Hugo José Amado Redinho

MERGE Audio: Audio Analysis and Feature Engineering for Music Emotion Recognition
MSC Thesis

1 2 9 0
UNIVERSIDADE Ð COIMBRA

1 2 9 0

UNIVERSIDADE Ð
COIMBRA

Hugo José Amado Redinho

MERGE Audio: Audio Analysis and Feature Engineering for Music Emotion Recognition
MSC Thesis

VOLUME 1

Dissertation in the context of the Master in Informatics Engineering, specialization in Intelligent Systems, advised by Professors Rui Pedro Paiva, Renato Panda and Ricardo Malheiro and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

September of 2023

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE Ð
COIMBRA

DEPARTMENT OF INFORMATICS ENGINEERING

Hugo José Amado Redinho

MERGE Audio: Audio Analysis and Feature Engineering for Music Emotion Recognition

Dissertation in the context of the Master in Informatics Engineering, specialization in Intelligent Systems, advised by Professors Rui Pedro Paiva, Renato Panda and Ricardo Malheiro and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

September 2023

This page is intentionally left blank.

This work is funded by the FCT - Foundation for Science and Technology, I.P./MCTES through national funds (PIDDAC), within the scope of project MERGE (Music Emotion Recognition - Next Generation) - PTDC/CCI-COM/3171/2021 and CISUC R&D Unit - UIDB/00326/2020 or project code UIDP/00326/2020.

FCT Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR

CISUC

This page is intentionally left blank.

Acknowledgements

Ever since I joined this research group in September of 2019, it has been one of the best experiences of my life. I have learned so much throughout the years and got interested in things that I would never thought I would have liked. Throughout the years many people have passed through this research group, with some being close friends of mine. However, the three people that have always stayed and have been integral to the completion of this work have been my three advisors, Prof. Renato Panda, Prof. Ricardo Malheiro and Prof. Rui Pedro Paiva. For all the countless hours spent guiding me in the right path, pointing out mistakes and listening to my concerns, I thank you from the bottom of my heart.

I would like to extend a further amount of gratitude to Prof. Rui Pedro, who has always guided me in the right path ever since becoming literally and figuratively my teacher, and whom without this thesis would not have happened.

I would also like to thank my colleague Pedro Louro, who has been a helping hand throughout the months that took me to write this thesis. He has helped me when I needed the most, and made sure my focus was on the right path and was always ready to deal with things in order for me to be as distraction free as possible.

I also thank all my friends that have stuck with me throughout the years, given me moments to relax and unwind, and told me that everything will work out in the end, even thought it might not have seen at the time, those words have stuck with me after all this time.

Furthermore, throughout the years I have met many people at University of Coimbra, and made friendships that I will take for life. To my friends that stuck with me late hours doing projects, listening to my rambles on my code doesn't work when it should, I thank you from the most bottom point of my heart, as I would not be here today cherishing all the memories I made without your help. For anyone that might have interacted with me throughout all these years, given me the slightest bit of help and motivation, I would like to thank you as well.

And finally, I want to thank the people without which none of this would have been even remotely possible, my family. First I want to thank my brother for always believing in me and supporting me, even when I doubted myself, and for not being afraid to put me in line when I stepped out of it. The motivational talks that might have seen to have fallen in an empty void instead were the last push I needed to give it my all to produce my best work. Secondly, I want to thank my parents for always believing in me and supporting me, and for making sure I had everything I could ever require and more in to thrive.

This page is intentionally left blank.

Abstract

With the increase of widely available digital streaming options for music, the interest in the field of music emotion recognition has seen the same increasing effect. This field is still dominated by classical approaches that use feature engineering to classify the perceived emotion of a song. Furthermore, in recent years, there has been a surge of deep learning approaches that use neural networks to tackle this same problem. However, these approaches suffer from various problems such as the use of small, private, or low quality datasets, as well as the use of features not designed for emotion classification, amongst others.

This work proposes a set of three new datasets, denominated Music Emotion Recognition - Next Generation (MERGE), with three components: audio, lyrics and bi-modal. These datasets are an extension of the previous 4QAED dataset (Panda, 2019) and achieved F1-scores of 71% using the same feature set as 4QAED, while having a much greater size.

Furthermore, in this work, we propose a set of new emotionally relevant features to help tackle the problem aforementioned using techniques such as automatic music transcription with tools such as Magenta MT3 (3.5.2). From this framework, a set of features extracted from the outputted MIDI file are proposed.

Finally, using the percussion stem extracted from Demucs (3.6.3), a novel set of features extracted from the percussion track is also proposed. A subset of this novel set of features achieved an overall F1-Score of 74.1% on the MERGE_Bimodal_-Complete dataset (See Section 4.1.2 for further details on the datasets).

Keywords

Music emotion recognition, Music information retrieval, Audio analysis, Feature engineering, Music, Emotion

This page is intentionally left blank.

Resumo

Com o aumento das opções de streaming digital de música amplamente disponíveis, o interesse no domínio do reconhecimento de emoções musicais tem registado o mesmo efeito crescente. Este domínio ainda é dominado por abordagens clássicas que utilizam a engenharia de características musicais para classificar a emoção sentida de uma canção. Além disso, nos últimos anos, tem havido uma onda de abordagens de *deep learning* que utilizam redes neurais para resolver este mesmo problema. No entanto, estas abordagens sofrem de vários problemas, como a utilização de conjuntos de dados pequenos, privados ou de baixa qualidade, bem como a utilização de características não concebidas para a classificação de emoções, entre outros.

