiment, and the fact that is from movie soundtracks can be refuted as not diverse enough.

Also in [24] the Bi-Modal[6] dataset was used, being this a combination of lyrical and audio information. It was used only for its audio and the associated labels. A total of 200 clips, 20 seconds each, were in the initial dataset from a variety of genres which were annotated by 39 individuals. The perceived emotion was captured by assigning values from -4 to 4 to both valence and arousal dimensions and converted to the 4 quadrants from the Russell Model [16]. The final values were the mean of all subjects and the clips with higher standard deviation were discarded as well as kept in mind the agreement between the annotators. The final product consists of 133 audio clips. As the previous dataset, its main problem is its small size.

The 4QEAD (4 Quadrant Emotion Analysis Dataset) [1] is a 900 30 seconds clips, each from a specific song. The used emotion taxonomy was the aforementioned Russell AV quadrants [16]. The clips were retrieved from the AllMusic platform as well as their mood tags , which were then mapped to their quadrants according to Warriner's adjectives list that contains a value for arousal and valence for each 200 out of the 289 AllMusic claimed emotion tags. This dataset also contains its metadata (i.e. genre, emotion tags as well as its AV values). A similar approach was taken later in this project from a continuous emotion problem standpoint (see Section 3.2.2). After some filtering 2200 song clips were achieved after the quadrants and the genres in those quadrants were balanced. In the pursuit for a better quality dataset, a blind check was applied and all clips with poor quality and unmatched AllMusic annotations were discarded, resulting in a 900 clip dataset. The dimension of this dataset is somewhat small. However, it went through a significant filtering stage given the semi-automatic process behind its retrieval. One possible issue is the lack of knowledge from the AllMusic platform, as it is not stated how the 30 seconds clip is chosen.

The 1000 Songs for Emotional Analysis of Music [7] consists of 1000 clips both static and dynamically annotated by over 10 subjects. The annotations follow Russell's AV model with continuous values. The clips are continuously labeled and the annotations are extracted with a 2 Hz rate. A strong filtering phase was employed, as potential participants were tested for their capability perceiving the emotion present in the song as well as the genre, arousal variation in a test set and even a demographic background was considered. All of this was taken into account and annotations from low confidence annotators were discarded. Two possible issues are the distribution of the dynamic annotations given that a significant part is concentrated on the center of Russell's model and the annotations were made using the aforementioned Amazon Mechanical Turk platform. Nevertheless, this dataset was tested heavily in the paper [25] and is used by Du et al. [26] having some promising results.

2.3 Deep Learning Explained

The purpose of this section is to introduce basic elements of Deep Learning that will be discussed later on.

[6] `http://mir.dei.uc.pt/downloads.html`,
[7] `https://dl.acm.org/doi/10.1145/2506364.2506365`,

2.3.1 Basic concept

A deep neural network is essentially a set of various layers of neurons that mathematically transform data. After a training phase, it can transform a group of inputs into the desirable output learning the relationships between them. We can divide a network in input, hidden and output layers, each composed of neurons (See Figure 2.4).

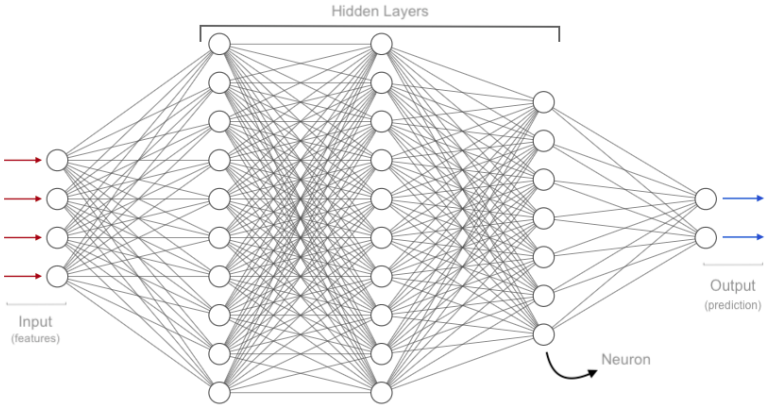


Figure 2.4: Simple Fully Connected Neural Network structure

These neurons or nodes (also referred as perceptrons) are basically a mathematical function analogous to a biological neuron. Its value is usually a weighted sum of the input followed by an activation function (See Figure 2.5). Each input value has its weight (trainable parameter) and the neuron has its bias. The result of this weighted sum is fed to an activation function (i.e. *step function* in the Figure 2.5).

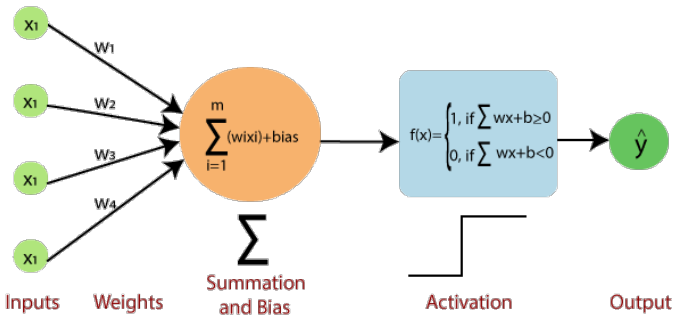


Figure 2.5: Neuron basic structure[8]

An essential part of any neural network is the loss and optimization methods used in order for the model to adapt to the data. The loss function measures the quality of the prediction compared to the actual true class, in this case, the quadrant label. Being a categorical problem (assigning a category, a label), we used the categorical cross entropy function as it measures how distinguishable the two probabilities distributions are (see Figure 2.6).

The model learns by calculating the gradient of the loss at every parameter. The optimization method (or optimizer) minimizes that loss by updating the weights of each layer, working from the end of the model to the beginning, increasing the value for the correct output node (corresponding to the correct label) and decreasing it for the incorrect output

[8] from `https://www.javatpoint.com/single-layer-perceptron-in-tensorflow`
[9] from `https://peltarion.com/knowledge-center/`

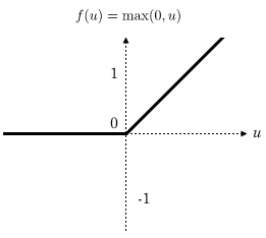
$$\text{Loss} = -\sum_{i=1}^{\substack{\text{output}\\ \text{size}}} y_i \cdot \log \hat{y}_i$$

Figure 2.6: Categorical cross entropy loss function[9]

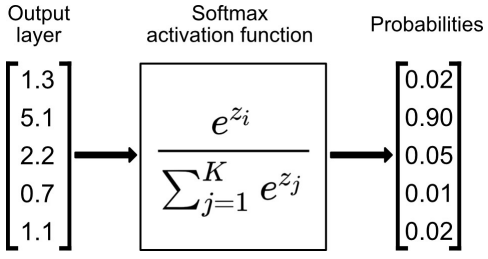
nodes. This process is known as backpropagation. It can be implemented with different types of optimizers which follow the same overall rule, following the gradient. After a batch (number of samples of training samples) goes through the network, the overall average loss and derivative with respect to the weights are calculated and the weights are updated. This update depends on the learning rate, in other words, the step size the model takes along the gradient in search for the minimum loss possible.

The chosen optimizers were the SGD (Stochastic Gradient Descent) and the Adam (Adaptive Momentum Estimation). Note that when referring to SGD, the process is what is known as Mini-Batch Gradient Descent. After a batch of samples are processed by the network, its weights are updated at a constant learning rate following the gradient at each neuron. Batch size is a hyperparameter that can and will be explored, such as learning rate and epochs (number of times the network cycles through the all training data). Adam implements an adaptive learning rate for each parameter based on previous gradients, as well as adaptive momentum, which helps the model not get stuck to a local minimum by adding to its learning rate based on previous gradient changes [27]. Adam is the most popular optimizer, as it deals well with large scale and complex models [28] but it can tend to not perform as well as other in the test data as, especially in our case, it overfitted most models as it just tends to ignore the initialized weights and adopts a very large initial learning rate adapting too well to the training dataset.

An activation function is applied to the output of the neuron in order to introduce non linear properties to the network. The explored function is the ReLU (Rectified Linear Unit) activation function (see Figure 2.8a). The one used in the last layer, the output layer, is the Softmax activation (see Figure 2.8b) function (usually used in the output layer) as it maps the probability for each possible output. The one with an higher probability is then chosen as the output class.



(a) ReLU activation function



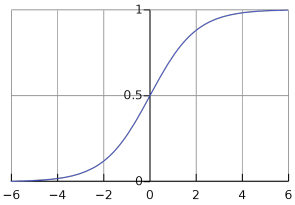
(b) Softmax activation function

Figure 2.7: ReLU and softmax activation functions

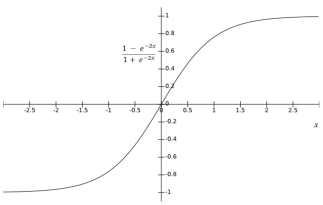
The ReLU activation function is used in every layer (apart from the last one) as it is more likely to tackle the vanishing gradient problem [29]. The vanishing gradient problem is a common issue in DL. As the error is back propagated through the network, updating the weights, it slowly decreases given the derivative of the activation function (i.e sigmoid). ReLU helps by having a constant gradient and a faster computation as it is just choosing the max value.

In regard to a regression problem, two main activation functions were used: sigmoid and

tanh. The sigmoid function keeps the output value between 0 and 1 as the tanh keeps it from -1 to 1.



(a) Sigmoid activation function



(b) Tanh activation function

Figure 2.8: Sigmoid and tanh activation functions

The depth of a neural network is the amount of layers it has. The optimal depth is very hard to predict, it depends on the problem but, typically, a larger dataset with various possible outputs tend to need a more complex and deeper network [6]. The width of a layer is the number of nodes in the layer, as for CNNs, the width is the number of feature maps (see Section 2.3.2). A classical neural network where all nodes are connected in between layers (not within the layer itself) is a Fully Connected Neural Network or a Dense Neural Network (DNN) and usually used as the classifier section of complex neural network (i.e. CNN) as it receives features that have been automatically retrieved previously.

Similar to a classical machine learning model, the network goes through a training phase where it adjusts its weights attempting to reduce the overall loss. This loss is calculated thought a loss function which measures the difference between the output (i.e. Arousal and Valence values) and the actual truth (i.e. annotated values). A method called backpropagation is responsible for updating the weights computing a gradient of the loss function. The model aims to minimize its loss and, sometimes, can overfit. Overfitting is when a model achieves a very good result for the training set but can not generalize well when faced with unseen data. Generally, a dropout layer can be used to prevent this by randomly discarding a selected amount of values.

2.3.2 Convolutional Neural Network

A convolutional neural network (CNN) is heavily used in Computer Vision as it was intended. In simple terms, it is able to detect patterns in images and has been a major step forward in various applications (i.e. self driving car capability). It receives images as input (the values of each pixel), for example, when receiving a Spectrogram, each value is the the amplitude of a given frequency at a given time. A kernel (a matrix of learnable parameters) "sweeps" the the input performing a dot product between itself and the input. The kernel can slide, for example, one pixel horizontally and vertically meaning an input with 20x20 values becomes 19x19. This sliding value is called stride and this all operation is called convolution (see Figure 2.9).

The output of a convolution operation portrays an activation map that will hopefully represent a detectable pattern, this activation map is called a feature map and the number of feature maps per layer define the width of the layer. In sum, it can be said that the convolution operation computes the correlation between the kernel and the input [6]. Going through the layers, this patterns get more and more defined.

Pooling layers are used to reduce the size of the feature maps, downsampling them. The two main methods are max pooling and mean pooling. Max pooling picks the maximum value from a defined space region (see Figure 2.10). Mean pooling, as the name implies,

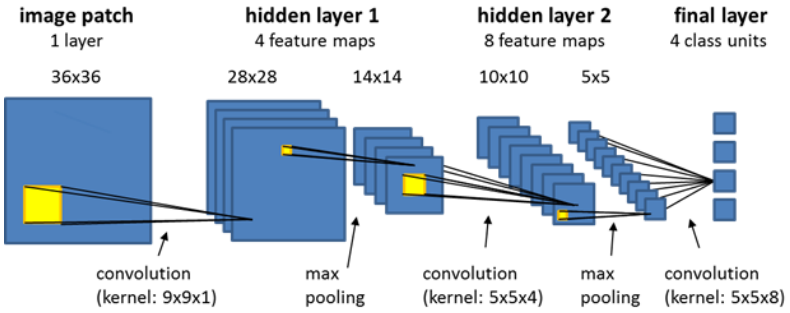


Figure 2.9: Basic CNN structure[10]

computes the mean of the selected region. The latter method is not heavily used, although some authors opt use it in the last convolutional layer. An usual CNN structure consists on a series of convolutional and pooling layers (feature extractor) followed by a fully connected neural network (classifier).

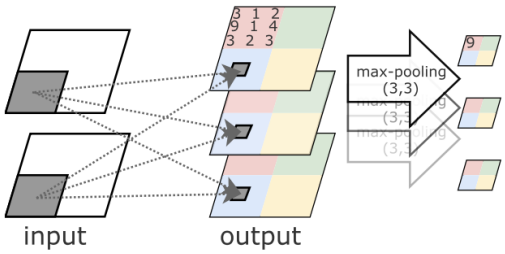


Figure 2.10: Max pooling (adapted from [30])

2.3.3 Generative Adversarial Network

With the set divided into a training and testing set, the network may not have enough data to learn from and this is where data augmentation is useful. Adding some distortion to the training set (e.g. pitch shifting) while preserving the core properties can then be advantageous. Many authors have tackled this problem by time stretching or pitch shifting a sound signal [23]. Generative Adversarial Networks (GAN) are a recent breakthrough in DL, particularly in Computer Vision (see Figure 2.11).

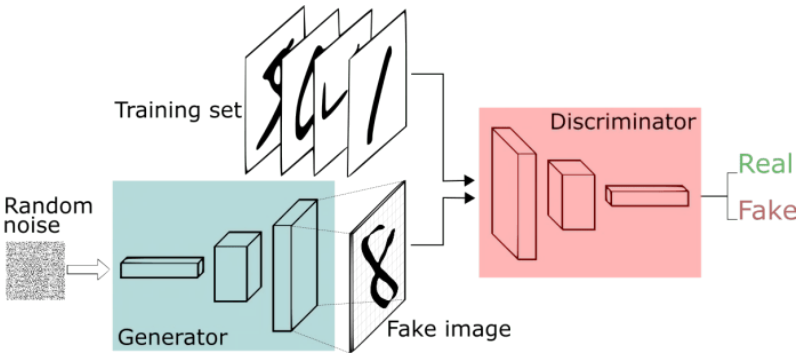


Figure 2.11: GAN[11]

[10] from `https://docs.ecognition.com/`
[11] from `https://deeplearning4j.org/`

It basically consists in two models competing with each other: a discriminator and a generator. The role of the discriminator is to receive real images (real input) and fake images (generated input) and distinguish which is which. The fake images are the result of the generator model that receives an input filled with noise and attempts to generate an image closest possible to the real one.

As both models train, both will aid one another. The main goal is to get as close to real as possible in order for the discriminator to get to a point that it can not tell the difference. This can be used to increase the size of the training set and to our knowledge, it has not been explored in MER.

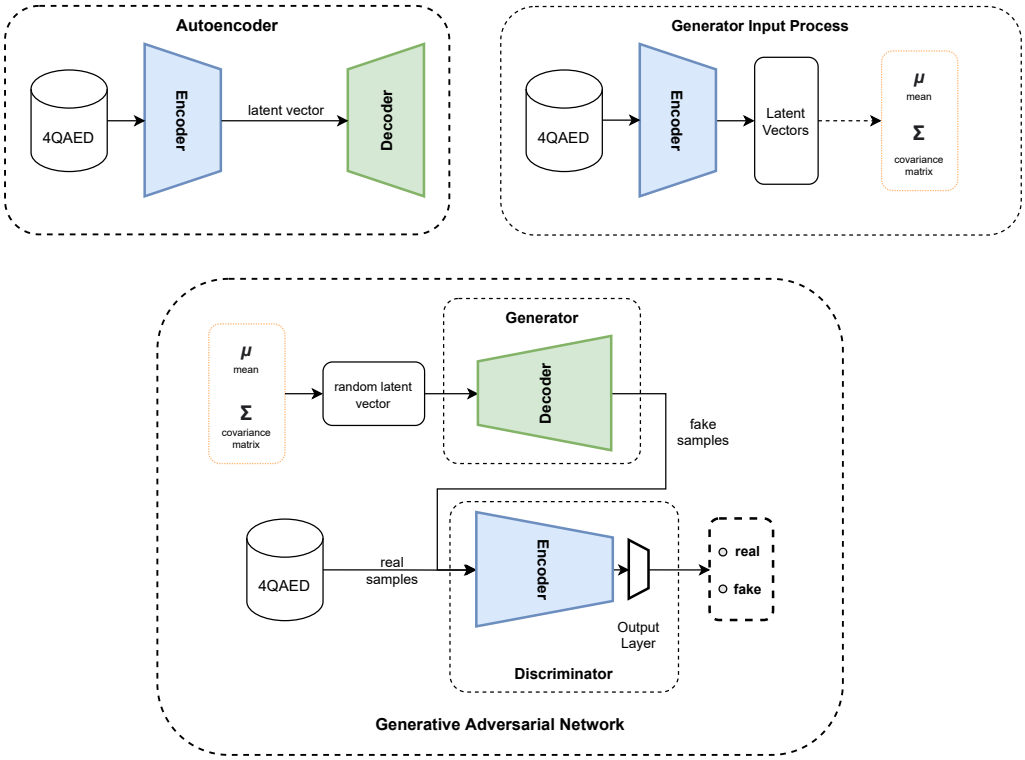


Figure 2.12: Autoencoder and GAN process

As a standard in GAN development, the activation function used is the LeakyReLU, which is a variation of the ReLU activation function previously discussed but allows for a negative input. When the input value is negative, the output is the sum product of the input with a constant.

In order to have a strong baseline to train the GAN model, and to better take advantage of a small dataset, it is common to use autoencoders [31][32]. An autoencoder follows the same structure as a GAN, it has an encoder and decoder portions. Having the same topology as the GAN model, it can be trained by using all the samples in the dataset with two main goals: provide a strong foundation for the GAN model by using the encoder portion of the autoencoder as part of discriminator and the decoder portion as the generator; also, we can take full advantage of the output of the encoder section by drawing a distribution from it, in order to generate random latent vectors to feed to the generator section of the GAN (see Figure 2.12).

2.3.4 Recurrent Neural Network

In simple terms, recurrent neural networks (RNN) allow for previous information to be fed as input in future iterations (being stored in a so called hidden state). We then assume that there is a a relationship between data samples. The network not only uses the information it learned from training but also from past inputs, so hypothetically, being the networks in the same level, the same input can produce different results depending on previous inputs (see Figure 2.13). The hidden states that contain the information are updated moving forward. In basic RNNs, as the gap between relevant information grows it becomes unable to assess it. This is the primary reason why Long Short Term Memory Networks (LSTM) came to be.

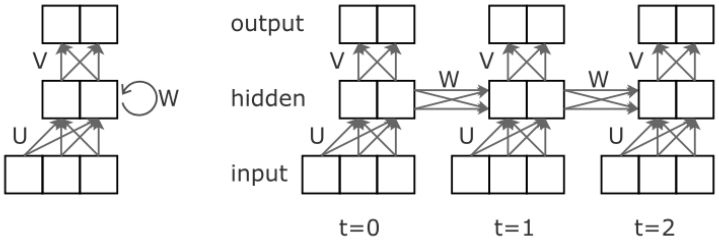


Figure 2.13: Recurrent Neural Network basic flow (adapted from [6])

In a basic RNN, the input data is combined with the data from the hidden state via a *tanh* function which we can interpret as a neuron. LSTMs adds three neurons. The backbone of the LSTM is to regulate the information that passes by with gates. Three gates are typically employed: forget, input and output, which essentially control the information going through. This decision is based on previous outputs, allowing the model to actively pick the most relevant information. A Bidirectional LSTM not only gets access to past information but also, future input samples.

As expected, RNN as a whole can be very heavy computationally speaking. However, for example for MEVD, they are a key architecture and one that was explored in combination with CNN.

2.4 Static Music Emotion Recognition

This section corresponds to an analysis on the progress and current state of research in MER as a static evaluation - singular excerpts not accounting relevant past information. Table 2.4 gathers the information retrieved and all approaches are explained in detail after.

Table 2.2: Static MER approaches

Author	**Approach**	**Database**	**Features/Input**	**Models**	**Emotion Taxonomy**	**Results**	**Observations**
Song et al.	Classical ML	2904 songs Last.FM	Dynamic (i.e. energy), Rhythm, Spectral (i.e. MFCC), Harmony (i.e. key mode)	SVM	4 emotion tags - happy, angry, sad and relaxed	0.54 Accuracy	Only pop music and tags as well as the emotion model can be contested as not supported
Markov and Matsui	Classical ML	1000 Songs for EAM	MFCC, LSP (Line spectral pairs), TMBR, SCF and SFM, CHR	SVR and GP Regression	Russell AV (continuous values)	0.69 and 0.47 Accuracy (for Arousal and Valence)	Possible limitations to the dataset used, but overall great performance
Panda et al.	Classical ML	MIREX-like	9 melodic (i.e. pitch and duration, vibrato) and 2 standard features	SVM	MIREX 5 clusters	0.64 F1-Score	Taxonomy is not psychologically supported
Panda et al.	Classical ML	4QAED	29 novel features (i.e. glissando detection) and 71 standard	SVM	Russel AV (4 quadrants)	0.76 F1-Score	A thorough process creating the new dataset and good overall results
Seo et al.	Classical ML	K-Pop 20 seconds clips	AVG height of the the soundwaves, peak average, the number of half wavelengths, AVG width and BPM	SVM	Russel AV (4 categories)	0.73 Accuracy	The dataset only consists of K-Pop songs and the project tends to hint to a detection of inductive emotion and not a perceived one
Choi et al.	DL	Million Song Dataset	Mel-Spectrogram	CRNN	50 tags (12 mood tags)	0.85 AUC	Multi tag problem, not as fitted to MER
Liu et al.	DL	CAL500 and CAL500exp	Spectrogram	CNN	18 emotion tags	0.59 F1-Score	It is not explained how the 18 emotion tags are retrieved and which architecture had the best results
Liu et al.	DL	1000 Songs for EAM	Spectrogram	CNN	Russel AV (4 quadrants)	0.72 F1-Score	Has some discrepancies regarding the architecture

Table 2.2: Static MER approaches

Author	**Approach**	**Database**	**Features/Input**	**Models**	**Emotion Taxonomy**	**Results**	**Observations**
Seo et al.	DL	K-Pop 20 seconds clips	AVG height of the the soundwaves, peak average, the number of half wavelengths, AVG width and BPM	DNN	Russel AV (4 categories)	0.72 Accuracy	The dataset only consists of K-Pop songs and the project tends to hint to a detection of inductive emotion and not a perceived one. Used only as a classifier
Cañón et al.	DL	Speech Recognition + 4QAED	Spectrogram	CNN	Russel AV (4 quadrants)	0.48 F1-Score	Transfer learning approach outputting decent results indicating a co-relation to speech
Er et al.	DL	Turkish 500 clips and Soundtrack (data augmented)	Spectrogram	AlexNet and VG-16 (CNNs) (+ SVM)	4 emotion tags - happy, angry, sad and relaxed	0.76 F1-Score	Data augmentation process makes the process biased and unrealistic
Sarkar et al.	DL	Soundtrack, Bi-Modal and 4QAED	Mel-Spectrogram	CNN (adapted from VGGNet)	Russel AV (4 quadrants)	0.83 F1-Score (4QAED)	The datasets are split into 5 seconds clips but it is not explain how to annotations follow this division

2.4.1 Classical Machine Learning Approaches

MIR started as a traditional ML problem and for that, an essential part of the solution would be a strong and rich selection of features as they are the main ingredient for a great performing model. MER is a solid proof of this as since 2003 more and more features are conceptualized and extracted in order to outperform the latest state of the art model.

The overall process behind classical ML approaches in MER is described in Figure 2.14. An annotation process is undergone to label the audio clips to the emotion taxonomy adopted, this being discrete or continuous as mentioned above. A set of features is carefully extracted, processed and with both features and annotations, a model is trained which is than evaluated in the test set.

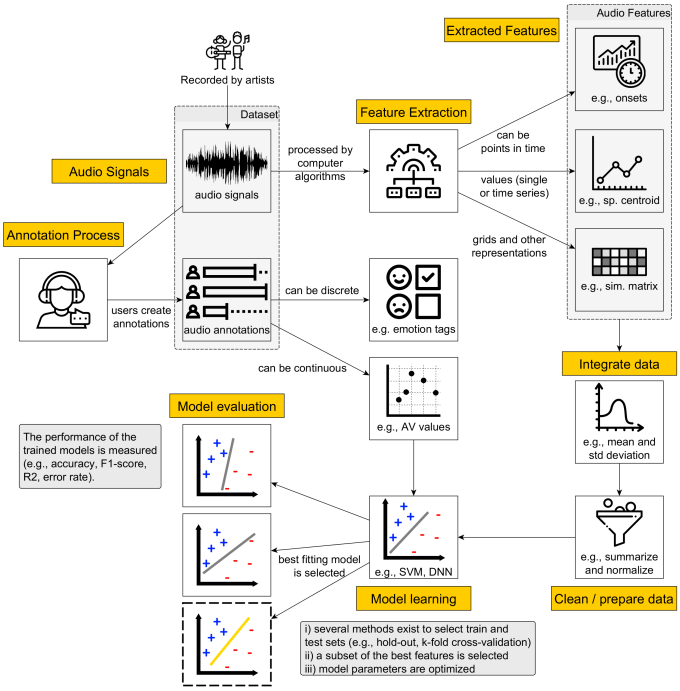


Figure 2.14: Classical ML approach to MER (adapted from [33])

Meyers [19] came up with a two way approach as he considered information from the audio and lyrics components in music. The objective was to generate a playlist around the listener being the mood he experienced or the main activity in question the main focus. The emotion model targeted was a mapped version of the Hevner's adjective model [34] to Russell's [16]. Features as mode, harmony, tempo, rhythm and power were extracted which, combined with the lyrics information, made up for some good results as stated by author with comparison to the music experts opinion. This model was designed with a 372 song database and the results were compared to experts opinions. One important aspect to mention is the use of a decision tree followed by a K-nearest neighbors for the classification of the songs that, added to the lyrics emotional value, made up for some miss classifications. The evaluation of this model is not quantified although it suggests a good performance comparing the output label with All Music Guide experts tags.

Song et al. [35] used a Support Vector Machine (SVM) with a 2904 pop song database with four emotion words associated ("happy", "sad", "angry" and "relax") collected in The Last.FM platform. These tags were used as the emotion model and it is fair to say that this model primarily focuses on induced emotion and not perceived ones. They showed that combining spectral, rhythmic and harmonic features improved the accuracy. As the

database used consisted of pop songs only, it is hard to say that this system provided some good results as it did not encompass a wide range of genre/sounds. Additionally, the emotional model followed, as well as the tags retrieved, do not form a controlled environment.

Markov and Matsui [36] proposed a Gaussian Process (GP) approach that, according to the results, showed a better performance over standard SVMs. GP is not widely used in ML but Markov and Matsui tested its behavior in MER. It basically consists in assigning a probability to possible regression functions and based on its distribution we are able to give a level of confidence to a solution [37]. Noted that it can also be used in categorical classification tasks. In this case the emotion model used was the Russell's [16], outputting the arousal and valence values for each clip with the Medieval 2013 database being utilized (composed of 1000 45 seconds clips from random locations in songs from 8 different genres, uniformly distributed from a wide range of unique artist and manually annotated). Results showed an improvement over SVMs and overall better performance in the arousal section (0.69 accuracy) as expected given that, as noted before, it is an easier and more accurate distinction being based on loudness and rhythm.

Panda et al. [38] proposed a model combining the use of standard audio and lyrics features with melodic ones extracted from the audio dataset in attempt to break the so called glass ceiling, which is a problem in the MER community demonstrated for instance by the annual Music Information Retrieval Evaluation eXchange - Audio Mood Classification (MIREX) task where the results have stagnated in recent years in recent years [39]. This dataset was obtained by mapping AllMusic tags to five different clusters, summing up to 903 30 seconds clips reasonably diverse and uniformly distributed across all five categories. Feature selection was used and so the best result was obtained by using only 11 features, melodic and standard by a SVM model with 0.64 F1-Score.

Novel audio features were introduced [1] with the results supporting a 9% improvement (76.4 % F1-Score) over existing ones across all quadrants. This was achieved with 100 features, 29 novel and 71 standard ones. This contribution was accompanied by a 900 clip dataset (4QAED) creation by mapping existing AllMusic tags mapped to Russell's quadrants followed by a validation phase where clips were discarded for bad quality and no agreement among annotators.

Seo et al. [40] contributes to the MER field with the development of SVM, Random Forest and even a simple fully connected neural network (see Section 2.5.2) models to tackle MEVD, in order to contribute to intelligent IoT Applications such as personal voice assistants. Features like average height of the the soundwaves, peak average, the number of half wavelengths, average width and BPM are extracted, which is unusual in the field as it does not fully encapsulate the overall emotional information in a song. The annotations are based on the arousal and valence dimensions using 20 seconds clips from 2 minutes songs. Arousal and valence values from -100 to 100 were annotated by the participants, which were mapped to four categories: happy, glad; calm, bored; sad, angry; excited, aroused, and put these categories in between the first and fourth; fourth and third; third and second; and second and first quadrants respectively (see Figure 2.15). The annotators had no specific background on music but the study was comprised of 39 subjects in various ages. The dataset consists only of K-Pop songs, which is a downside. It is not stated the amount of samples used in training but 35 unseen K-Pop songs were used as a test set and the SVM model had an agreement rate of 73% with all annotators. An important aspect is that although is not stated in the paper itself, the project tends to hint to a detection of inductive emotion and not a perceived one as it refers to the annotators experience.

Lower agreement tends to indicate a lower score and that was the case for the third and

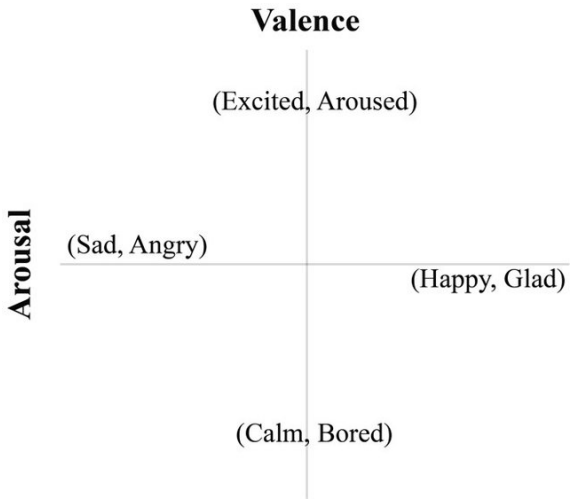


Figure 2.15: Russell Model with 4 categories (adapted from [40])

fourth quadrant (Russell's [16]). This is something noticed in the community as these quadrants share such a similar musical pattern. This is one of the reasons why deep learning is such an interesting approach as a good model can give such a rich insight to some unseen/underlying connections.

Very recently, a survey was made approaching the general audio features used in MER [39]. Recent studies support themselves on existing standard features (i.e. already used in other MIR task) and improved machine learning techniques and although this attitude, towards a better performance in the field is a valid one, a focus on capturing the emotional content conveyed in music through better designed audio features is interpreted by the authors as the main and crucial strategy.

As this short and concise review shows, traditional machine learning heavily supports itself on the quality of the features it uses and sometimes can fall towards a model that is not capable of assessing a diverse dataset when put to the test.

2.4.2 Deep Learning Approaches

Building on previous observations, traditional ML approaches rely on the quality of hand crafted features and this is one of the reasons why, in recent years, there has been a shift to DL techniques across all research fields. In MER we as human beings, have a distinct sense of what is an emotion and what sets of that particular emotion and finding acoustic features in music that significantly contribute for this detection is a very challenging task.

Also, an important aspect in the MER research field is that most researchers tend to prepare their own dataset which is always a great contribution, however it can be considered as too precise and not broad in the way that they do not gather sufficient diversity to be applicable to other significant model contributions or in a real world scenario.

The first deep neural network used in MER, to our knowledge, was presented by Feng et al. in 2003 [41]. It consisted on a 3 layer feedforward neural network and rely solely on tempo characteristics. Another criticism is the dataset used as it was heavily unbalanced as it output decent but misleading results.

Choi et al. [30] suggested the use of Convolutional Recurrent Neural Networks (CRNNs) to take advantage of Convolutional Neural Networks (CNNs) (see Figure 2.18) for local feature

extraction and recurrent neural networks RNN for temporal information of the extracted features. They state that CRNNs fit the music tagging (multi-label MER problem) task well as RNNs are more flexible in selecting how to summarize the local features than CNNs which are rather static by using a weighted average. A CRNN model was compared to various other CNN based ones (combining 1-D and 2-D kernels with 1-D and 2-D convolution operations). The input was downsampled to 12kHz and transformed into a Mel-Spectrogram which encapsulates the audio data through the Mel scale - where the different pitches can be interpreted by the human ear as equally spaced from one another (see Figure 2.16).

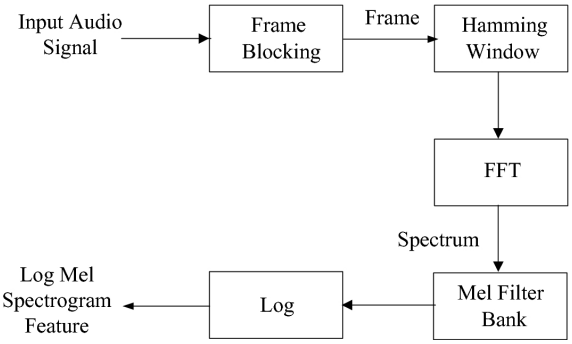


Figure 2.16: Mel-Spectrogram computation (adapted from [42])

This problem is a multi-label classification one, as the output is mapped to 50 tags by a sigmoid activation function. The CNN base models range from 4 to 5 convolution layers followed by 2 fully connected layers that act as a classifier. The dataset used is the Million Song Dataset with *last.fm* tags associated. This is a concern given that this dataset, although massive, has a very poor quality. The CRNN model consists of 2 recurrent layers with gated recurrent units (GRU) after the four layer CNN (k2c2) (see Figure 2.17). The experimental phase is extensive as various levels of parameters are tested ranging from 0.5M values to 3M and noted that although for the most part, more values means more information, therefore a better result, this is not the case as a reduced number of feature maps also removes redundancy among the samples. As expected the CRNN model outperforms the CNN ones with 0.86% AUC (Area Under the Curve) for the general part as it is also much more complex and has a computational cost well above the others. Given the massive output options, this progress, as so many others, tend to go unnoticed as so many other authors choose different emotion models.

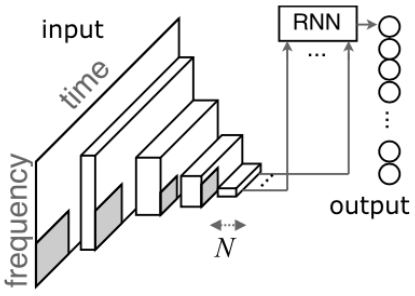


Figure 2.17: CRNN (adapted from [30])

Liu et al. [43] faced the emotion recognition problem as a multi-label task, with 18 possible tags (each with values from 1 to 5). The authors developed a CNN with 4 layers, ranging from 100 to 200 filters each (see Figure 2.18). It is never said which architecture had the best results. The dataset used was the CAL500 which is composed by 500 songs with 18

emotion tags associated, annotated with 1 to 5 values and CAL500exp based on the songs from the previous dataset but segmented to accommodate the possibility of having different emotions, so the dataset grows to a total of 3223 clips. Results as stated by the authors claim a better performance over traditional ML approaches with micro F1-Score of 0.709% and good AUC results. This is not the case for the macro F1-Score (average of F1-Scores for all classes) as it registers a highest value of 0.596%. The disparity between the datasets is also addressed, as the CAL500 does not provide for a good training experience compared to the CAL500exp and has a lower agreement among annotators.