Este trabalho propõe um conjunto de três novos conjuntos de dados, denominados Music Emotion Recognition - Next Generation (MERGE), com três componentes: áudio, letras e bimodal. Estes conjuntos de dados são uma extensão do anterior conjunto de dados 4QAED (Panda, 2019) alcançaram *F1-Scores* de 71% usando o mesmo conjunto de características do 4QAED, tendo no entanto um tamanho muito maior.

Além disso, neste trabalho, propomos um conjunto de novas características emocionalmente relevantes para ajudar a resolver o problema acima mencionado, utilizando técnicas como a transcrição automática de música com ferramentas como o Magenta MT3 (3.5.2). A partir desta *framework*, é proposto um conjunto de características extraídas do ficheiro MIDI produzido.

Finalmente, utilizando a faixa de percussão extraída do Demucs (3.6.3), é também proposto um novo conjunto de características extraídas da faixa de percussão. Um subconjunto deste novo conjunto de características obteve uma pontuação F1 global de 74,1% no conjunto de dados MERGE_Bimodal_Complete (ver Secção 4.1.2 para mais detalhes sobre os conjuntos de dados).

Palavras-Chave

Reconhecimento de Emoção em Música, Recuperação de Informação em Música;, Análise de áudio, Criação de features, Música, Emoção

This page is intentionally left blank.

Contents

List of Figures

List of Tables

This page is intentionally left blank.

Abbreviations

4QAED 4 Quadrant Audio Emotion Dataset.

AMT Automatic Music Transcription.

BCRSN Bidirectional Convolutional Recurrent Sparse Network.

BPM Beats Per Minute.

CNN Convolutional Neural Network.

DL Deep Learning.

DSP Digital Signal Processor.

ERB Equivalent Rectangular Bandwith.

F0 Fundamental Frequency.

GP Gaussian Process.

GPU Graphical Processing Unit.

HWPS Harmonically Wrapped Peak Similarity.

IMT Instrument Music Transcription.

ISFTF Inverse Short-Time Fourier Transform.

ISMT Instrument Specific Music Transcription.

KNN K-nearest Neighbor.

LPC Linear Prediction Coefficient.

LPCC Linear Predictive Coding Coefficients.

LSP Linear Spectral Pair.

MAE Mean Absolute Error.

MER Music Emotion Recognition.

MERGE Music Emotion Recognition - Next Generation.

MEVD Music Emotion Variation Detection.

MFCC Mel-Frequency Cepstral Coefficients.

MIR Music Information Retrieval.

MIREX Music Information Retrieval eXchange.

ML Machine Learning.

NN Neural Network.

PCC Pearson Correlation Coefficient.

PLP Predominant Local Pulse.

PSD Power Spectral Density.

RBF Radial Basis Function.

RMS Root-Mean Square.

RMSE Root Mean Squared Error.

RNN Recurrent Neural Network.

SAR Signal to Artifacts Ratio.

SCF Spectral Crest Factor.

SDR Signal to Distortion Ratio.

SII-ASF Sequential-information-included Affect-salient Features.

SIR Signal to Interference Ratio.

SNR Signal to Noise Ratio.

SSM Self-Similarity Matrix.

STFT Short-Time Fourier Transform.

SVM Support Vector Machine.

ZCR Zero Crossing Rate.

Chapter 1

Introduction

Ever since the beginning of humanity, humans and music have always been an inseparable duo. Humans have used music for many different purposes throughout the years, from rituals, entertainment, connecting with other people, but most important of all, expressing their emotions.

Music can convey a wide range of emotions, from sadness to happiness, anger to peacefulness. In reality, one piece of music is able to convey a wide range of different emotions. With music pieces having this kind of ability, being able to understand what characteristics of a specific song or piece help convey these emotions is paramount.

Before, while music might have been reserved for the select few that could, for example, see it performed in theaters and orchestras, it rapidly became increasingly common to have cassette tapes of your favorite artists and albums. Still, this time period had the problem of listeners only having access to small sets of the whole library of music available. Nowadays, music is universal and exists in every part of society, moreover, every passing year having access to music is getting increasingly easier.

These days, with the appearance of digital streaming, not having access to music is no longer a problem. Any person is able to use any of the major digital streaming services, such as Spotify or Apple Music (Variety, 2022), and instantly get access to tens of millions of songs.

However, with the increase of the amount of available data, many problems arose, such as finding ways to categorize this huge library of music. There are many different possible types of classification, for example classifying based on the music genre or what type of music piece it is, such as acoustic or not. One of these types of classification is emotion classification.

The field of Music Emotion Recognition (MER) aims to tackle this problem, by creating a set of tools involving machine learning (ML) techniques to classify the emotion, or emotions, of a certain song. There are also subcategories of MER, such as Static MER, which aims to identify the dominant emotion of the complete song, as well as Music Emotion Variation Detection (MEVD), which aims to identify the changes in emotion in a song.

The field of MER has many practical uses, such as the automatic creation of playlists based on emotion, music recommendation, amongst others.