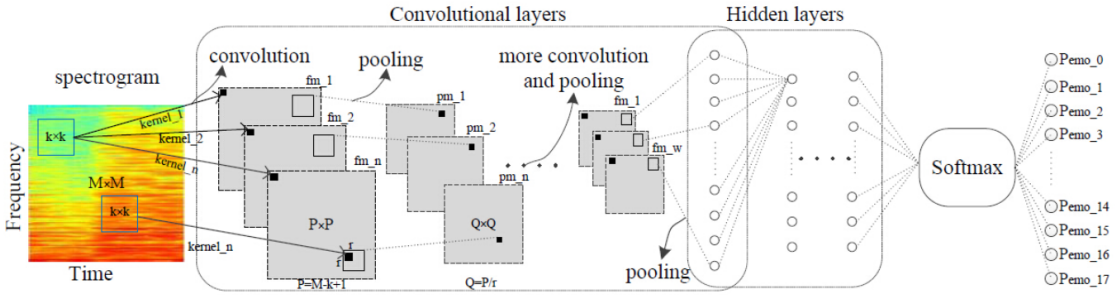


Figure 2.18: CNN (basic architecture) used by Liu et al. (adapted from [43])

Liu et al. [44] propose a CNN model using the 1000 song dataset (referenced in Section 2.2) and as a form of input, utilized a basic spectrogram. The emotion model followed is the Russell AV as a categorical target based on the four different quadrants. The network consists of 8 layers, 3 convolutional layers (with a reLu activation function as most researchers use), each followed by a mean pooling layer. In the latter portion there are two fully connected layers that act as a classifier. The paper has some inconsistencies as for describing the model detailing that it was added a batch normalization layer later and stating that a max pooling algorithm was used however, prior to this information, it is clear that a mean pooling layer was used. The 1000 songs dataset is filtered (eliminating duplicates) resulting in 744 songs that are split into 5 seconds clips. The AV values are mapped to their respective quadrants making this a discrete problem. The convolutional layers use a uniform 2-D kernel (5x5) and a dropout layer is added before the fully connected layers to prevent overfitting. This system is compared to a SVM but it is not referenced the input that this model uses, outputting 0.385% of F1-Score. Not surprisingly, the CNN outperforms the SVM with an average of 0.724%.

In Seo et al. [40] paper, the authors also created a DNN with 3 hidden layers, 30 nodes each trained over 1000 epochs. The model obtained a 72.9% accuracy being the second best model, behind the SVM. A key thing to note is that this does not take full advantage of DL as it receives the features extracted before, it only acts as a classifier.

Cañón et al. [17] build on previous work on the search for a language-sensitive emotion recognition model. The idea was to pre-train models using speech in both English and Mandarin and tune them with music clips from both languages with the hope that features from speech would be transferable to a music environment. Several studies such as the one mentioned in the paper [45] state significant differences of emotional ratings by listeners raised in different mother tongues where online surveys were conducted with participants with English, Spanish, German and Mandarin backgrounds. This only supports what was previously pointed about how subjective and therefore difficult MER is. The targeted emotion model was the Russell AV [16] 4 quadrants and the English library for music was the one used by Panda et al. [1]. The feature extractor for the speech model was a sparse convolutional autoencoder (SCAE), where the encoder section is mainly three

double 2-D convolutional layers (with 3x3 kernels) followed by max-pooling and dropout layers (to prevent overfitting) which is then fed into the decoder section with the same structure but instead of max pooling, up sampling layers (see Figure 2.19). This latter section is then removed and fitted a 3 layer fully connected layer with a total of 512 neurons and two models are made: *SCAE Feat. Ext.* where the weight values are frozen and the the DNN is fine tuned with the music datasets and *SCAE Full* where a the weights are unfrozen and the network is trained with a lower learning rate. Using the SCAE exclusively as a feature extractor shows average performance for all configurations (48% F1-Score). The model and authors state that even though the results for quadrant detection were not spectacular, arousal and valence detection as separate planes was good, which can be explained because the primary confusions tend to be between the first and second quadrant as well as the third and fourth. This paper can be seen as a personalized MER approach, using annotations from a specific type of user, and training a personalized model based on them. It gathers users according to individual factors (e.g., demographics, musical expertise, and personality), and averages the annotated data as common "ground truth". This an interesting and common approach, as a search for a independent solution is extremely hard. One point to make is that the models were not trained based on a 10-fold validation base, as they were only trained 4 times and the dataset which contained Mandarin music had almost double the size as the English one, same for the music.

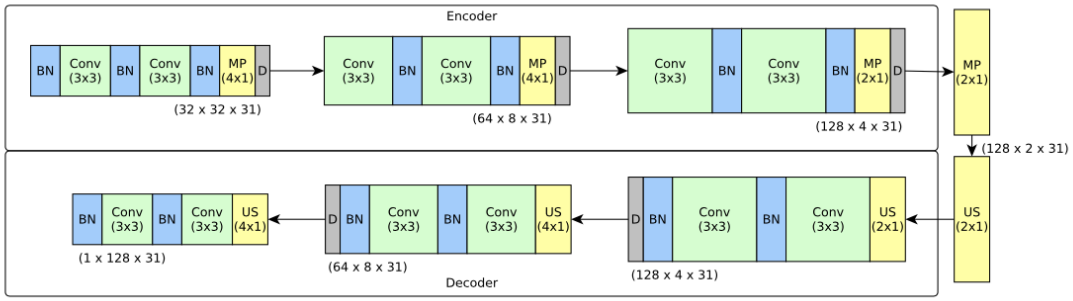


Figure 2.19: SCAE used by Cañón et al. (adapted from [17])

Er et al. [23] presented a recent and strong model as it used pre-trained models such as the award winning AlexNet and VGG-16 (convolutional) neural networks and fed the output to SVM systems tuned to the targeted 4 different categories - "happy", "sad", "angry", "relax" from a 150 music clip dataset Soundtracks. The AlexNet was trained on a 1.2M images and consists of 25 layers which made a huge contribution to the Computer Vision research distinguishing over 1000 objects. The VGG-16 model has 41 layers and same as the AlexNet, focused on distinguishing patterns. These networks were used as feature extractors and the last layers were adapted and fitted to the Soundtracks dataset (see Figure 2.20).

They also introduce a solely Turkish dataset with over 500 clips. Additionally, to increase the size of the input, data augmentation was introduced with two different deformations to each music recording in both datasets. The first was shifting, and the second one was stretching. The sound samples that are obtained at the end of each process are added to the original sound sample class as new data. This poses a problem as augmented data should not be present in the test set as it is not a true representation of real music and makes the model itself biased. The Soundtrack dataset goes from 30 samples to 180 for each class and the Turkish dataset goes from 100 to 600 therefore making them six times bigger. The best classification result before data augmentation is applied, is from the output of the *Fc6* layer of the VGG-16 with a softmax classifier with 76% F1-Score in the

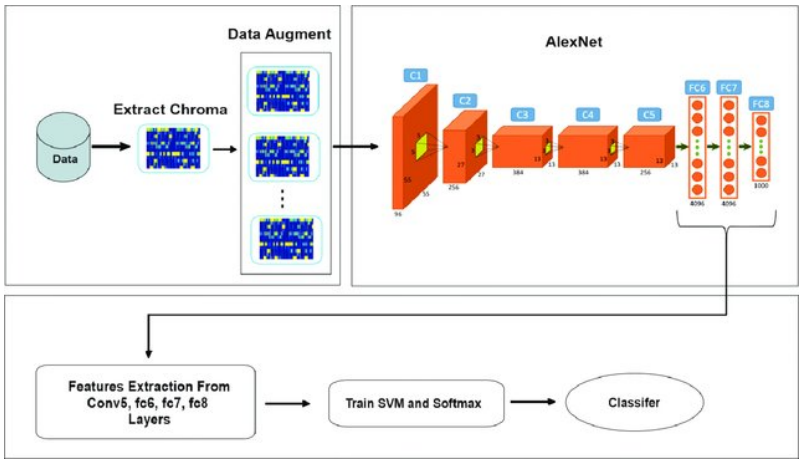


Figure 2.20: AlexNet adaptation (adapted from [23])

Turkish music clips. After the data augmentation, the best classifier success was obtained from the *Fc7* layer of the VGG-16 with the SVM classifier with 89.2% on the Turkish dataset. According to the results, it is safe to say that pre-trained deep learning model can be used for MER problems. Bear in mind that using a considerable amount of augment data in the test input, as stated before, is a concern because it obligates the model to fit around data that is not real and not labeled as original.

Sarkar et al. [24] contribute to this field by exploring the VGGNet (already mentioned above), making it lighter in layers and consequently, less computationally expensive assessing its behavior in the following datasets *Soundtracks* (360 audio-clips collected from background tracks of movies with duration around 30 seconds) and *Bi-Modal* (162 songs where each song clip is of 30 seconds duration and both audio signal and lyrics data is included, being the later one ignored) and *MER_ taffc* (4QAED) [1]. The songs are divided in 5 seconds clips which are transformed in Mel- spectrograms resulting in a 196x128 input size. A 3x3 kernel is used while the stride varies across all 7 convolutional layers, having 5 max pool and 3 dropout layers in between followed by 2 fully connected layers with 2 dropout layers in between them too. The emotion model used is Russell's AV one, mapping the values to their separate quadrants. The model is pitted against a SVM, a Random Forest, a Decision Tree (trained on 66 standard features including time domain, spectral, LPCC (Linear Predictive Cepstral Coefficients) and MFCC as well as their mean and std) and a basic Feed Forward Network. The *Bi-Modal* and *MER_ taffc* presented the overall better results. The final CNN model (which possessed over 1.2M trainable parameters) performed relatively well against the traditional approaches getting a F1-Score of 82.95% for *MER_ taffc* and a 77.82% for the *Soundtrack* dataset. A potential issue not address is that when comparing the results with proposals from Panda [1] it is not fully explained how is this compared given the fact that Panda's approach is a 30 second clip classification and the one in this paper only assesses 5 seconds.

2.5 Music Emotion Variation Detection

This section corresponds to an analysis on the progress and current state of research in MEVD accounting for past/relevant information that pose some interesting and challenging approach worth considering. The following Table 2.5 gathers all information retrieved as the next section explains it in detail.

Table 2.3: Dynamic MER approaches

Author	**Approach**	**Database**	**Features/Input**	**Models**	**Emotion Taxonomy**	**Results**	**Observations**
Schubert et al.	Classical ML	Romantic music (4 songs) (1 second)	Loudness, tempo, melodic contour, texture and spectral centroid	Linear regression	Russell AV (4 categories)	0.33 Accuracy (in changes detected)	Very limited dataset with a unusual result metric
Panda et al.	Classical ML	57 song (25 seconds)	Standard (i.e. intensity, rythm, timbre)	SVM	Russell AV (4 quadrants)	0.44 Accuracy	Very small dataset
Markov et al.	Classical ML	1000 Songs for EAM (0.4 seconds)	Standard (i.e. intensity, rythm, timbre)	GP Regression	Russell AV (4 quadrants)	0.69 R^2 for arousal and 0.44 R^2 for valence	The entire song is not considered, on 45 seconds and the complexity of the GP model can be a problem when applied in a large scale.
Malík et al.	DL	DEAM (0.5 seconds)	Standard features + Mel-Spectrogram	CRNN	Russell AV (continuous values)	0.231 RMSE for arousal and 0.279 RMSE for valence	45 seconds do not contain significant emotional disparsity
Li et al.	DL	DEAM (0.5 seconds)	Standard	Bidirectional LSTM	Russell AV (continuous values)	0.225 RMSE for arousal and 0.285 RMSE for valence	Very complex system as it combines the results of various Bidirectional LSTM models
Hizlisoy et al.	DL	Turkish Emotional Music	Spectrogram + 66 standard features	CNN + LSTM + DNN	Russell AV (3 quadrants)	0.99 F1-Score	No agreement among the annotators is referenced and only 3 quadrants are used
Du et al.	DL	1000 Songs for EAM (0.5 seconds)	Mel-Spectrogram + Cochleogram	CNN + Bidirectional LSTM	Russell AV (continuous values)	0.07 RMSE for arousal and 0.06 RMSE for valence	Data augmentation is used and it is not specified if it is in the test set, being its presence a serious problem as it makes the model biased

2.5.1 Classical Machine Learning Approaches

To the best of our knowledge, the first model to tackle MER as a time varying problem was present by Schubert et al. [46]. It took features like loudness, tempo, melodic contour, texture and spectral centroid and fed it to a linear regression model. Sixty-seven participants rated four Romantic music pieces expressing different emotions in the form of a two-dimensional emotion space - happy/sad for valence and aroused/sleepy for arousal) following the Russell emotion model [16] once per second. Two linear regression models were trained separately and it was evident by the results that changes in loudness and tempo were associated positively with changes in arousal and the melodic contour varied positively with valence, though this finding was not conclusive.

Panda et al. [47] presented a mood tracking platform based on SVM models in order to detect the variation of the song's emotional content where the before mentioned 4 quadrant taxonomy is used. The features from the Marsyas[12] framework used were based on a sliding window structure. The model was trained with a 194 song dataset with arousal an valence values for 25 seconds clips and for the test set, two volunteers listened to 57 full songs (from the original dataset) and registered changes between quadrants for the entire song's duration in 25 seconds clips. The new annotations were analyzed and compared in order to measure the matching ratio between volunteers. Being the remainder only 29 songs, where volunteers agreement was higher than 80%. The best result obtained was an average of 44.09% accuracy. The size of the dataset was the main limitation for this problem and as the author states, a use of a ranked set of features could result in a better performance.

Markov et al. [48] presented an unusual approach do MEVD in the form of a GP model. As aforementioned, GP models are becoming more popular for their superior capabilities to capture highly nonlinear data relationships. The database used was the MediaEval 2013 database (described before as the 1000 Song Dataset) with 1000 45 seconds music clips and a window of size 23.2 milliseconds was used that with the help of the Marsyas framework provided a feature vector of the standard features in traditional MER research such as MFCC, LSP and TMBR. A total of 20 frames were grouped into a window that retained their mean feature value and standard deviation (both as features). The sliding windows overlapped as the window shift was set to 1 frame. A SVM (SVR) model was trained on the same features and the GP model considerably outperformed it, achieving 0.665% and 0.442% R^2 for arousal and valence, with one particular model achieving 0.692% and 0.473%. The valence result is impressive given that valence distinction is a difficult task in MER. An important note stated by the authors is that GP models have a great training complexity which makes them difficult to use in large scale tasks.

2.5.2 Deep Learning Approaches

Malík et al. [49] propose a CRNN model outputting to a 2-dimensional VA emotional space. This model as been stated before [30] as it is getting traction among sound detection and speech recognition. The network consists in a 8 layer CNN (followed by a dropout layer) outputting to basically two branches (one for arousal and one for valence) which are composed of 8 fully connected layers and 8 bi directional recurrent layers. Standard parameters are used in the CNN such as a 3x3 kernel, ReLU activation functions (providing nonlinear relations).The dataset in question is part of the DEAM dataset referenced in Section 2.2. It contains 431 music clips with 45 seconds total (being the first 10 discarded) continuously annotated arousal and valence at a 2Hz rate (every 0.5 seconds) from -1

[12] `http://marsyas.info`,

to 1. Using a 45 second clip as a song can not possibly contain significant emotional disparity but the test dataset consists of 58 full songs which is a nice evaluator. Two input approaches are used: standard features (extracted from the openSMILE[13] toolbox) as well as a Mel-spectrogram and only the spectrogram. The best configuration was the CRNN with both inputs achieving a RMSE (root mean square error) of 0.279 for valence and 0.231 for arousal. The idea to separate the valence and arousal classifiers is a great way to differentiate the dimensions as some features better fit each one.

For the same dataset as the one mentioned above, Li et al. [50] present a Deep Bidirectional Long Short Term Memory (DBLSTM) model. Emotion can be associated with previous and future information, therefore, a bidirectional approach is present. A bidirectional LSTM can be seen as two LSTMs for each training sequence forward and backward,respectively having both information impacting the model. This is then connected to an output layer (see Figure 2.21). The authors also reference that MEVD results are highly dependent on the sequence length and therefore present a model that combines the information from different time scales. The model can be separated into 2 major parts:

- the DBSLTM system itself that provides valence and arousal values, post-processing that contains a better temporal correlation as the AV (arousal and valence) values are transformed from a center point in the segment to a time-continuous value segment with the aid from a triangle filter smoothing, MLR (Multiple Linear Regression) and a SVR;
- the last part, the so called fusion, fuses the results from various DBLSTM models with different time scales where the average, MLR, SVR and even a neural network are used to combine them.

The test set contains 58 full songs as well. The model is trained with standard MER features, 260 to be precise, from the 0.5 seconds segments. The BLSTM was trained for valence and arousal separately and consisted of 5 layers being the first 2 trained based on the features (like an autoencoder) and had their weights frozen. Gaussian noise was also introduced to prevent overfitting. The best model was the combination of fusion followed by post processing parts with a triangle filter and a neural network being used for arousal detection outputting 0.225 RMSE and 0.285 for valence with SVR being used. As said by Malík et al. [49] this is a very complex approach but it presents some interesting ideas and results to back those.

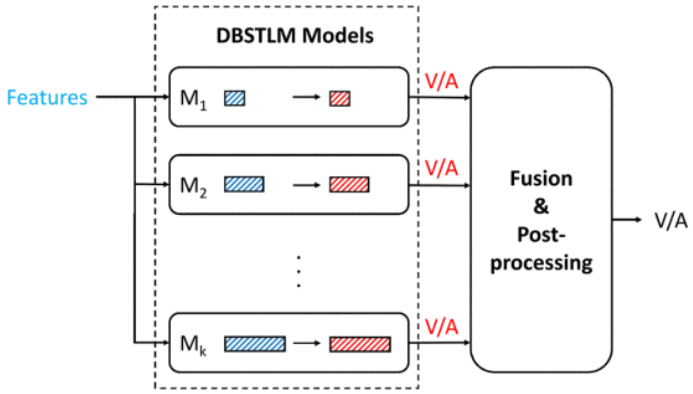


Figure 2.21: DBLSTM model outputting Valence and Arousal (V/A) values (adapted from [50])

[13] https://www.audeering.com/opensmile/,

Hizlisoy et al. [51] propose a solution based on convolutional long short term memory deep neural network (CLDNN) architecture. Adding to this, they created a new Turkish emotional music database (TEM) composed of 124 Turkish traditional music excerpts with a duration of 30 seconds each that is later on divided into 10 seconds segments, resulting in a 372 clip dataset. The music clips were evaluated by 21 university students. The emotional content of each music was rated on a scale from -5 to 5 for valence and arousal. The mean of annotations for each excerpt was calculated by taking the average over all annotators decision for arousal and valence. This raises a question as there is no pre-validation agreement over the annotators. Being only the mean used, some clips can have a major gap in both dimension values which is overlooked. The music clips are distributed into three quadrants making it a 3-class music emotion classification which can be considered a major issue as it removes the third quadrant which is one that alongside the fourth brings out the worst results sharing so many acoustic similarities. The CLDNN model uses the output of CNN as features and combination of LSTM and 2 dense (FC)layers as the classifier. The LSTM layer consists of 200 hidden units. A 1-D CNN layer is used to extract features followed by a flatten layer outputting 1x768 feature vectors. Each FC layer has 100 hidden units and softmax output layer is added to obtain a final category (quadrant). The authors also note that Log-Mel Filterbank Energies and MFCC (Mel-Frequency Cepstral Coefficients) are the most widely used features because they are considered to convey the most relevant information for speech recognition and MER. In addition to this, standard audio features are extracted from the audio signal and fed to the DNN, which is combined with the output of the CNN (see Figure 2.22). These features suffer a selection based on correlation ranging from 66 to 87. The 4 layer CNN is followed by a dropout layer with 0.50% rate. Combining all features brings it to a total of 110 which outputs the best result at 99.19% accuracy after a 10-fold cross validation. Bearing in mind the absence of one quadrant, it is presented as a great result to a not very large but very distinct dataset as by comparison, a SVM model with all selected features has a F1-Score of 97.7%. This indicates a high correlation and fit to the dataset created which suggests a poor performance in other environments.

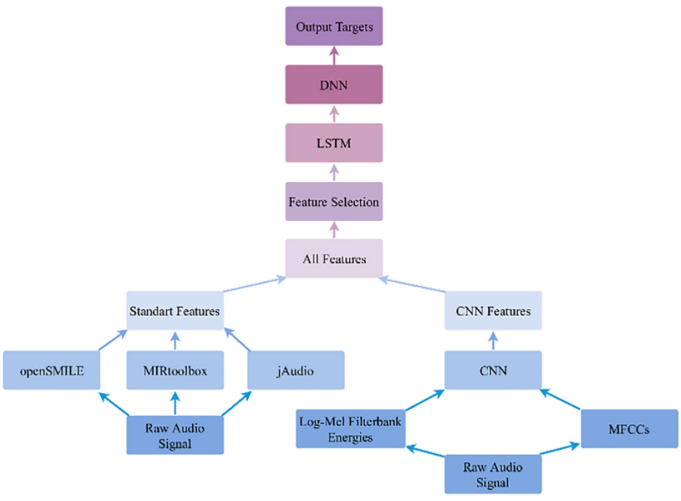


Figure 2.22: CLSTM architecture used by Hizlisoy et al. (adapted from [51])

Du et al. [26] contribute with a CNN-BLSTM model which uses an emotion taxonomy that is heavily adopted, the Russell emotion model with AV continuous values being output. The input is set to a Mel-spectrogram and a Cochleogram (converts a sound waveform into a multidimensional vector with the intention to represent the information sent from the ear to the brain). Basically, the network as a whole consists in two CNNs, each for its different

input, either on having a total of 3 convolutional layers with 2x2 feature maps (6 filters - width each), plus a max pooling and a batch normalization layers, which are then flattened and fed together to a fully connected layer that connects to the BLSTM and the last fully connected layers outputting arousal and valence values respectively (see Figure 2.23). A potential issue with the approach is that, while using a dataset very similar to one used in the aforementioned papers (1000 song dataset7) which possesses 1000 45 seconds clips annotated at a 2 Hz rate with AV values, they perform a data augmentation by changing the frequency and amplitude (not explicit how). These processes, although valid, when used for the test set and effectively doubling the dataset dimension can have a negative effect as the model tends to fit to them and not be a true representation for a real world scenario. The results were great, averaging a RMSE of 0.07 and 0.06 for the arousal and valence dimensions respectively which outperformed the baseline models.

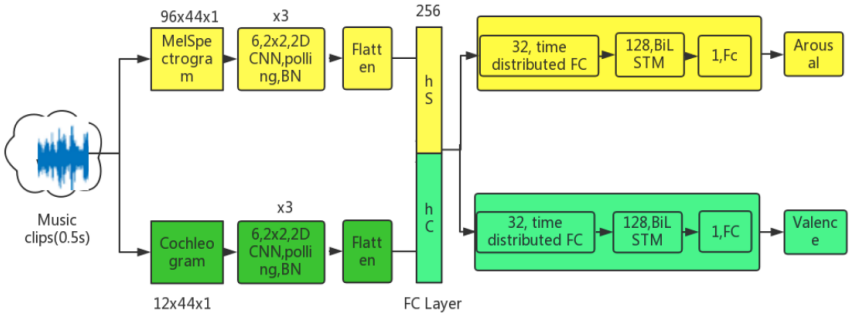


Figure 2.23: CNN-BLSTM (adapted from [26])

One note to take from this complex models is its complexity. Adding various possibilities and outputting (for the most part) better results, it comes with some heavy computational cost which will be taken into account in this project.

2.6 Overview

In conclusion, we can say that the current state of the art heavily favors quantity over quality, therefore it is important to get a grasp on what can be accomplished with a quality dataset (4QAED) with DL approaches compared to what was achieved before with traditional ML. We based our models on previously discussed architectures regarding static MER as well as MEVD. A primarily exploratory strategy was used in order to get a better understanding of the dataset and which approaches translated in the best overall results.

This page is intentionally left blank.

Chapter 3

Static MER

The purpose of this section is to expose and detail the datasets used for static MER as well as the experiments made with them with a central emphasis on the best model for each different approach.

3.1 Data

This section introduces the data used for the static MER models.

3.1.1 Database - 4QAED

This dataset consists of 900 thirty seconds clips with a balanced target, 225 samples for each quadrant (see Figure 3.1) [1]. All these samples went through a heavy selection process, as a matter of fact, the initial number of entries was 370611.

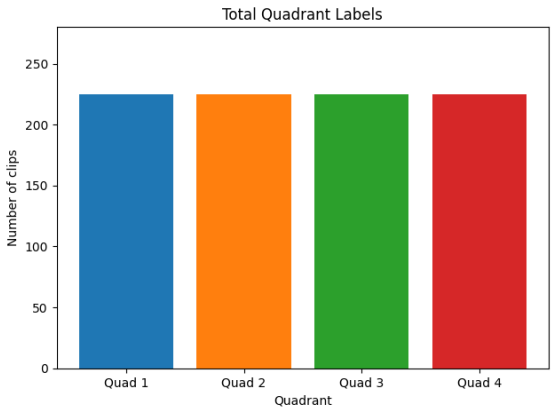


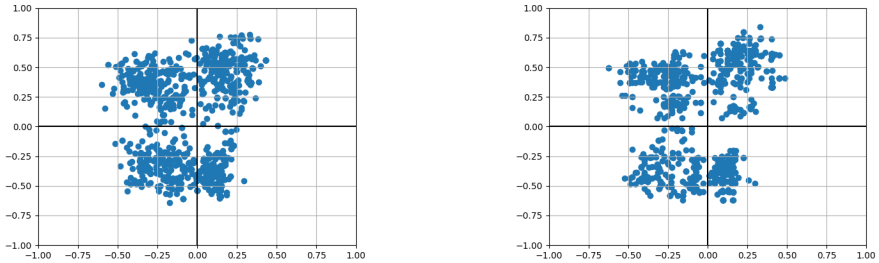
Figure 3.1: Quadrant distribution on 4QAED dataset

The construction of this dataset started by querying the top songs on AllMusic[1] with the 289 emotion tags associated. These emotion tags are not from any known taxonomy, although, according to the AllMusic platform, they are assigned by "professional editors". The next step was to relate the 289 tags to the Russell's AV emotion quadrants based on Warriner's list of English words, which have an assigned value of arousal and valence.

[1] https://www.allmusic.com

This made it possible to appoint a quadrant to each tag. Samples with a non-dominant quadrant (at least 50% of the tags associated being from one quadrant) were discarded as well as duplicates and samples with less than 3 tags. The resulting 2200 samples were manually inspected: noisy and unclear clips were removed and a quadrant was annotated for each sample. The samples in which the given annotation did not match the mapped quadrant from the emotion tags, based on the Warriner's list, were discarded. The dataset was then balanced in terms of the possible 4 targets (see Figure 2.2), resulting in 900 samples. Given that the emotion tags have an associated arousal and valence values, these were saved in order to use this dataset in regression problems as well (see Section 3.2.2).

The average and median arousal and valence values were considered and, for the experiment, the median was chosen as it clearly gives a better separation between quadrants (see Figure 3.2b). The average tends to place the samples closer to the axis (see Figure 3.2a) which does not leave much room for error as the model receives various points from different quadrants but with very little variation between them.



(a) Average Arousal and Valence values for each sample

(b) Median Arousal and Valence values for each sample

Figure 3.2: Arousal and Valence distribution

In order to use the 30-second clips in the following DL approaches, each of them were converted to a melspectrogram (see Figure 2.16). The melspectrogram, as explained before, is a representation used with the intent to present the sound information as the human ear perceives it. We do not perceive the different intervals (with the same gap) of frequencies the same way. For example, our perception of distance between a 500 Hz tone and a 1000 Hz tone is not the same as for a 7000 Hz tone to a 7500 Hz tone, although the gap between both remains the same, 500 Hz. It preserves the most perceptually important information to a human ear. This input form is the most used in the MER and MEVD deep learning approaches as it presents the best results [6]. The data was downsampled to 16kHz, preserving the core information bearing in mind the cost to compute such a dimension. The *wav* files were converted to melspectrograms with 128 filter banks, a hop size of 512, which resulted in a 942x128x1 sample, only having one channel. This are the most used values when converting to a mel-spectrogram as they give a wide range of frequency intervals and keep a reduced dimension [24]. However, other input parameters should be tested in the future.

3.1.2 Features

This dataset is based on the audio clips in 4QAED, where, in total, 1714 features were extracted for each audio clip [1]. The standard features were extracted with the *MIR Toolbox*, *Marsyas* and *PsySound*, which resulted in 1603 values per sample. The nature of these features varied from melodic, to dynamic, to rhythmic, etc. Given the strong

connection between all frameworks, a feature reduction was performed by removing the ones with a correlation higher than 0.9 and the *RelieF* [52] feature selection algorithm was used to further reduce the number of standard features. This brought down the number to 898. Besides these standard baseline features, novel features were also added to the feature set, as described in [1] (this work was carried out by Prof. Renato Panda in his PhD research work). In total, the dataset with 1714 features has 898 decorrelated standard features and 816 novel ones. The novel features significantly improved the classifier by reaching a **76.4%** F1-Score (previously 67.5%), which was reproduced in Section 3.2.1 with only the top 100 features, with 71 standard and 29 novel features (see Appendix A).

3.2 Methods and Results

This purpose of this section is to explain the different static MER approaches and their results. It is important to note that every model was evaluated with a **10-fold cross validation repeated 10 times**.

3.2.1 Classic Machine Learning

To replicate the previously referenced result [1], we used an SVM with a polynomial kernel, a cost of 8 and a gamma value of 0.001953125 with a 10-fold cross validation repeated 10 times. The data used was the **top 100 features from the 4QAED dataset** [53] and the best result achieved was an average macro F1-Score of **76.0%**. The main purpose was to solely achieve the previous result in order to compare with other DL approaches.

3.2.2 Deep Learning

From the start, it became clear that the evolution of the architecture was basically set on one objective: achieving the best result while preventing the model from overfitting (when it models the training data too well and can not generalize). Given the size of the dataset, the overfit problem was expected as the model is not capable of generalizing patterns in the training set. The fewer the training samples, the fewer the possible patterns the model can learn from and, therefore, the lower the performance on samples it has never seen before.

One way to prevent this is using a simpler model with fewer trainable , as it brings unneeded complexity. To achieve this, instead of the initially thought 3 to 4 dense layers (see Figure 3.3), we used 2 or even 1, bringing a lower training accuracy but a better performance in the test set. Another way to prevent overfitting is to use dropout layers as previously discussed. Their purpose is to randomly ignore the information outputted from the neurons. A dropout layer set for example to 0.4 transforms the output of 40% of the neurons to 0 making it irrelevant on the next layer. This layer was initially set just before the classifier section of the network but its use after each convolutional layer became clearly advantageous as it prevented the model from reaching a higher value of accuracy early on the training set.

Many other possible alternatives were also tested, for example, label smoothing. Basically it consists in randomly modifying the target value of the sample, going from [0, 1, 0, 0] to [0, 0.8, 0, 0], which directly influences the loss calculated when training, adding some noise, which can prevent the model from becoming too 'overconfident' with its predictions. Unfortunately, neither of these improved the accuracy of our models.

For the most part, two major architectures were used after some extensive testing (see Figure 3.3) with two distinct kernel sizes: 3x3 and 5x5, the latter one outperforming the smaller one in the case of larger samples. We started by having 3 convolutional layers but quickly gravitated towards 4 as it heavily reduced the size of the output and provided better results. The same can be said for the number of filters used for each layer. It is considered a standard to increase the number of filters per layer as we go deeper in the network, not only in MER [6] but also in object recognition [54]. As the data goes through the model, layer by layer, the patterns, the features it recognizes, become more and more complex so it makes sense that this is a basic rule of thumb. Although, when dealing with a small dataset, this quickly becomes detrimental to the model as it overfits with the unnecessary complexity.

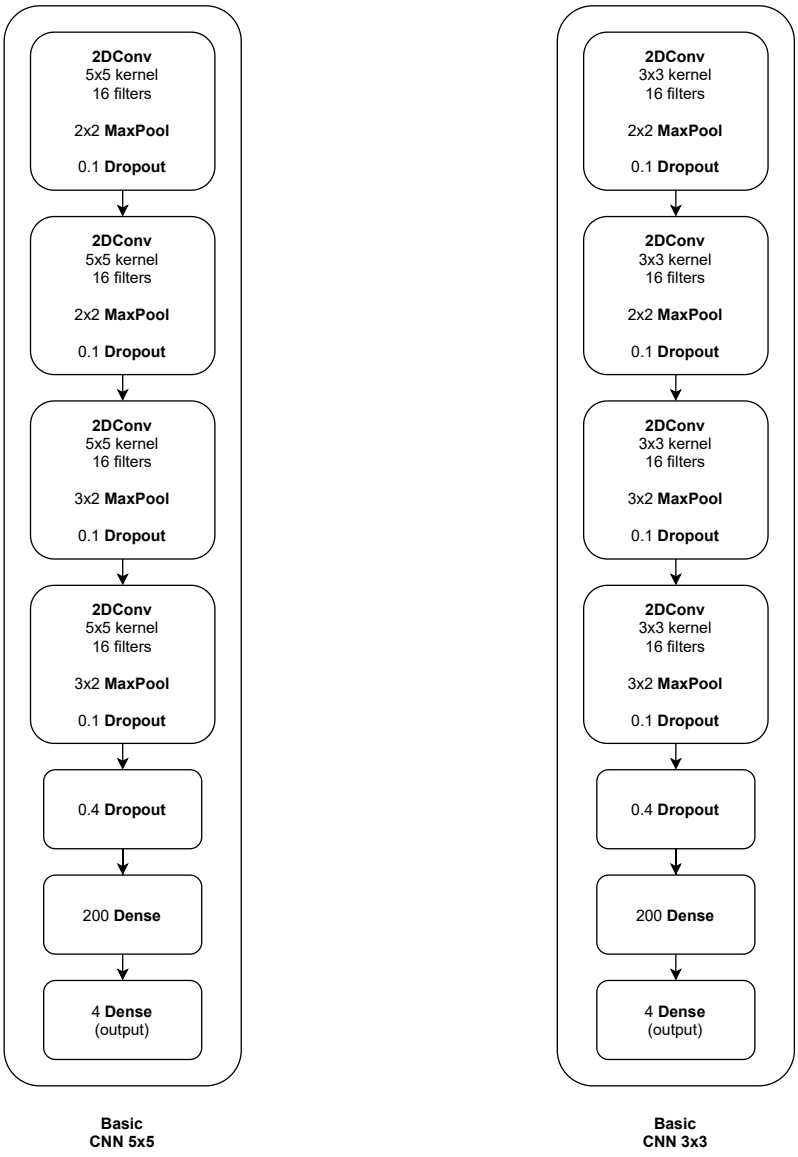


Figure 3.3: Basic CNN Architectures

Regarding the data, there are some ways to restrain from overfitting such as data augmentation. Two main processes were taken: classical audio data augmentation (i.e. pitch shifting, time shifting, time stretching and power shifting) (see Section 3.2.2) and generating new artificial data with a GAN (see Section 3.2.2).

One key aspect heavily experimented and far from standard was the used optimizer, as

well as its associated learning rate. When working with a small amount of data, the learning rate had to be minimized in order to not have the model overfit after just 10 or 20 epochs. The opposite can be said for a large amount of data, as a relatively low learning rate actually made the model underfit. The batch size (number of samples seen by the model before updating its weights) gravitated from the lowest possible to the maximum with a 50 step size, being the few hundreds (150/200/300) the most effective sizes. The number of epochs strongly correlated with the optimizer and its associated learning rate. We tested from 10 to 200 epochs with a 10 step size, also depending on the dataset used. The learning rate for SGD was mainly 0.1, as it translated into the more safe and less prone to overfitting models, and Adam with values ranging from 0.0005 to 0.01 with a 0.0005 step size. Although being considered one of the last major step forwards in DL, it did not perform as well, making the model very prone to overfit. All values are presented in Table 3.1. It is important to note that these values are a template of what was experimented with. As explained before, the parameters heavily depend on each other and especially on the model and the data used.

Table 3.1: Hyperparameters intervals

Parameter	**Range**
Adam - learning rate	[0.0005:0.0005:0.01]
SGD - learning rate	[0.005, 0.01]
Epochs	[10:10:200]
Batch size	[1:50:900]

4QAED - Original Melspectrogram

Firstly, we experimented with the dataset without any alterations, with a simple CNN which we worked on and improved to have the architecture Basic CNN 5x5, displayed in Figure 3.3. The 5 by 5 kernel performed better than the 3 by 3 which is expected as a bigger input tends to benefit from a bigger kernel. The highest F1-Score was **63.56%** which is not, by any means, impressive. We can quickly see a discrepancy in the accuracy among quadrants, primarily between the second quadrant and the third (and forth) (see Table 3.2).