1.1 Problem and motivation

As stated before, with the massive growth in the amount of available music in recent years, there have been more and more efforts in the way of categorizing and enabling filtering of this huge amount of data to make it possible to cater to every person's interests. This has been expressed in many ways, such as automatic playlist generation, amongst others.

However, due to the nature of music, it is very hard to categorize songs, partly due to the sheer amount of variables that can change from song to song, such as singer, the number of instruments, notes, tempo, beats per minute, amongst many other characteristics. Sometimes, even within the same song, there are sudden changes that make any classification, especially emotion classification, especially challenging for even humans, let alone automatic classification systems.

The lack of audio emotionally-relevant features in the MER field is also a problem. Most approaches use a similar set of features that was originally proposed to address other audio analysis problems (e.g. speech recognition) and often lack emotional relevance (Panda et al., 2020a).

Furthermore, finding and creating datasets (particularly larger ones) with high quality annotations has proven to be a very hard task, with most of the studies being done on smaller datasets (less than a thousand songs). This also makes testing other approaches such as Deep Learning (DL) that require a huge amount of data practically impossible, with most studies having to resort to techniques such as data augmentation in order to obtain sufficient results.

Moreover, since there is a degree of subjectivity in emotion perception, it is also paramount that this is tackled properly. Efforts should be conducted to minimize its impact, such as having a high rate of agreement between annotators of the dataset.

This emotion classification problem has been tackled in many ways since 2003 (Feng et al., 2003), but even the simple task of solving basic emotion classification (such as classifying one song within 5 possible categories) has proven immensely difficult, with a "glass ceiling" of around 70% accuracy having been reached (Panda et al., 2020a). This result was obtained in the Music Information Retrieval eXchange (MIREX) task (a benchmark in the field), with this task consisting of classifying songs onto five possible categories.

There has also been research conducted on more complex problems such as multi-label classification or MEVD, with these approaches reporting even worse results (around 20%). (Aljanaki et al., 2017; Wu et al., 2014).

Thus, it is important to tackle the aforementioned basic problems before aiming to solve the complex ones. First, feature engineering and expansion/creation of datasets should be conducted. Afterwards, the next step is the exploration of Static MER (3.4) using a low number of classes (e.g. four). Finally, the problem of MEVD (A.1) should be addressed, building on the outcomes of the research conducted in Static MER.

1.2 Objectives and approaches

Thus, the objectives of this thesis that aim to address the aforementioned problems can be summarized as such [1]:

1. Create new audio features particularly in the category of musical texture, using tools for melody transcription and instrument identification. Furthermore, create novel features using the isolated percussion stem of the track.
2. Update and enlarge current datasets, both for Static MER and also for MEVD.

Table 1.1 summarizes all the objectives ranked on a priority scale of *High*, *Medium* and *Low*, representing their importance in regards to this work, with all possible efforts being made to accomplish them. Objectives with low priority were not possible to explore in this work, as time did not permit their exploration.

Objective	**Priority**
Static MER - Research and development of audio analysis and feature engineering approaches	High
Update current Static MER and MEVD Datasets	High
Static MER - Research and development of audio analysis and feature engineering approaches involving DL	Low
MEVD - Research and development of audio analysis and feature engineering approaches involving DL	Low
MEVD - Research and development of audio analysis and feature engineering approaches	Low

Table 1.1: Objectives of this work.

In order to better explain the scope of this work, fulfilling the project's objectives should answer the following questions:

- Are the newly created MERGE datasets viable for classical ML approaches and DL approaches?
- Do the newly contributed features help in the problem of emotion classification in Static MER?

[1] Two previous objectives were removed due to changes in the main focus of this work (Feature Engineering for MEVD approaches and creating a web application for MER).

1.3 Results and contributions

In this section, the main results obtained are presented, as well as the main contributions and limitations found during this work.

The main results obtained were:

- Testing of the new datasets and achieving reasonable F1-Scores when compared to 4QAED.
- Achieving **74.1% F1-Score** in MERGE_Bimodal_Complete using 250 features from MERGE_All feature set (See Section 4.1.2 for further details on the datasets).

As for the main contributions made with this work:

- Validation of the new MERGE datasets, which achieved reasonable F1-scores and thus provide a good option for future DL approaches;
- A new set of features extracted from the percussion track and MIDI files that proved to be relevant for MER.

The main limitations found during this work are:

- There is much confusion between the third and fourth Russell's emotion quadrants (See Section 2.1.3), which is something classical ML approaches are having a lot of difficulty solving.

1.4 Organization and planning

1.4.1 Experimental Environment

The presented experiments were mostly conducted on a Graphical Processing Unit (GPU) server, shared with the team. Due to the very demanding nature of most approaches experimented with, GPUs are required to properly develop and evaluate these in reasonable time. The specifications of the server are:

- Intel Xeon Silver 4214 CPU @ 2.20GHz x 48
- 3x NVIDIA Quadro P5000 16GB
- 7x NVIDIA RTX A5000 24GB
- 320GB RAM

However, this is a shared server with other students, which means that not all the resources are available on demand. This also means that there might be heavier and lighter loads in the server, changing the time it takes to complete the tasks.

The methodologies were mostly conducted in a Python 3.9.7 virtual environment for replicability purposes. Libraries such as *numpy* and *pandas* were utilized for data manipulation, as well *librosa* to manipulate audio signal data and *scikit-learn* was utilized for calculating the relevant metrics for analyzing performance.

1.4.2 Organization

First semester

Due to personally already having experience with the topics that this work encompasses from working on various projects from 2019 to 2022, the literature review for the creation of the State of the Art was planned to be done in the first one and a half months of this semester. Afterwards, the main focus would be expanding and updating the current datasets, while at the same time gathering information in regards to more specific topics such as feature engineering and audio analysis for both Static MER and MEVD datasets for the State of the Art.

After the objective of updating the current dataset had been completed, an evaluation of the current algorithms based on classical feature engineering techniques would be required to see the impact of the expansion of the datasets in the classification results.

This first semester went according to plan, with allowing not only for a deep dive on the current literature, the current used features, but also allowing for initial experimentation with tools and frameworks that will be used in the following semester to create new features for the MER field.

Work was also conducted in looking at the current algorithms and current ways to extract features, thus learning how they work so that in the second semester the work of creating new features could begin straight away.

The work of expanding the datasets was also almost partially concluded. Regarding the Static MER dataset, data acquisition was completed and the annotations are almost finalised. As for the MEVD dataset, data acquisition was also completed. However, it is likely that the annotation process will not be finalised during the course of this thesis, given the complexity of the manual annotation process involved.

Second semester

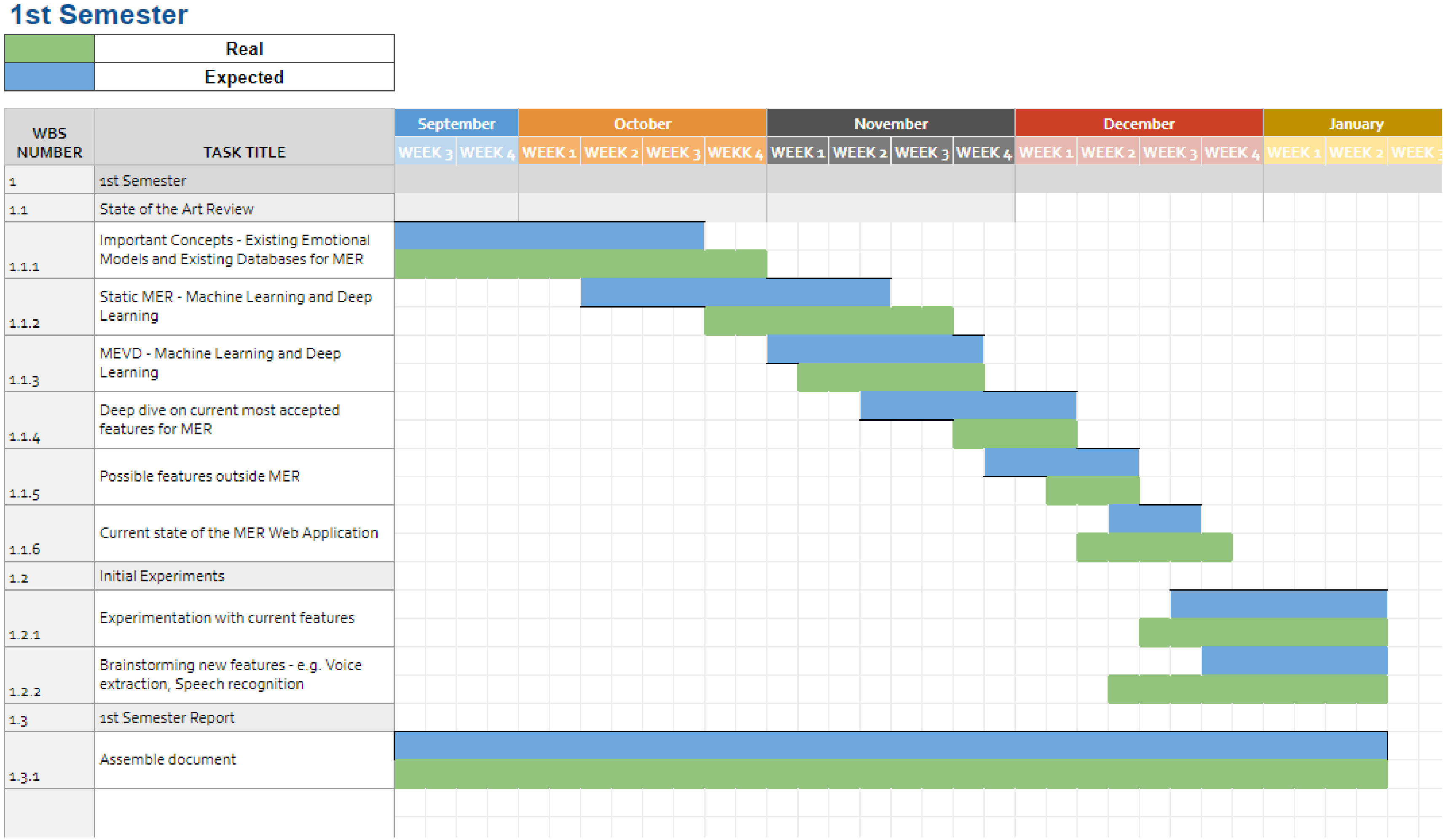
In the second semester, the work regarding the expansion of the datasets was finalized. Furthermore, verifications were done in order to ensure that throughout all the annotations and merging processes that everything had worked out correctly. Some minor rectifications had to be done (e.g. ensuring the right artist and song title were present in the metadata). Meanwhile while the annotation process was being finalized, work was done starting to extract the features from the percussion track and the MIDI files.