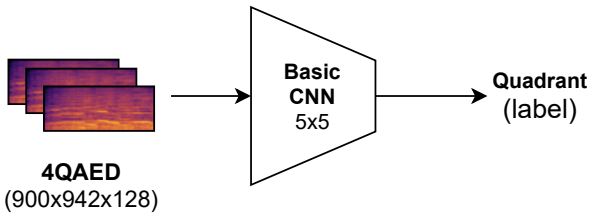


Figure 3.4: CNN on 4QAED

Table 3.2: Confusion matrix and F1-Score per quadrant for 4QAED 30-second dataset (predicted labels vertically and annotated labels horizontally)

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score**
Q1	**1605**	381	148	116	64.31%
Q2	421	**1734**	79	16	76.59%
Q3	393	83	**1233**	541	55.43%
Q4	314	62	669	**1205**	57.85%

Note that when discussing a neural network, in this case a CNN where a sample of input corresponds to a label, the assigned input size is fixed as opposed to an RNN where we can have different approaches (see Section 4.2).

A similar approach was made with the intention to experiment with a larger dataset, double the number of samples. Assuming the emotion remains the same across the 30-second clip, and that in a 15-second clip we can accurately detect that emotion, the dataset samples were split into 2, 15 seconds each. For this, the network architecture remained the same but with a smaller kernel given the smaller sample, 3x3. The larger number of samples made an increase in filters possible, instead of a fixed 16 filters for each convolutional layer, the latter 2 doubled by having 32 each.

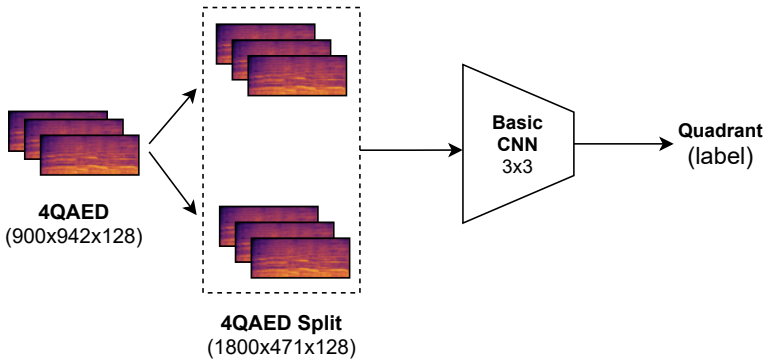


Figure 3.5: CNN on 4QAED Split (half)

This resulted in a higher F1-Score of **66.59%**, which is statistically significant with a p-value of 2.02×10^{-7}, outperforming the previous model.

Table 3.3: Confusion matrix and F1-Score per quadrant for 4QAED 15-second dataset

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score**
Q1	**3170**	655	193	482	67.54%
Q2	586	**3703**	99	112	81.64%
Q3	555	140	**2312**	1493	56.39%
Q4	555	70	978	**2897**	60.79%

We also experimented with regression by focusing on the arousal and valence values separately in the hope to achieve a better valence accuracy, given that it is the weaker part of all MER approaches. As described in Section 3.1.1, we used this same dataset, but, instead of being mapped to the respective quadrant, we took the median arousal and valence values as targets for our model (see Figure 3.6).

Each branch is an independent network. Later on, we mapped the predictions for the arousal and valence to their respective quadrants and achieved an F1-Score of **50.36%** which is beneath the two previous models. The RMSE (Root Mean Squared Error) with respect to the arousal was **0.189** and the valence, as expected, was far greater at **0.424**.

Audio Augmented Data

As explained before, the scarcity of samples poses as our biggest issue. Audio augmentation is a way to upsample the training set while maintaining only original samples for the test set, therefore respecting a real-world scenario where the model has never seen the samples or similar ones. Four different ways of augmentation were experimented with: time shifting,

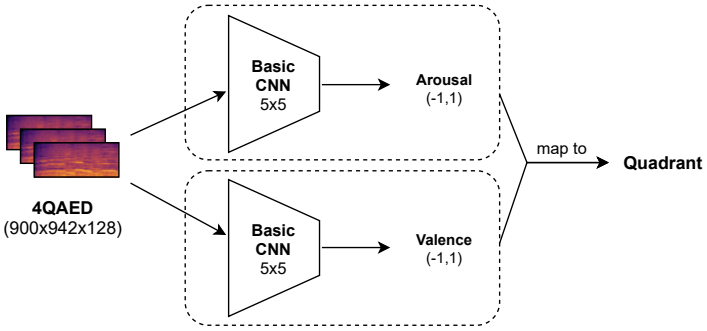


Figure 3.6: Double branch CNN for Arousal and Valence

pitch shifting, time stretching and power shifting. The following figure represents a normal, original melspectrogram (see Figure 3.7).

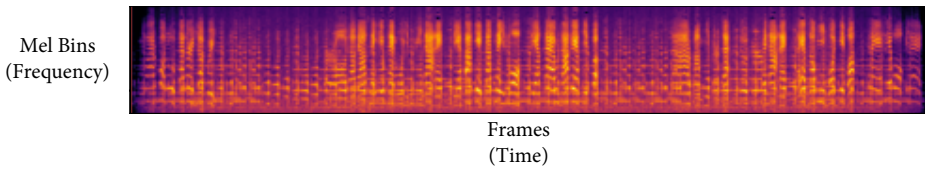


Figure 3.7: Original sample from 4QAED

Time shifting: the audio sample shifted to left or to the right, backwards or forwards, at random, from 0.5 to a maximum of 3 seconds, for each sample (see Figure 3.8).

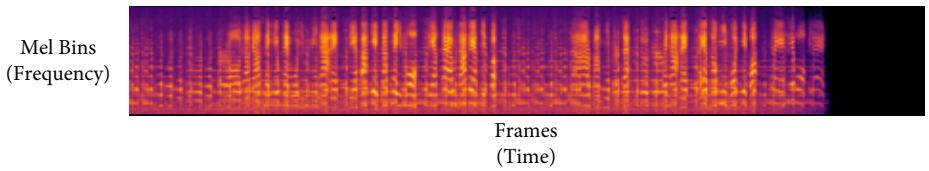


Figure 3.8: Sample after time shift

Pitch shifting: randomly lowered or raised the pitch by a tone which according to Aguiar et al. [55] outperformed a half of a tone shift (see Figure 3.9).

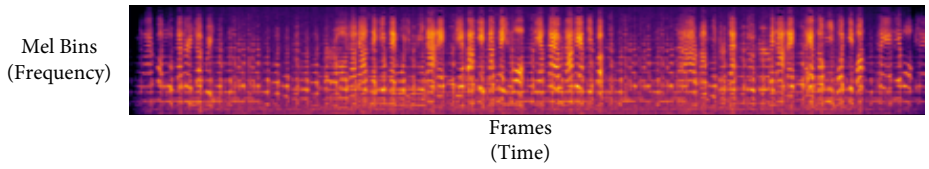


Figure 3.9: Sample after pitch shift

Time stretching: randomly increases or decreases the speed of the song by a factor of 0.5 (see Figure 3.10).

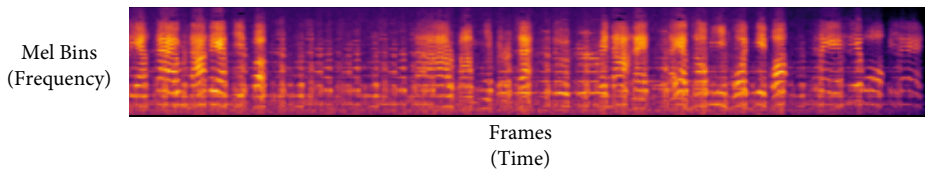


Figure 3.10: Sample after time stretch

Power shifting: as the name implies, we randomly added or subtracted 10 dB over the whole sample (see Figure 3.11).

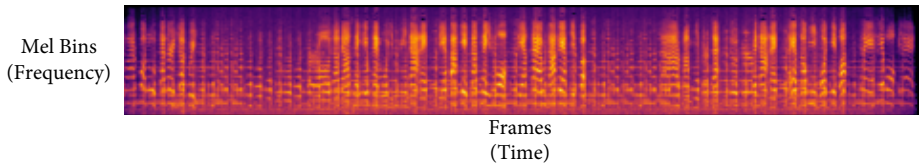


Figure 3.11: Sample after power shift

All of these methods serve a purpose: adding some robustness and generalization capability to our model.

We ended up with over 4 times the original number of samples for training, 4410. We experimented with the whole dataset, as well as individually, with each augmentation method.

Firstly, with the maximum number of samples, the basic 5x5 model used before began to underfit given the larger amount of samples. To compensate for this, another two layers were added to the classifier section of the network (see Figure 3.12).

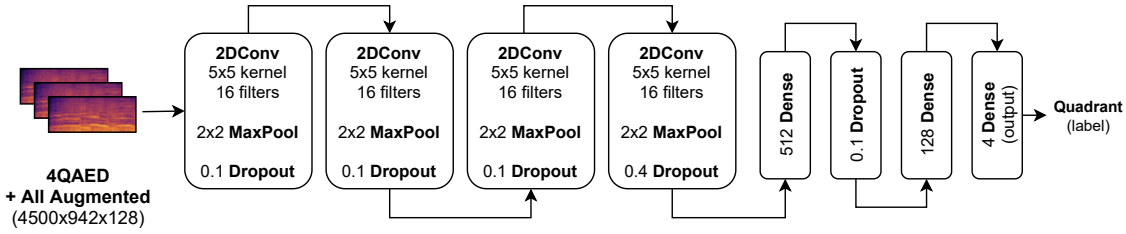


Figure 3.12: CNN for all augmented data

To add to this, the Adam optimizer proved to be a better fit which meant that a lower learning rate was used as it tends to aggressively update the weights from the start.

The following approach was the same for the augmented data from each method (see Figure 3.13).

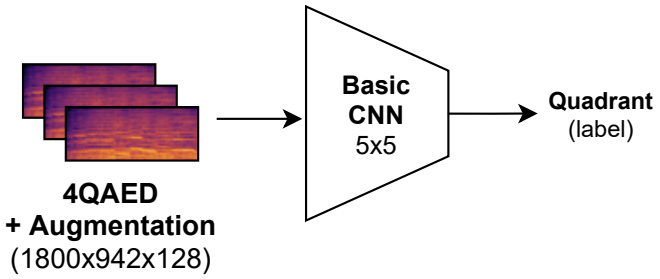


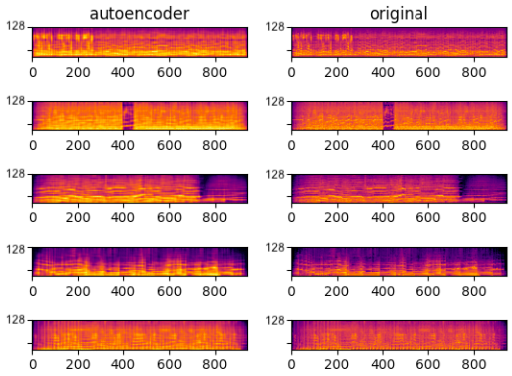
Figure 3.13: CNN for each type of augmented data

None of the models trained with the individual sets of data performed better than the one with the whole data. The model trained on the augmented data had an F1-Score of **66.36%**, which is a statistically significant result, with a p-value of 9.59×10^{-8} compared to the best model trained on the 900 original (30-second) samples, but not significantly different from the model trained on the dataset split into 2, with a p-value of $0.5447 > 0.5$.

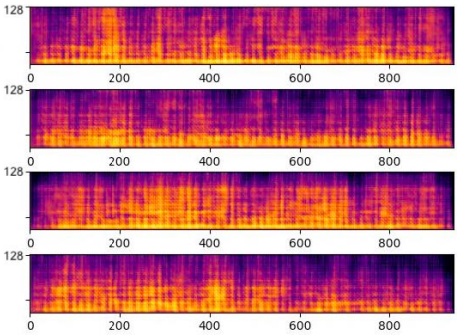
GAN Generated Data

Just as the audio augmentation previously covered, a higher number of samples can be beneficial. With that in mind, we aimed to develop a GAN that was capable of generating replicas of melspectrograms for each class. We began to create a cGAN (Conditional GAN), which takes into account the class of samples, in this case, the quadrant, and is able to adapt to it. As explained in Section 2.3, an autoencoder was trained on the whole dataset (see Figure 3.14a) to use as starting point, a baseline for the GAN train, but to also create four different distributions in order to generate a random input for the GAN model based on the quadrant (see Figure 2.12).

We realized that there were too few samples in order for the model to truly adapt to the data. Although the output data was not far from the original (see Figure 3.14), it did not make a positive impact on the overall performance of the network. We tested with 25 to 100 additional samples from the GAN, for each quadrant, the best being the 25 additions per quadrant with an F1-Score of **60.44%**, underperforming compared to the first model trained on the 900 samples.



(a) Autoencoder results comparison



(b) GAN output example

Figure 3.14: Autoencoder and GAN outputs

Voice Separated with *Spleeter*

With the aim in mind to improve the valence accuracy and overall performance, we used the **Spleeter**[2] tool to separate the voice from the instruments for each sample. It was developed and released by *Deezer* for this same purpose, to be used with research in mind.

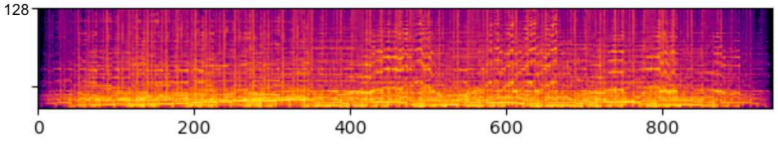


Figure 3.15: Original sample

It is capable of separating the sample up to 5 different channels: vocals, drums, bass, piano and other instruments. We decided to only separate the voice from the rest in order

[2]https://github.com/deezer/spleeter

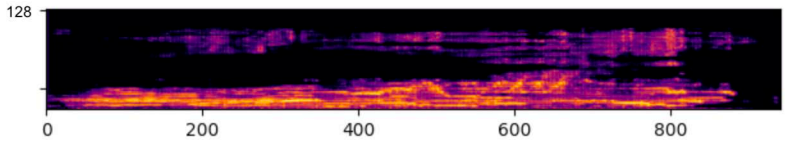


Figure 3.16: Voice only sample

to evaluate its performance as an addition to the original dataset (see Figure 3.16). The vocals tend to significantly influence emotions from quadrant 3 and 4 [1], as variations in tremolos, vibratos and other techniques can be associated with sad and depressed songs. In order to do it, the voice input has its own independent branch (working as a feature extraction solely for the vocals) and a second branch for the original, untampered, dataset (see Figure 3.17).

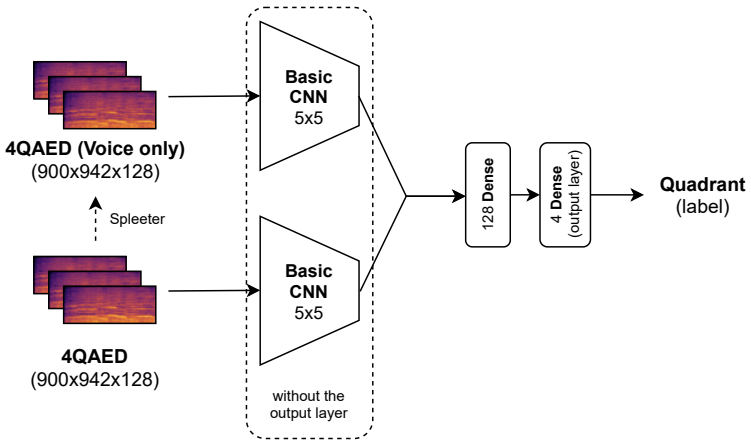


Figure 3.17: Double branch CNN for voice and original 4QAED inputs

The results proved that the attempt to improve the overall performance or the valence differentiation was not successful, achieving an F1-Score of **63.38%** not being a statistically significant result compared to the original dataset model results with a p-value of $0.1167 > 0.05$.

Transfer Learning

Efforts were also made in order to study the performance of models trained on different datasets for different purposes. One major constraint was the fact that not all authors make their models public and others tend to not display the entire information regarding the training phase of their network (i.e. number of epochs, batch size). We found a music genre classification model[3] that performed well, attaining an F1-Score of 83.2%, on a well-known public dataset, *gtzan*[4] with 1000 samples and 10 target genres: blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae and rock. It is important to note that there is some correlation between genres and emotions [56] (e.g. hip-hop and heavy-metal with Q2, reggae with Q4), which makes transfer learning experiments with this types pf models worthwhile. Nevertheless, the *gtzan* dataset has just 100 more samples than the 4QAED dataset, which do not gave us much hope for a great result, not having a great

[3]https://github.com/Hguimaraes/gtzan.keras/tree/10ec9ac896c181a5703b70de8987f3689544a350
[4]https://www.tensorflow.org/datasets/catalog/gtzan

amount of variation.

Each sample represents a 30-second clip which is split into 1.5 seconds windows with a 50% overlap. The labeling process is done by majority voting, where the most voted genre is the chosen one. Therefore, the same process was applied to our dataset, the 30-second were divided and the label, in this case, the quadrant associated with each sample was decided by majority voting.

Similar to all transfer learning approaches, the model is loaded with the pre-trained weights and the feature extraction section is frozen (unable to further train). Three layers were added in order to act as a classifier (see Figure 3.18).