It also became clear that the expansion of the MEVD dataset would not be complete during the time-frame of this work, and as such, the objective of creating new features for MEVD was removed from the scope of this work. This also meant that work exploring features related to form and chorus detection approaches was no longer pursued. Nonetheless, the work regarding this topic can be found in Appendix A.

The overall planning was followed, albeit delayed a few months. The magnitude of this work proved a setback. There was also the setback of the removal of the MEVD objective from this work. Furthermore, there was one thing that was unaccounted for in the original planning, which was the time it would take to do hyper-parameter optimization and result gathering. As it was chosen that for each combination of features hyper-parameter optimization would be done, this took several days.

Figures 1.1 and 1.2 respectively represent the Gantt charts for the first and second semester, with the first figure containing both the estimated and the real effort for the first semester and the second figure displaying the same information regarding the second semester.

Overall, the timelines set in the beginning of the semester were accomplished, minor a few setbacks due to the magnitude of this work. However, certain tasks initially proposed in the Gantt chart were removed from the scope of this work, and as such, there are tasks that do not have the real effort taken displayed in the chart.



1st Semester

🟩	Real
🟦	Expected

WBS NUMBER	TASK TITLE
1	1st Semester
1.1	State of the Art Review
1.1.1	Important Concepts - Existing Emotional Models and Existing Databases for MER
1.1.2	Static MER - Machine Learning and Deep Learning
1.1.3	MEVD - Machine Learning and Deep Learning
1.1.4	Deep dive on current most accepted features for MER
1.1.5	Possible features outside MER
1.1.6	Current state of the MER Web Application
1.2	Initial Experiments
1.2.1	Experimentation with current features
1.2.2	Brainstorming new features - e.g. Voice extraction, Speech recognition
1.3	1st Semester Report
1.3.1	Assemble document

Figure 1.1: Estimated and real effort for the first semester.

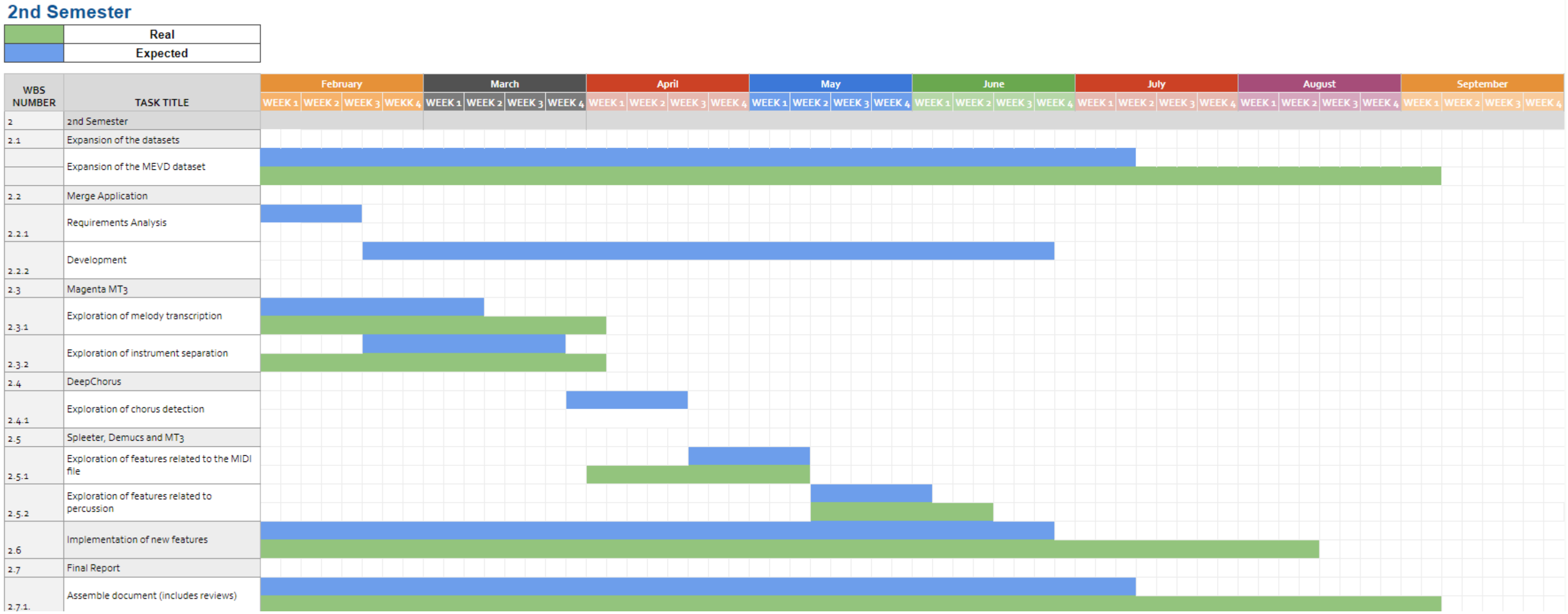


Figure 1.2: Estimated and real effort for the second semester.

1.5 Outline

The following chapter presents an introduction to important concepts for this work, such as the concept of emotion (Section 2.1) and musical dimensions (Section 2.2).

In the next chapter, a review of the state of the art of the MER field is presented, particularly the most common used datasets for MER (Section 3.1), followed by a review on the most commonly used features (Section 3.2). Furthermore, there is a section regarding the proposal of new features for this work (Section 3.3). Finally, the final two sections of the following chapter give a brief overview on Automatic Music Transcription (AMT) frameworks (Section 3.5) as well as music source separation frameworks (Section 3.6).

In Chapter 4, an overview of the methodology for the dataset expansion used in this work is provided as well as the information regarding the final datasets.

Chapter 5 provides an explanation of used methodology in this work, from the preliminary steps, from the first steps of audio pre-processing to the final classification and evaluation metrics.

Finally, in Chapter 6, a comparison with the 4QAED dataset is provided (Section 6.1) as well as testing the novel datasets with combinations of the new set of features (Section 6.2).

Appendix A provides information regarding the state of the art MEVD approaches, as well as information regarding possible new frameworks that might help in aiding to solve the MEVD task. This is added as an appendix due to being removed as one of the main focuses of this work.

This page is intentionally left blank.

Chapter 2

Background concepts

In this chapter, an explanation of important background concepts for MER is provided. This explanation pertains to what is emotion, different types of emotion, models to classify emotion, as well as introducing important musical concepts, such as melody, percussion, amongst others.

2.1 Emotion

Grasping the concept of emotion has always been a challenge for humans. It has been a concept debated for hundreds of years, and a consensus on a concrete definition has not been reached yet (Dixon, 2012). In the following sections of this work there will be a presentation related to emotion, such as the various types of emotions as well as the various models that exist for classifying emotion.

2.1.1 Defining emotion

One way to try and understand the concept of emotion is by looking at the origin of the word. Etymologically [1], emotion originates from the french word *émotion* (to stir up) with this word coming from the Latin word *emovere* (move out, remove, agitate).

Emovere is composed by the form "*ex*" (out of, from) and "*movere*" (to move). Emotion first entered the English vocabulary in the 16th century.

Ever since the article "What is an Emotion" by Wiliam James was published in 1884 (James, 1884), many attempts have been made at providing a concrete definition for emotion, however, they all have the same problem. Emotion from a literature standpoint is a very ambiguous concept, as Thomas Dixon stated: "the problem is not that the term 'emotion' has no clear meaning, but that it has many meanings". (Dixon, 2012)

[1] https://www.etymonline.com/word/emotion

In the Merriam-Webster dictionary [2], emotion is defined as:

1. - **a:** a conscious mental reaction (such as anger or fear) subjectively experienced as strong feeling usually directed toward a specific object and typically accompanied by physiological and behavioral changes in the body
   - **b:** a state of feeling
   - **c:** the affective aspect of consciousness
2. - **a** excitement
   - **b:** (obsolete): disturbance

In the next sections, the different types of emotions are discussed as well the types of models used in MER literature to categorize emotion.

2.1.2 Different types of emotion

In the MER field, emotions are usually divided into three categories: expressed, perceived and felt (also called induced) emotions: (Gabrielsson, 2001)

- Expressed emotions: Refers to the emotion that the composer is trying to convey in the piece.
- Perceived emotions: Refers to the emotion the listener apprehends from listening to the piece, which can be, and often is, different than the expressed emotion.
- Felt/Induced emotions: As the name suggests, refers to the emotion that is felt by the listener while listening to a specific piece.

Most of the time, the emotions expressed by a composer in the piece are also the emotions that the listener identifies, however this is not the case for the induced emotions. Since the induced emotions depend on the person's characteristics and personality, they might change from person to person and even the same person in different circumstances of life might feel different emotions from the same piece (Yang and Chen, 2012).

One example of this is described as the "paradox of negative emotion" whereby "music described in terms of negative emotions (e.g. sadness, grief, despair) is often judged as enjoyable." (Pannese et al., 2016)

This also raises problems in terms of conducting studies in the MER field, as not all studies deal with the same kind of emotion, and with induced emotions being more subjective when compared to perceived emotions (Pannese et al., 2016), this can lead to the creation of poor datasets. This requires that the people involved

[2] https://www.merriam-webster.com/dictionary/emotion

in these studies are instructed to focus on perceived emotion rather than induced emotion.

As such, most of the work on the MER field is focused on perceived emotion, as that is the emotion that has the most agreement between the various listeners. However, there is still subjectivity, so it is paramount that measures are taken in the creation of the datasets in order to minimize this subjectivity (Section 3.1).

2.1.3 Models for emotion classification

Throughout the years, many models and new ways of classifying emotions have been proposed. The models that are used to classify emotion can be split into two categories: discrete models, in which emotions are represented by words or groups of words, and dimensional models, in which emotions are classified in a discrete or continuous multi-dimensional space.