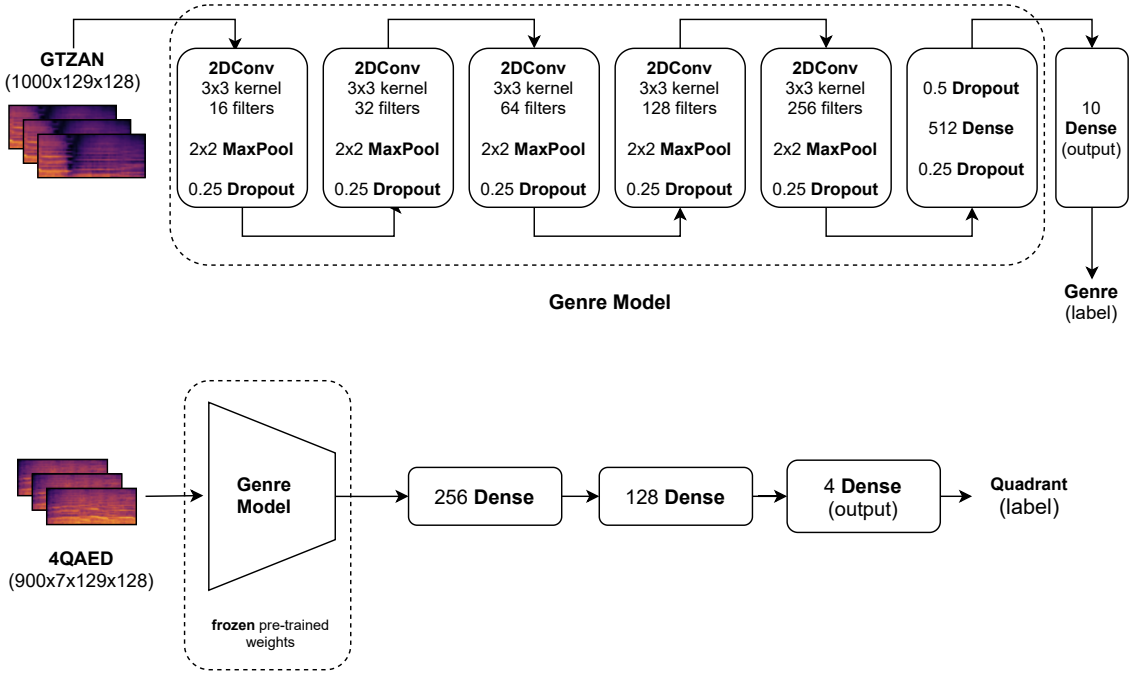


Figure 3.18: Genre trained model for transfer learning

The results were very low, with the best F1-Score (related to the total sample) being **19.1%**. The F1-Score regarding the 1.5 clips was **36.26%** but it is still far worse than the previous seen models.

Table 3.4: Confusion matrix and F1-Score per quadrant for Genre Transfer learning model

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score**
Q1	**137**	560	438	1115	9.53%
Q2	64	**539**	463	1184	24.45%
Q3	118	458	**501**	1173	14.92%
Q4	100	529	498	**1123**	27.48%

As trained models for genre and speech emotion recognition are not available, we decided to experiment with other pre-trained models. One of the most known and used neural network is the VGG19 [54]. It was trained to detect over a 1000 objects, from different dog breeds, to musical instruments, to common daily objects like a desk or a can opener. This is a very complex problem, as it deals with a thousand objects that are, for the most part, unrelated. This network is very deep, with 19 layers (see Figure 3.19). The training dataset had 1.3 million images and even with several high end GPUs, it took up to three weeks to train a single model. It is important to point out that given the complex nature

of this problem, this network is ready to detect various patterns and the many convoluted ways they correlate, in order to differentiate from a thousand unrelated objects.

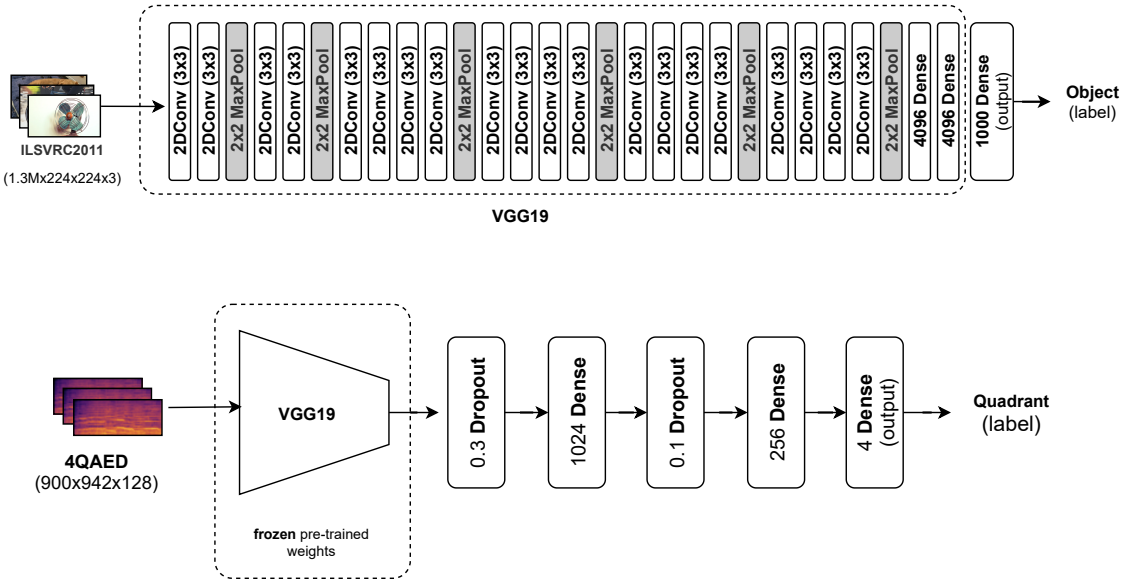


Figure 3.19: VGG19 model for transfer learning

We removed the last dense layer and were able to use two dense layers that acted as classifiers. The previous convolutional layers were frozen. Surprisingly, after several iterations for the classifier section we reached an F1-Score of **67.86%** (see Table 3.5). This is a statistically significant result compared to the split dataset model result, with a p-value of 0.0063.

Table 3.5: Confusion matrix and F1-Score per quadrant for VGG19 Transfer learning model

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score**
Q1	**1562**	292	196	200	69.44%
Q2	235	**1835**	96	84	80.95%
Q3	182	100	**1410**	558	60.87%
Q4	222	51	616	**1361**	60.20%

Features

As discussed in Section 3.1.2 and using the data used for the SVM (see Section 3.2.1), we designed a DNN to explore a DL approach to the features. This was also an opportunity to experiment with a recent addition to the dataset: 11 new features retrieved from the Spotify[5] platform [53]. We trained three different models: one for the top 100 features [1], one for the top 100 features plus the top 11 Spotify features and one for the all set of 1714 features retrieved in [1] (see Figure 3.20).

Given the small number of samples, the network was very difficult to manage even using the SGD optimizer and a lower learning rate with the Adam optimizer. To put into perspective, the initial network had 6 layers with 150 neurons each and by the tenth epoch, it overfitted. Being a DNN and receiving only 100 or even 1714 values, as opposed to the 115200 values

[5]https://developer.spotify.com/discover/

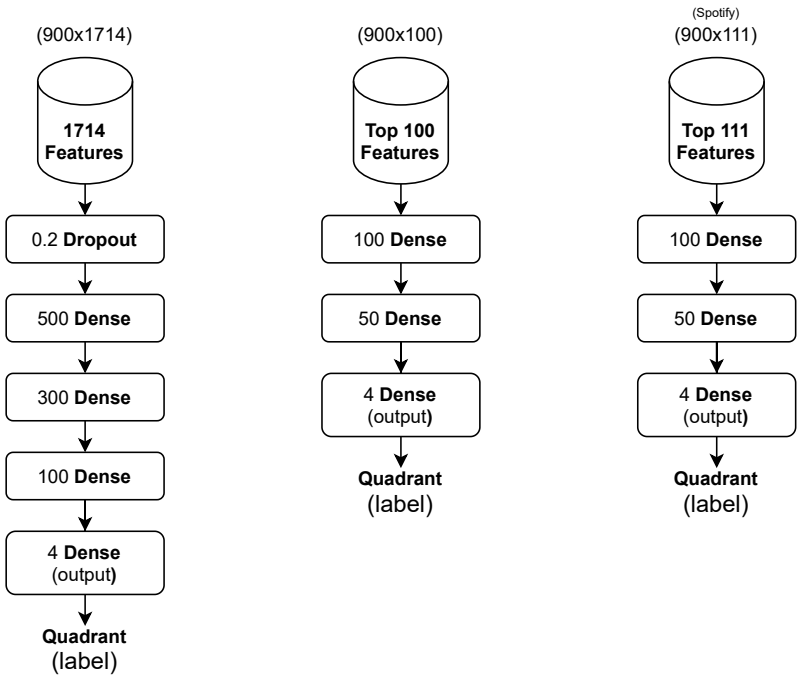


Figure 3.20: DNN models for features dataset

that the initial CNN receives, the overall training time is heavily reduced accompanied by the fact that the Adam optimizer is much faster (and much more aggressive) at training.

The model with the top 100 features reached an F1-Score of **72.88%** which is a lower result being statistically significant compared to the **76.0%** achieved with the SVM, with a p-value of 1.04×10^{-8}. The model with the top 111 features outperformed the one with the top 100 reaching an F1-Score of **73.83%**. The model that received all the features available reached an F1-Score of **82.34%**, being a statistically significant improvement over the previously best performing model with this dataset with a p-value of 6.01×10^{-26} (see Table 3.6).

Table 3.6: Confusion matrix and F1-Score per quadrant for the DNN with all 1714 features

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score**
Q1	**1910**	143	93	104	83.79%
Q2	110	**2030**	75	35	89.35%
Q3	68	87	**1847**	237	79.93%
Q4	221	35	365	**1629**	76.32%

The latter model suffered the most changes, as it constantly overfitted, which is expected as the higher dimensionality and much higher number of learnable parameters (over one million as opposed to fifteen thousand) are the best ingredients for the model to adapt too well to the training data. To handle this, a dropout layer was added, the number of epochs was reduced to 10 and we limited the maximum training accuracy the model could reach to 90% before stopping the training phase.

Features + 4QAED

We combined both the CNN and DNN models with the aim to improve the performance of the DNN model with all the features (see Figure 3.21). For this, we trained the DNN model

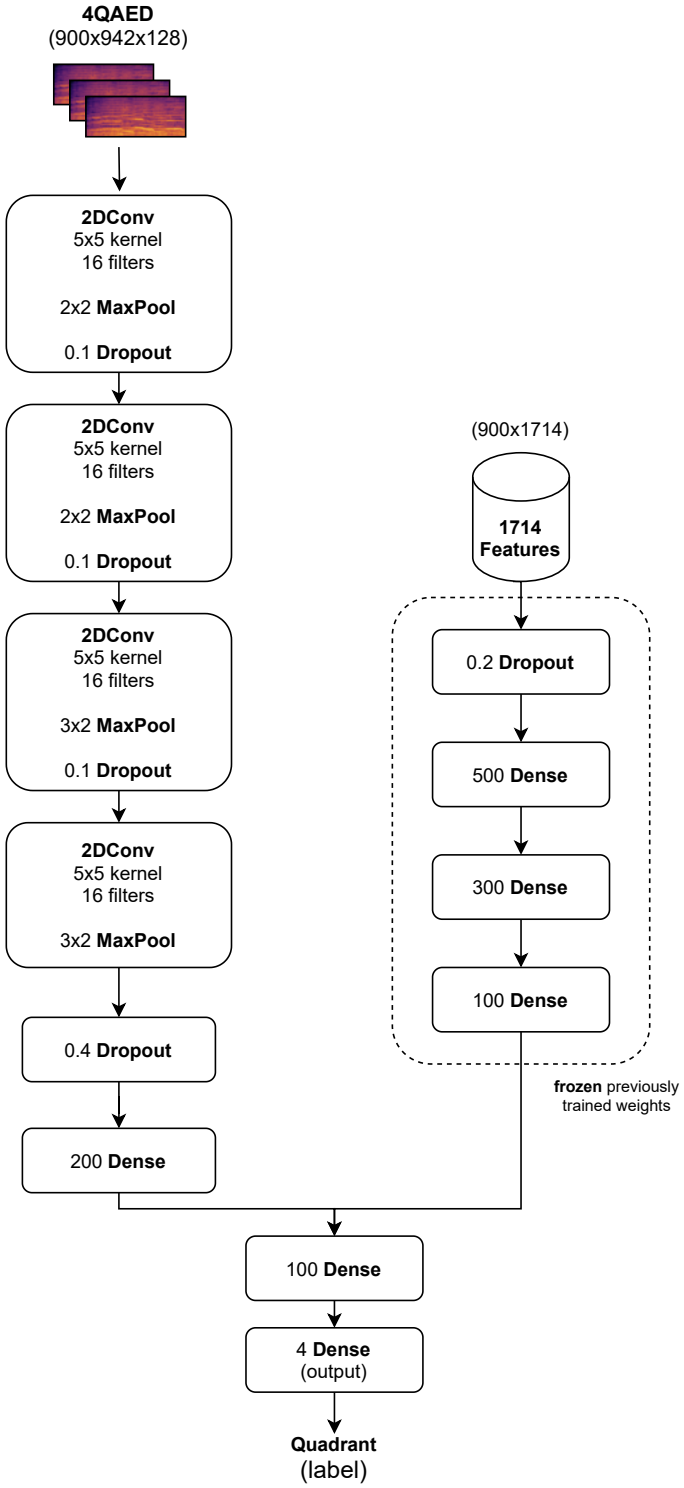


Figure 3.21: Hybrid model with pre-trained features model and 4QAED

and froze its weights for every fold. We then loaded the model and proceeded to only train the CNN with the original 900 melspectrograms input. A hidden layer and the output layer were added to combine the information from the two branches. The model reached an F1-Score of **88.45%**, proving to be a statistically significant improvement compared to the DNN trained solely on the features, with a p-value of 3.6367×10^{-17}.

The difference is especially significant regarding the fourth and third quadrants when

compared to previous models, which still remain the hardest to differentiate (see Table 3.7).

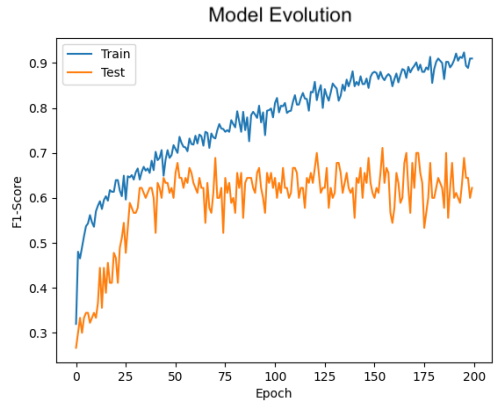
Table 3.7: Confusion matrix and F1-Score per quadrant for the CNN-DNN hybrid model

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score**
Q1	**2067**	33	44	106	89.13%
Q2	128	**2026**	61	35	92.11%
Q3	64	59	**1882**	235	86.58%
Q4	120	21	119	**1990**	85.98%

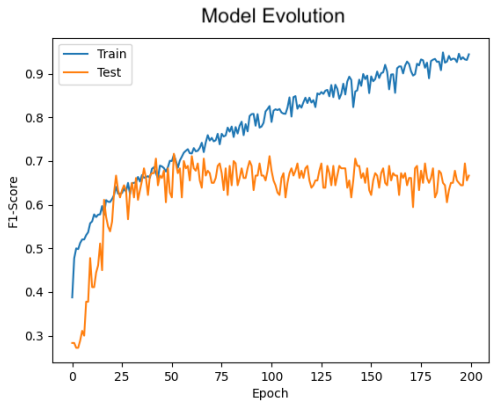
3.3 Results Analysis

All the best model's hyperparameters, results and time of computation are present in Table 3.8.

Overall, it is easy to say that the reduced number of samples is the main problem for the CNN approach. Splitting the data into 15-second clips improved the model's ability to generalize (see Figure 3.22). We can see that the split dataset model takes more iterations to reach the same training accuracy as the model trained on the original, untampered dataset.



(a) CNN (trained on the original 30-second dataset - 900 samples) F1-Score evolution per epoch



(b) CNN (trained on the split 15-second dataset - 1800 samples) F1-Score evolution per epoch

Figure 3.22: Model evolution over 200 epochs

As seen in the graphs, it is very difficult for the model to generalize and have a a good performance on the test set. This is true for almost every model and can be associated, once again, with the small number of samples.

It is important to mention that although giving the model more data is important, it is not the answer. Several experiments were made with only 1, 2, 5 and 10-second clips and it did not translate into better results. The same can be said, for example, for the audio augmentation dataset where the individually augmented data did not improve the overall result, only all the data combined brought some robustness to the model.

Related to the transfer learning model, trained on the genre dataset, a possible explanation can be found for the poor performance. According to study by Griffiths et al. [56], the perceived emotions of genres such as blues, classical, reggae and jazz can be associated with a sad and relaxed emotion which is associated by the authors as a quadrant 3 and quadrant 4 emotion, respectively. This can be one of the reasons for the model results as the model seemed to deviate towards the fourth quadrant (see Table 3.4). The accuracy for the third and fourth quadrants are the worst, for all models and this is a recurring theme in the MER and MEVD world [6], as it easier to perceive the difference between positive and negative arousal than positive and negative valence, even for us, annotators.

Regarding the DNN models, and applicable to all models, the initial thought to create a deep network to take full advantage of the server capabilities did not work as it became clear that more parameters generally meant less accuracy in the test set. A smaller dimensionality usually means that the data is less sparse and more statistically significant for the model to learn from. This, surprisingly, proved not to be the case for the DNN trained

on all 1714 features but it is crucial to keep in mind that the training phase was heavily limited and extra efforts were made for the model not to overfit as explained previously.

The computational time it took for each network to train was recorded but it became logical that the overall combinations of different hyperparameters, number of trainable parameters as well as different datasets was too complex to be able to draw any specific conclusion regarding the elapsed time. Adding to this, the server, as previously discussed, was not only dedicated to these experiments. In other words, the resources were not a stable variable. However, it can be said that besides the experiment with the dataset split into 2s clips with a 12.5% overlap, the maximum elapsed time was 996.7 minutes which is explained by the 1800 samples, which is not a huge amount of time given the number of samples.

A key aspect to note is that this computational time refers to the training and testing process only. In other words, when using the model to classify any sample, the model processes it in seconds, being the transformation from audio to melspectrogram the most time consuming computation. This represents a huge advantage when compared to a traditional ML approach, where the feature extraction represents a much more time consuming step.