Categorical models

The models in this category function on the premise that emotions are distinct from one another and can be separated into categories, a concept first introduced by (Ekman, 1992).

Ekman's Model

This model classifies emotions into six categories: anger, disgust, fear, happiness, sadness and surprise. These are considered the basic emotions from which all the others derive from: "other non-basic emotions are combinations of the basic emotions" (Ekman, 1992).

However, this model was not developed with musical studies in mind, but with the intent of classifying different facial expressions. Thus, it is not as important in the MER field as it is missing some of the emotions required for the MER studies (e.g. calm) while having some that are not applicable (e.g. disgust, surprise).

Hevner's Adjective Circle

(Hevner, 1936) developed a method of grouping similar emotions into various groups. In total there are 67 adjectives, split into 8 groups of varying sizes displayed in a circular pattern, with the adjectives in the same group sharing close meanings. Furthermore, groups next to each other have a higher similarity in terms of emotion. Figure 2.1 showcases this circle.

However, this approach has a few problems, mainly the lack of balance in the number of adjectives of each group and also the fact that this model was proposed with classical music in mind, which may lead to some of the adjectives no longer being associated with current music.

Over the years there have been a few proposed adjustments to the Hevner's adjective circle, such as (Farnsworth, 1954) and (Schubert, 2003).

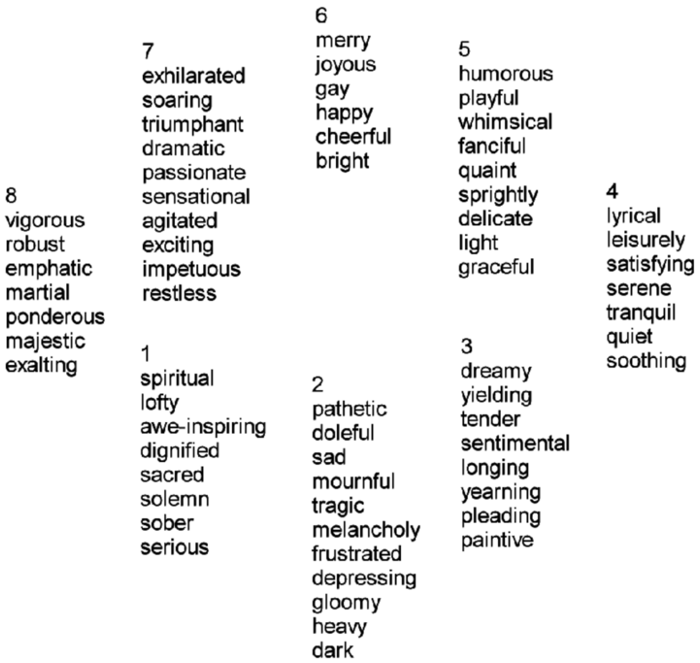


Figure 2.1: Hevner's Adjective Circle (Hevner, 1936).

Dimensional models

In dimensional models, emotions are mapped onto a multi-dimensional space, usually with 2 dimensions. This also enable judging the similarity between two audio clips based on their distance in the plane.

The most famous model of this type, and one of the most influential in the MER field (Laurier, 2011), is the model proposed by (Russell, 1980), known as the Russell Circumplex Model of Emotion. It has received support by several studies. (Posner et al., 2005)

Russell's Circumplex Model of Emotion

Russell proposed a two-dimensional model, in which the two proposed dimensions are: valence, also known as pleasure-displeasure, and arousal. Valence represents how pleasurous an emotion is, and arousal represents the intensity of the emotion.

The result, represented in Figure 2.2, is a two-dimensional plane (arousal-valence) where the X-axis represents valence and the Y-axis represents arousal. The resulting four quadrants can be roughly defined as:

1. **Q1** - Positive valence and positive arousal, referring to happy and energetic emotions, such as happiness and delight.

2. **Q2** - Negative valence and positive arousal, referring to frantic and energetic emotions, such as anxiety or fear.

3. **Q3** - Negative valence and negative arousal, referring to melancholic and sad emotions, such as depression or sadness.

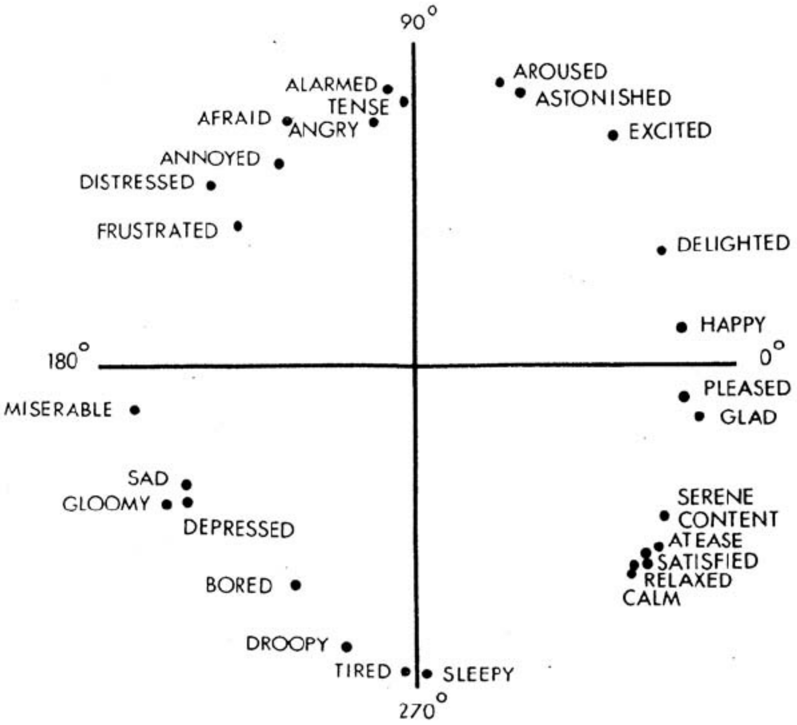


Figure 2.2: Russell's Circumplex Model of Emotion (Russell, 1980).