Table 3.8: Best results for DL static MER approaches

Model	Input	Filters	# of parameters	Optimizer (lrate)	Batch	Epochs	F1-Score (mean)	F1-Score (std)	Time (min)
Original Data CNN	4QAED - Melspectrogram (900 samples - 942x128)	16 16 16 16	624,360	SGD (0.01)	150	200	0.6356	0.0462	524.4
Split Data CNN	4QAED - Melspectrogram Half Dataset (15s) (1800 samples - 471x128)	16 16 32 32	1,271,900	SGD (0.01)	150	200	0.6659	0.0337	444.6
Top 111 Features DNN	4QAED Top 100 + 11 Spotify features (704 samples)		18,236	Adam (0.0005)	300	80	0.7383	0.0513	10.67
Top 100 Features DNN	4QAED Top 100 features (900 samples)		15,354	SGD (0.01)	150	300	0.7288	0.0422	22.25
All 1714 Features DNN	4QAED 1714 features (900 samples)		1,045,160	Adam (0.0005)	450	10	0.8234	0.0397	1.05
Original Data + All 1714 Features Hybrid CNN + DNN	4QAED 1714 features + MelSpectrogram (900 samples)	16 16 16 16	721,596	SGD (0.01)	300	100	0.8845	0.0357	271.33
Split Data (1.5s) Pre trained CNN on Genre Dataset	Pretrained with GTZAN 4QAED - Melspectrogram Split (1.5s clips) (6300 samples - 129x128)	16 32 64 128 256	395,012	Adam (0.05)	1800	20	0.1909	0.0503	20.15
Original Data Pre trained CNN (VGG19) on ILSVRC2012	4QAED - Melspectrogram (900 samples - 942x128)		61,081,860	Adam (0.005)	300	80	0.6786	0.0541	77.8
Original Data + All Augmented Data CNN	Power Shift (900) Time Shift (900) Pitch Shift (900) Time stretch (900) + 4QAED - Melspectrogram (4500 samples - 942x128)	16 16 16 16	624,360	Adam (0.005)	120	450	0.6636	0.0503	996.7
Original Data + Time Shift Augmented Data CNN	Time Shift (900) + 4QAED - Melspectrogram (1800 samples - 942x128)	16 16 16 16	624,360	SGD (0.01)	150	100	0.6259	0.0492	542.3
Original Data + Pitch Shift Augmented Data CNN	Pitch Shift (900) + 4QAED - Melspectrogram (1800 samples - 942x128)	16 16 16 16	624,360	SGD (0.01)	150	100	0.635	0.0525	530.1
Original Data + Time Stretch Augmented Data CNN	Time Stretch (900) + 4QAED - Melspectrogram (1800 samples - 942x128)	16 16 16 16	624,360	SGD (0.01)	150	100	0.5964	0.048	529.1
Original Data + Power Shift Augmented Data CNN	Power Shift (900) + 4QAED - Melspectrogram (1800 samples - 942x128)	16 16 16 16	624,360	SGD (0.01)	150	100	0.6372	0.0466	520.56
Divided MEVD Data (2s/250ms) CNN-LSTM	MEVD - Melspectrogram (2s/250ms) (900 samples - 281x86x128)	16 16 16	1,914,200	Adam (0.005)	50	80	0.3447	0.1286	2011.7
Original Data + GAN Augmented Data (200 samples) CNN	50 per quad GAN Generated + 4QAED - Melspectrogram (900 samples - 942x128)	16 16 16	624,360	Adam (0.01)	200	100	0.4804	0.0554	300.9
Original Data + GAN Augmented Data (100 samples) CNN	25 per quad GAN Generated + 4QAED - Melspectrogram (900 samples - 942x128)	16 16 16	624,360	SGD (0.01)	200	100	0.6044	0.0296	551.2
Original Data + Voice Isolated Double CNN	4QAED - Melspectrogram + Voice only - Melspectrogram (1800 samples - 942x128)	16 16 16 16	1,318,764	Adam (0.0005)	150	200	0.6338	0.0492	609.8
Original Data (to AV) Double CNN	4QAED - Melspectrogram (900 samples - 942x128)	16 16 16 16	1,270,386	SGD (0.01)	300	100	0.5036	0.0473	537.4

This page is intentionally left blank.

Chapter 4

Music Emotion Variation Detection

The purpose of this section is to introduce the data used for the MEVD problem as well as to explain the process and the decisions behind the experiments.

4.1 Data

The original dataset used by Yang et al. [57] contained the arousal and valence values annotated across 25 seconds clips of 194 different songs. Various problems regarding this dataset were addressed by Panda [47] which resulted in a continuous annotation of 29 entire songs from the same data. The original dataset presented a significant proximity to the origin of the arousal and valence axis with over 71% of the songs being in the [0, 0.5] interval for both arousal and valence values. From that dataset, the full songs were extracted and Panda proposed that the oriental songs were excluded, leaving 57 entire songs. These were annotated continuously by two annotators and only samples with a 80% or more agreement rate were considered. This downsampling process, although necessary, brought down the number of songs to 29 and the quadrant distribution poses an even more difficult problem as, for example, the third quadrant is strongly underrepresented (see Figure 4.1).

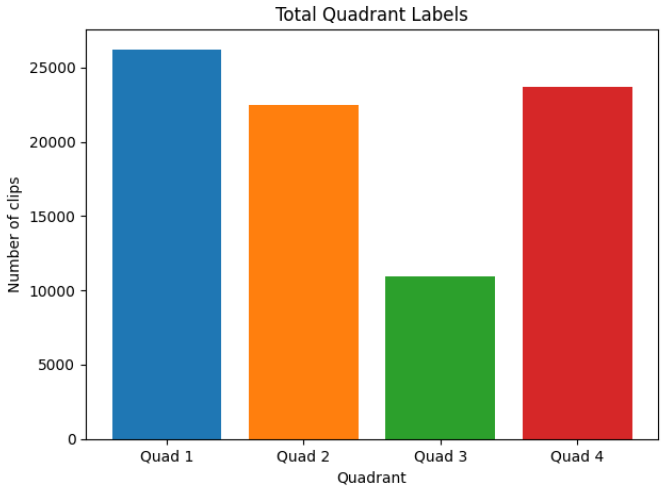


Figure 4.1: Total quadrant distribution

As for the continuous annotations for each song, we can see that a maximum of two quadrants are represented in a single song. Also, apart from one song, in all cases of songs

where only one quadrant was identified, that quadrant is the fourth one (see Figure 4.2).

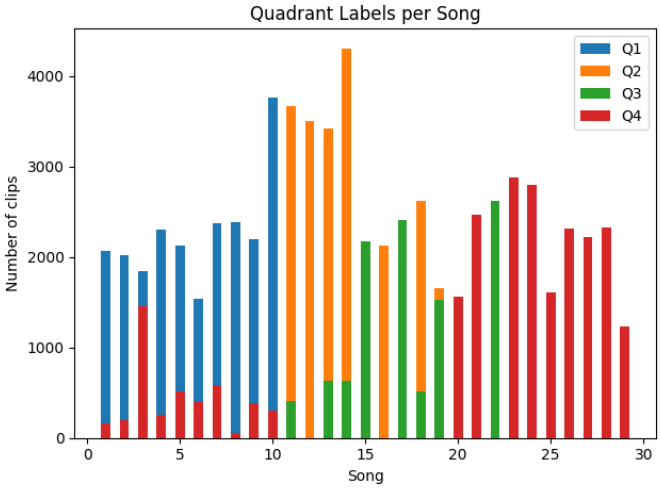


Figure 4.2: Quadrant distribution per song

The dataset was tested with **1** and **2-second windows** and with an **overlap** of **250** as well as **100 milliseconds**.

4.2 Methods and Results

The purpose of this section is to explain the MEVD approaches and their results. It is important to note that every model was evaluated with a **4-fold cross validation repeated 10 times**. The reason why a 4 fold was chosen was the need for the training set to be significantly larger than the test set while still maintaining a test set with all quadrant being present (21 for training, 8 for testing).

4.2.1 Deep Learning

The main dataset that was used was the 2-second windows with a 100-millisecond overlap as the 1 second windows did not provide the model with the sufficient information given that the accuracy in the test set did not surpass the 20% accuracy rate. The 100 milliseconds overlap was chosen as it gave more samples per song, a total of 296930 samples.

Unfortunately, the chosen dataset did not make a major significance as the primary problem remained: overfit. Attempts were made in order to fight this, such as pre-training the model with the 900 samples dataset (sampled at 22kHz to match the window size of the MEVD dataset) split into 2-second windows with the 100 milliseconds overlap.

The pre-training of the feature extraction section of the network was made with a sequence-to-one approach (see Figure 4.3), which means that a series of, in this case 281 clips representing the 2-second windows, runs through the model before it returns a single output, which refers to the entire sequence of windows. Therefore, we have an output for each 30-second sample. This approach is different from the one used in the MEVD models, which is a sequence-to-sequence method, where each output corresponds to a single frame, a 2-second window (see Figure 4.3 and 4.5).

In the end, the result did not differ, as the best model (see Figure 4.4) outputted an F1-

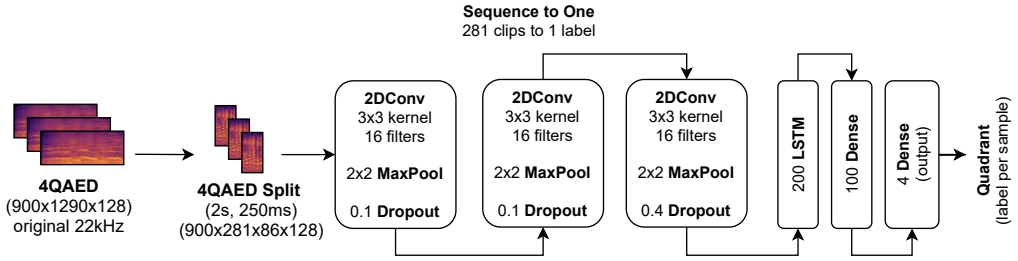


Figure 4.3: Static CNN-LSTM model

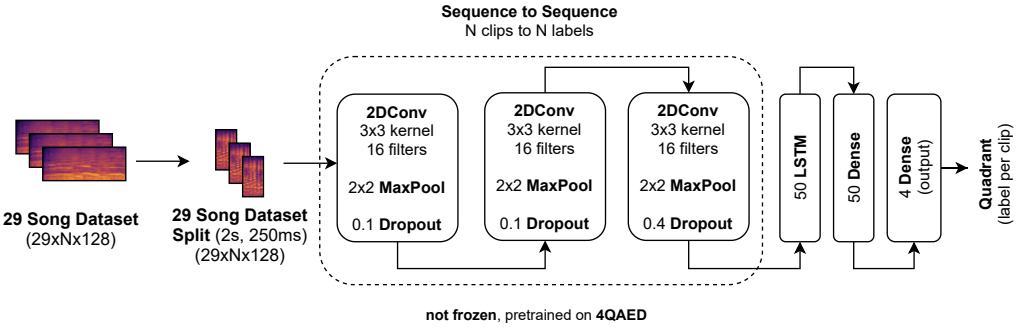


Figure 4.4: MEVD CNN-LSTM model

Score of **20.25%** with an accuracy of 47.5%. This was reached using a similar approach to the static models, with three convolutional layers and a dropout layer in between, to prevent the model from overfitting. The LSTM layer was added to account for changes given that, as previously explained, it can recognize past results. In order to try to give more information to the model, a BiLSTM layer was used (see Figure 4.5) but the results were not statistically significant, with an F1-Score of **20.68%** and an accuracy of 49.1%.

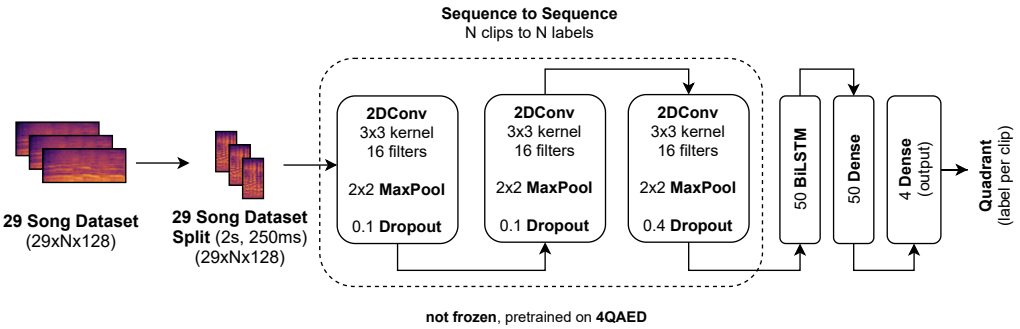


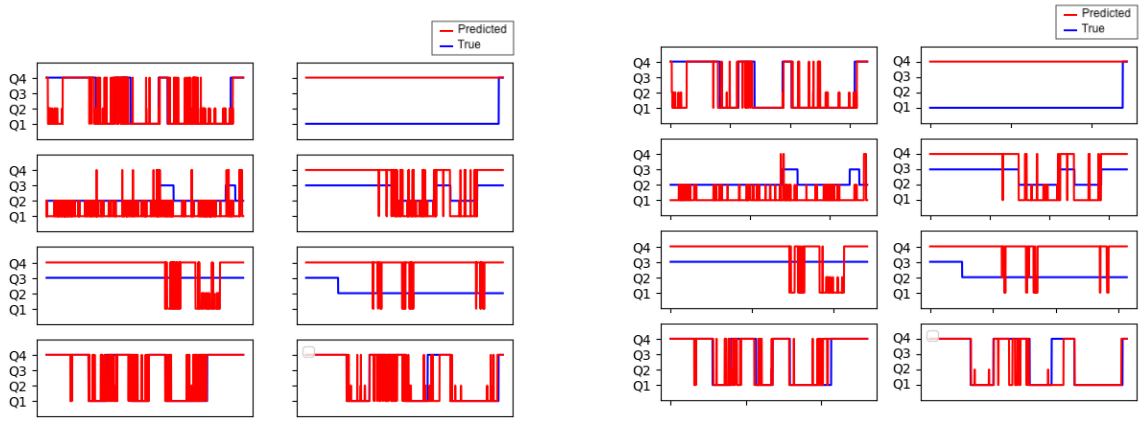
Figure 4.5: MEVD CNN-BiLSTM model

4.3 Results Analysis

Overall, the results do not impress but they follow the same line as the results from Panda's MEVD approach [47] (accuracy of 47.42%), suggesting that there is a glass ceiling with this dataset due to the low amount of samples. One of the issues is that the initial model, even being pre-trained on the 4QAED dataset, after a few epochs immediately started to assign the same quadrant to the songs that had very few changes in terms of emotion (see Figure

4.6). As a consequence, the number of neurons in the latter layers had to be stripped-down (see Figure 4.4 and 4.5). It is also clear that the model struggles to give an accurate label depending on the train and test set, as the standard deviation on both models results reached a value much greater than any model from the static evaluation with 57.28% for the CNN-LSTM model and 51.02% for the CNN-BiLSTM model (see Table 4.1).

Several different methods were attempted to make the prediction significantly softer, that is to say, to control the sudden peaks and valleys present in the output labels across the song (see Figure 4.6a). Such methods consisted in saving the output labels for each clip and creating a DNN that received it as an input, trained based on the real labels per clip. However, this did not perform as expected. Given the various different sizes possible for the output (different size songs), it consistently trained the model to output the same label for the entirety of the song. We discovered that by using the mode, with a 5 clip window, the visual output improved (see Figure 4.6b). However, it did not translate into a better F1-Score or accuracy.



(a) LSTM output per clip

(b) LSTM output comparison after being filtered (mode)

Figure 4.6: MEVD model output for the test set over the entire duration of the songs

In terms of computational effort, the RNN layers, namely the LSTM and BiLSTM layers, required much more training time, as the dataset had to be split into smaller samples and the number of trainable parameters increased, specially in the model with the input data of the split 4QAED dataset, taking over 33 hours to train on a 10-fold cross validation.

All the best model's hyperparameters, results and time of computation are present in Table 4.1.

Table 4.1: Best results for DL MEVD approaches

Model	Type	Filters	# of parameters	Optimizer (lrate)	Batch	Epochs	F1-Score (mean)	F1-Score (std)	Time (min)
MEVD Dataset CNN-LSTM	MEVD 29 Musics - Melspectrogram	16 16 16	449,850	SGD (0.01)	1	50	0.2025	0.5728	512.5
MEVD Dataset CNN-BiLSTM	MEVD 29 Musics - Melspectrogram	16 16 16	894,550	SGD (0.01)	1	50	0.2068	0.5102	545.4
Split Data (2s/250ms) CNN-LSTM	MEVD 29 Musics - Melspectrogram	16 16 16	1,914,200	Adam(0.005)	50	80	0.3447	0.1286	2011.7

Chapter 5

Conclusion and Future Work

The aim of this chapter is to summarize and briefly discuss the main findings and contributions from this project. As several different approaches were tackled, we feel that we successfully explored the datasets and achieved significant results (Section 5.1). However, some important aspects deserve further attention in future work (Section 5.2).

5.1 Conclusion

In retrospect, the expected time to create and test all approaches above was not realistic. Being the first time working with DL models (with a real-world dataset), especially with CNNs, the planed deadlines were not pragmatic. The extended time frame allowed for a better dive into different perspectives and fresh technologies such as the GAN models, and the *Spleeter* tool, but more importantly, it was possible to apply the knowledge gained along the project to approaches such as the DNN with the input of all the features. To put into perspective, in the early stages, this model's performance was so poor that we considered not including it. This and several other performance improvements, as well as a wider range of experiments than expected, goes to show the effect of a better experience and familiarity with the technologies and the problem at hand.

Another aspect to note is the infinite combination of possible hyperparameters and architectures that can (and should) be tested. Our aim was to cover as many technologies and different methods as possible, with a primary focus on static MER. Evidently, an effort was made to reach the best possible results within the several approaches and, fortunately, that was the case. The hybrid model proved to be the best approach with an F1-Score of **88.45%**, the best result so far in the literature for the dataset in question.

As previously pointed out, the number of samples is the primary problem of both datasets, predominantly for the MEVD dataset as the model was not able to perform better than an F1-Score of *20.68%*. The use of LSTM layers also did not bring any improvement to the static evaluation. One possible reason is the effective size of the window, as it can be too small for the model to learn from.

5.2 Future work

Therefore, we propose that the following points, be considered for future experiments:

- extending the database for both the static MER and MEVD problems, with a special consideration for the MEVD;
- diving deeper into the audio augmentation solutions, for both static MER and MEVD problem as it proved to improve the overall result (static MER);
- tuning and experimenting with different architectures and hyperparameters for the DL approaches;
- experimenting with various others parameters for the creation of the mel-spectrogram;
- exploring a different method to smooth out the output from the MEVD models in order to prevent unexpected changes and hopefully raise the results;
- experimenting with a sequence-to-one approach for the MEVD problem (i.e. dividing the dataset in a way that a series of clips correspond to a single output instead of being an output for each clip).

We optimistically hope that, by following these suggestions and with a great effort and dedication, higher results can be reached.

Bibliography

[1] Renato Panda, Ricardo Malheiro, and Rui Pedro Paiva. Novel audio features for music emotion recognition. *IEEE Transactions on Affective Computing*, PP:1–1, 03 2018. doi: 10.1109/TAFFC.2018.2820691.

[2] Yakob Chandra, Lay Christian, Hanny Juwitasary, Robert Atmojo, and William Febrianto. Analysis factors of intention to use music as a service application: A case study of spotify application. pages 187–192, 11 2018. doi: 10.1109/IC3INA.2018.8629521.

[3] Biju Das Pranjul Agrahari, Ayush Singh Tanwar and Prof. Pankaj Kunekar. Musical therapy using facial expressions. *International Research Journal of Engineering and Technology - IRJET*, 7(p-ISSN: 2395-0072), 2020. URL `https://www.irjet.net/archives/V7/i1/IRJET-V7I1199.pdf`.

[4] Arefin Huq, Juan Pablo Bello, and Robert Rowe. Automated music emotion recognition: A systematic evaluation. *Journal of New Music Research*, 39(3):227–244, 2010. doi: 10.1080/09298215.2010.513733. URL `https://doi.org/10.1080/09298215.2010.513733`.

[5] Laura S. Sakka and Patrik N. Juslin. Emotion regulation with music in depressed and non-depressed individuals: Goals, strategies, and mechanisms. *Music & Science*, 1: 2059204318755023, 2018. doi: 10.1177/2059204318755023. URL `https://doi.org/10.1177/2059204318755023`.

[6] Keunwoo Choi, György Fazekas, Kyunghyun Cho, and Mark Sandler. A tutorial on deep learning for music information retrieval, 2018.

[7] Paul R. Kleinginna and Anne M. Kleinginna. A categorized list of emotion definitions, with suggestions for a consensual definition. 12 1981. doi: 10.1007/BF00992553.

[8] Rui Pedro Paiva Pedro Vale. The role of artist and genre on music emotion recognition. 11 2017. URL `http://hdl.handle.net/10362/26303`.

[9] Alf Gabrielsson. Emotion perceived and emotion felt: Same or different? *Musicae Scientiae*, 5:123–147, 01 2002. doi: 10.1177/10298649020050S105.

[10] yi-hsuan Yang and Homer Chen. Machine recognition of music emotion: A review. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3, 05 2012. doi: 10.1145/2168752.2168754.

[11] Paul Ekman. An argument for basic emotions. *Cognition and Emotion*, 6(3-4): 169–200, 1992. doi: 10.1080/02699939208411068. URL `https://doi.org/10.1080/02699939208411068`.

[12] Marcel Zentner, Didier Grandjean, and Klaus Scherer. Emotions evoked by the sound of music: Characterization, classification, and measurement. *Emotion (Washington, D.C.)*, 8:494–521, 09 2008. doi: 10.1037/1528-3542.8.4.494.

[13] Tuomas Eerola and Jonna Vuoskoski. A comparison of the discrete and dimensional models of emotion in music. *Psychology of Music*, 01 2011. doi: 10.1177/0305735610362821.

[14] Kate Hevner. Experimental studies of the elements of expression in music. *The American Journal of Psychology*, 48(2):246–268, 1936. ISSN 00029556. URL `http://www.jstor.org/stable/1415746`.

[15] James Russell. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39:1161–1178, 12 1980. doi: 10.1037/h0077714.

[16] Jonathan Posner, James Russell, and Bradley Peterson. The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development and psychopathology*, 17:715–34, 02 2005. doi: 10.1017/S0954579405050340.

[17] Juan Gómez Cañón, Estefanía Cano, Perfecto Herrera, and Emilia Gómez. Transfer learning from speech to music: towards language-sensitive emotion recognition models. 01 2021. doi: 10.5281/zenodo.4076791.

[18] Juan Sebastián Gómez Cañón, P. Herrera, E. Gómez, and Estefanía Cano. The emotions that we perceive in music: the influence of language and lyrics comprehension on agreement. *ArXiv*, abs/1909.05882, 2019.

[19] Owen Meyers. A mood-based music classification and exploration system. 10 2007.

[20] Paul Gilbert. The biopsychology of mood and arousal. robert e. thayer. *The Quarterly Review of Biology*, 67(3):406–406, 1992. doi: 10.1086/417761. URL `https://doi.org/10.1086/417761`.

[21] Sandrine Vieillard, Isabelle Peretz, Nathalie Gosselin, Stéphanie Khalfa, Lise Gagnon, and Bernard Bouchard. Happy, sad, scary and peaceful musical excerpts for research on emotions. *Cognition & Emotion - COGNITION EMOTION*, 22:720–752, 06 2008. doi: 10.1080/02699930701503567.

[22] Shuo-Yang Wang, Ju-Chiang Wang, yi-hsuan Yang, and Hsin-min Wang. Towards time-varying music auto-tagging based on cal500 expansion. volume 2014, pages 1–6, 07 2014. doi: 10.1109/ICME.2014.6890290.

[23] Mehmet Bilal Er and Ibrahim Aydilek. Music emotion recognition by using chroma spectrogram and deep visual features. *International Journal of Computational Intelligence Systems*, 12, 12 2019. doi: 10.2991/ijcis.d.191216.001.

[24] Rajib Sarkar, Sombuddha Choudhury, Saikat Dutta, Aneek Roy, and Sanjoy Saha. Recognition of emotion in music based on deep convolutional neural network. *Multimedia Tools and Applications*, 79, 01 2020. doi: 10.1007/s11042-019-08192-x.

[25] Mohammad Soleymani, Micheal N. Caro, Erik M. Schmidt, Cheng-Ya Sha, and Yi-Hsuan Yang. 1000 songs for emotional analysis of music. In *Proceedings of the 2nd ACM International Workshop on Crowdsourcing for Multimedia*, CrowdMM '13, page 1–6, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450323963. doi: 10.1145/2506364.2506365. URL `https://doi.org/10.1145/2506364.2506365`.

[26] Du Pengfei, Xiaoyong Li, and Yali Gao. Dynamic music emotion recognition based on cnn-bilstm. 04 2020.

[27] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1139–1147, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL `https://proceedings.mlr.press/v28/sutskever13.html`.

[28] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[29] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, USA, 2016. `http://www.deeplearningbook.org`.

[30] Keunwoo Choi, György Fazekas, Mark B. Sandler, and Kyunghyun Cho. Convolutional recurrent neural networks for music classification. *CoRR*, abs/1609.04243, 2016. URL `http://arxiv.org/abs/1609.04243`.

[31] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U. Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning, 2017.

[32] Giovanni Mariani, Florian Scheidegger, Roxana Istrate, Costas Bekas, and Cristiano Malossi. Bagan: Data augmentation with balancing gan, 2018.

[33] Renato Panda and Rui Pedro Paiva. Using support vector machines for automatic mood tracking in audio music. In *Audio Engineering Society Convention 130*, May 2011. URL `http://www.aes.org/e-lib/browse.cfm?elib=15845`.

[34] K. Hevner. Experimental studies of the elements of expression in music. *American Journal of Psychology*, 48:246–268, 1936.

[35] Yading Song, Simon Dixon, and Marcus Pearce. Evaluation of musical features for emotion classification. 10 2012.

[36] K. Markov and T. Matsui. Music genre and emotion recognition using gaussian processes. *IEEE Access*, 2:688–697, 2014. ISSN 2169-3536. doi: 10.1109/ACCESS.2014.2333095.

[37] Jochen Görtler, Rebecca Kehlbeck, and Oliver Deussen. A visual exploration of gaussian processes. *Distill*, 2019. doi: 10.23915/distill.00017. https://distill.pub/2019/visual-exploration-gaussian-processes.

[38] Renato Panda, Bruno Rocha, and Rui Pedro Paiva. Music emotion recognition with standard and melodic audio features. *Applied Artificial Intelligence*, 29(4):313–334, 2015. doi: 10.1080/08839514.2015.1016389. URL `https://doi.org/10.1080/08839514.2015.1016389`.

[39] R. Panda, R. M. Malheiro, and R. P. Paiva. Audio features for music emotion recognition: a survey. *IEEE Transactions on Affective Computing*, pages 1–1, 2020. doi: 10.1109/TAFFC.2020.3032373.

[40] Yeong-Seok Seo and Jun-Ho Huh. Automatic emotion-based music classification for supporting intelligent iot applications. *Electronics*, 8:164, 02 2019. doi: 10.3390/electronics8020164.

[41] Yazhong Feng, Yueting Zhuang, and Yunhe Pan. Popular music retrieval by detecting mood. pages 375–376, 01 2003. doi: 10.1145/860435.860508.

[42] Mie Mie Oo and Lwin Lwin Oo. *Fusion of Log-Mel Spectrogram and GLCM Feature in Acoustic Scene Classification*, pages 175–187. Springer International Publishing, Cham, 2020. ISBN 978-3-030-24344-9. doi: 10.1007/978-3-030-24344-9_11. URL https://doi.org/10.1007/978-3-030-24344-9_11.

[43] Xin Liu, Qingcai Chen, Xiangping Wu, Yan Liu, and Yang Liu. Cnn based music emotion classification. 04 2017.

[44] Tong Liu, Li Han, Liangkai Ma, and Dongwei Guo. Audio-based deep music emotion recognition. volume 1967, page 040021, 05 2018. doi: 10.1063/1.5039095.

[45] Juan Sebastián Gómez Cañón, P. Herrera, E. Gómez, and Estefanía Cano. The emotions that we perceive in music: the influence of language and lyrics comprehension on agreement. *ArXiv*, abs/1909.05882, 2019.

[46] Emery Schubert. Modeling Perceived Emotion With Continuous Musical Features . *Music Perception*, 21(4):561–585, 06 2004. doi: 10.1525/mp.2004.21.4.561. URL https://doi.org/10.1525/mp.2004.21.4.561.

[47] Renato Panda and Rui Pedro Paiva. Using support vector machines for automatic mood tracking in audio music. volume 1, 09 2011.

[48] Konstantin Markov and Tomoko Matsui. Music genre and emotion recognition using gaussian processes. *Access, IEEE*, 2:688–697, 01 2014. doi: 10.1109/ACCESS.2014.2333095.

[49] Miroslav Malík, Sharath Adavanne, Konstantinos Drossos, Tuomas Virtanen, Dasa Ticha, and Roman Jarina. Stacked convolutional and recurrent neural networks for music emotion recognition. 07 2017.

[50] X. Li, J. Tian, M. Xu, Y. Ning, and L. Cai. Dblstm-based multi-scale fusion for dynamic emotion prediction in music. In *2016 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2016. doi: 10.1109/ICME.2016.7552956.

[51] Serhat Hizlisoy, Serdar Yildirim, and Zekeriya Tufekci. Music emotion recognition using convolutional long short term memory deep neural networks. *Engineering Science and Technology, an International Journal*, 2020. ISSN 2215-0986. doi: https://doi.org/10.1016/j.jestch.2020.10.009. URL http://www.sciencedirect.com/science/article/pii/S2215098620342385.

[52] Kenji Kira and Larry A. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *AAAI*, 1992.

[53] Renato Panda, Hugo Redinho, Carolina Gonçalves, Ricardo Malheiro, and Rui Pedro Paiva. How does the spotify api compare to the music emotion recognition state-of-the-art? Zenodo, Jun 2021. doi: 10.5281/zenodo.5045100.

[54] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[55] Rafael L. Aguiar, Yandre M.G. Costa, and Carlos N. Silla. Exploring data augmentation to improve music genre classification with convnets. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2018. doi: 10.1109/IJCNN.2018.8489166.

[56] Darryl Griffiths, Stuart Cunningham, Jonathan Weinel, and Richard Picking. A multi-genre model for music emotion recognition using linear regressors. *Journal of New Music Research*, 0(0):1–18, 2021. doi: 10.1080/09298215.2021.1977336. URL `https://doi.org/10.1080/09298215.2021.1977336`.

[57] yi-hsuan Yang, Yu-Ching Lin, Ya-Fan Su, and Homer Chen. A regression approach to music emotion recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16:448 – 457, 03 2008. doi: 10.1109/TASL.2007.911513.

Appendices

Appendix A

Table 1: Top 100 features used by Panda et al. [1]

Feature Type	Name
Standard	Fluctuation (std)
Standard	Fluctuation (skewness)
Standard	Fluctuation (max)
Standard	Events Density
Standard	High-frequency Energy (std)
Standard	High-frequency Energy (skewness)
Standard	High-frequency Energy (min)
Standard	Spectral Flux (skewness)
Standard	Spectral Centroid (std)
Standard	Spectral Skewness (std)
Standard	Spectral Skewness (max)
Standard	Spectral Entropy (std)
Standard	Spectral Entropy (min)
Standard	Inharmonicity (mean)
Standard	Inharmonicity (std)
Standard	Inharmonicity (min)
Standard	Tonal Centroid 3 (std)
Standard	Rolloff (MeanA/StdM)
Standard	MFCC[1]0 (MeanA/StdM)
Standard	MFCC1 (MeanA/StdM)
Standard	MFCC0 (StdA/MeanM)
Standard	MFCC1 (StdA/MeanM)
Standard	Rolloff (StdA/StdM)
Standard	MFCC1 (StdA/StdM)
Standard	Rolloff (mean)
Standard	MFCC1 (mean)
Standard	MFCC1 (std)
Standard	MFCC1 (max)
Standard	LSP[2]_1 (mean)
Standard	LSP_3 (min)
Standard	LSP_4 (std)
Standard	LSP_4 (min)
Standard	LSP_5 (std)
Standard	LSP_5 (min)
Standard	SFM[3]_6 (mean)
Standard	SFM_7 (mean)
Standard	SFM_7 (skewness)
Standard	SFM_8 (skewness)
Standard	SFM_9 (mean)
Standard	SFM_9 (skewness)
Standard	SFM_10 (skewness)
Standard	SFM_11 (skewness)
Standard	SFM_12 (mean)
Standard	SFM_15 (mean)
Standard	SFM_15 (std)
Standard	SFM_17 (std)
Standard	SCF[4]_9 (mean)
Standard	SCF_10 (mean)
Standard	SCF_11 (mean)
Standard	SCF_12 (mean)
Standard	SCF_13 (mean)

[1]Mel-Frequency Cepstral Coefficients
[2]Line Spectral Pairs
[3]Spectral Flatness Measure
[4]Spectral Crest Factor

Feature Type	Name
Standard	SCF_15 (mean)
Standard	SCF_15 (std)
Standard	SCF_16 (std)
Standard	SCF_17 (mean)
Standard	FFT[5] Spectrum - Average Power Spectrum (mean)
Standard	FFT Spectrum - Average Power Spectrum (median)
Standard	FFT Spectrum - Spectral 2nd Moment (median)
Standard	FFT Spectrum - Skewness (median)
Standard	Dynamic Loudness (C & F) - Sharpness (std)
Standard	Loudness (MG & B PsySound2) - Loudness (skewness)
Standard	Loudness (MG & B PsySound2) - SharpnessA (skewness)
Standard	Loudness (MG & B PsySound2) - Volume (skewness)
Standard	Loudness (MG & B PsySound2) - Tonal Dissonance (HK) (std)
Standard	Loudness (MG & B PsySound2) - Tonal Dissonance (S) (skewness)
Standard	Loudness (MG & B PsySound2) - Spectral Dissonance (HK) (max)
Standard	Loudness (MG & B PsySound2) - Spectral Dissonance (HK) (std)
Standard	Loudness (MG & B PsySound2) - Spectral Dissonance (S) (skewness)
Standard	Pitch (Terhardt) - Chord Change Likelihood (median)
Standard	Pitch (SWIPEP) - SWIPEP Pitch Strength (mean)
Standard	Loudness (Moore, Glasberg and Baer) - Short-term Loudness (skewness)
Novel	ORIGINAL-EXPRESSIVE_TECHNIQUES-Glissando Extent (Std)
Novel	ORIGINAL-EXPRESSIVE_TECHNIQUES-Glissando Length (Std)
Novel	ORIGINAL-EXPRESSIVE_TECHNIQUES-Glissando Slope (Std)
Novel	ORIGINAL-EXPRESSIVE_TECHNIQUES-Glissando Direction (Up)
Novel	ORIGINAL-TEXTURE-Musical Layers (Mean)
Novel	ORIGINAL-TEXTURE-Musical Layers (Max)
Novel	ORIGINAL-TEXTURE-Musical Layers (Std)
Novel	ORIGINAL-TEXTURE-ML1-Monophonic Texture (Percentage)
Novel	ORIGINAL-TEXTURE-State Transitions ML1 - ML0 (Per Sec)
Novel	ORIGINAL-TEXTURE-State Transitions ML1 - ML2 (Per Sec)
Novel	ORIGINAL-TEXTURE-State Transitions ML2 - ML3 (Per Sec)
Novel	ORIGINAL-EXPRESSIVE_TECHNIQUES-ART-Other Notes Duration (Mean)
Novel	ORIGINAL-EXPRESSIVE_TECHNIQUES-Tremolo Notes in Cents (Mean)
Novel	ORIGINAL-EXPRESSIVE_TECHNIQUES-Tremolo Notes in Cents (Max)
Novel	ORIGINAL-EXPRESSIVE_TECHNIQUES-Tremolo Notes in Cents (Min)
Novel	ORIGINAL-EXPRESSIVE_TECHNIQUES-Tremolo Higher Notes Coverage (C4+)
Novel	ORIGINAL-EXPRESSIVE_TECHNIQUES-Vibrato Rate (Std)
Novel	ORIGINAL-EXPRESSIVE_TECHNIQUES-Vibrato Rate (Kurtosis)
Novel	ORIGINAL-EXPRESSIVE_TECHNIQUES-Vibrato Extent (Std)
Novel	ORIGINAL-EXPRESSIVE_TECHNIQUES-Vibrato Extent (Kurtosis)
Novel	ORIGINAL-EXPRESSIVE_TECHNIQUES-Vibrato Length (Kurtosis)
Novel	ORIGINAL-EXPRESSIVE_TECHNIQUES-Vibrato Length (Skewness)
Novel	ORIGINAL-EXPRESSIVE_TECHNIQUES-Vibrato Base Freq (Min)
Novel	ORIGINAL-EXPRESSIVE_TECHNIQUES-Vibrato Base Freq (Std)
Novel	ORIGINAL-EXPRESSIVE_TECHNIQUES-Vibrato Base Freq (Kurtosis)
Novel	ORIGINAL-EXPRESSIVE_TECHNIQUES-Vibrato Higher Notes Coverage (C4+)
Novel	ORIGINAL-EXPRESSIVE_TECHNIQUES-Vibrato to Non Vibrato Notes Ratio
Novel	ORIGINAL-VAT-Probability of Creaky Voice (Skewness)
Novel	VOICE-TEXTURE-State Transitions ML0 - ML1 (Per Sec)

[5]Fast Fourier Transform