4. **Q4** - Positive valence and negative arousal, referring to calm and positive emotions, such as serenity.

One characteristic of this model, also showcased in Figure 2.2, is that emotions are placed away from the center. This is because emotions close to the center are often considered ambiguous and the emotions are not easily identifiable. The description provided will be brief, with a more thorough analysis of the musical dimensions being provided in (Benward and Saker, 2008; Laitz, 2007)).

2.2 Musical dimensions

In order to help understand how music and emotion are related, it is important to have a better understanding of the fundamental music dimensions and how they are organized, as described in this section.

Musical dimensions are organized mostly into eighth categories (Owen, 2000), with each representing a concept. In this section a brief description of each of the category will be provided, based on the current literature. These eight categories are: melody, harmony, rhythm, dynamics, tone color (or timbre), expressivity, musical texture and musical form.

It is noteworthy that the organization of these dimensions is not strict, meaning that many of the musical features are connected and can interact and touch other dimensions, and as such it can be hard to pinpoint to which musical category a certain feature belongs.

2.2.1 Melody

Melody refers to the horizontal progression of musical tones or pitches, perceived as a singular musical entity. This idea is also supported by Johann Philipp Kirnberger, a student of Bach, who regarded melody as the essence and ultimate purpose of music. Melody can be understood through multiple attributes, summarized in Table 2.1.

Name	Description
Melodic Arrangement	How melodies are positioned, e.g., in sequence or counter-melodies.
Melodic Movement and Contour	Patterns of notes in terms of pitch direction and the shapes they create.
Pitch	Perception of sound as "higher" or "lower" related to frequency.
Pitch Range	The range from the highest to lowest notes in a melody.
Register	Perceived "height" of a sound; classified as high, middle, or low.
Melodic Features	Features added to enrich the melody or connect it to others.

Table 2.1: Summary of the melodic attributes of music.

2.2.2 Harmony

Harmony is the "vertical" dimension of music, contrasting with melody's horizontal aspect. It concerns the combination of different pitches (or notes) to create chords. The term "harmony" has its roots in the Greek language, signifying "agreement or concord of sounds" or a "combination of tones pleasing to the ear" [3].

Analyzing the harmony of a song involves the study of chords, made of several notes played simultaneously, and of chord progressions, which are the sequences of chords arranged together, as illustrated in Figure 2.3.

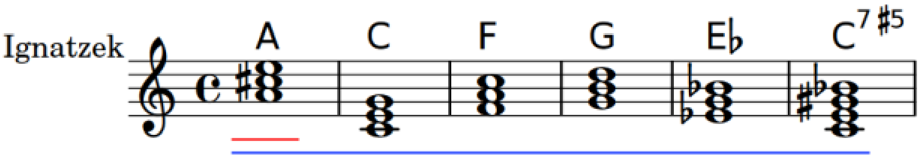


Figure 2.3: A 3-note chord in red and a chord progression of 6 chords in blue. Chord names are displayed at the top using the Jazz notation system proposed by Klaus Ignatzek.

Table 2.2 provides an overview of the harmonic characteristics of music.

[3] https://www.etymonline.com/word/harmony

Name	Description
Harmonic Rhythm or Harmonic Tempo	Rate of chord changes relative to the note rate.
Harmonic Progression	Sequence of musical chords guiding the melody's direction.
Modulation	Process of altering the key center in a piece.
Harmonic Perception	Relative roughness or smoothness of a sound; consonant sounds are pleasing while dissonant sounds are not.

Table 2.2: Summary of the harmonic characteristics of music.

2.2.3 Rhythm

Rhythm can be described as the musical element representing "time", involving patterns of elongated and abbreviated sounds and periods of silence. Rhythm embodies an organized sequence of contrasting elements. Multiple facets shape rhythm, encompassing elements like tempo, duration, and meter.

Table 2.3 provides an overview of the rhythm attributes of music.

Name	Description
Rhythm Types	Can be simple or complex, regular or irregular.
Note Values and Rests	Denotes note length or duration (e.g., semibreve, quaver, semiquaver).
Rhythmic Devices	Devices shaping the music, often hinting at its genre (e.g., riff, repetition, syncopation).
Rhythmic Layers	Grouping of instruments in a piece (e.g., instrumental sets and vocals).
Duration	Length a sound or silence persists.
Beat	Consistent pulse in music; can be strong or weak.
Metre	Organization of beats in an ordered sequence.
Tempo	Speed of the beat; can vary from fast to slow or change.

Table 2.3: Summary of the rhythmic attributes.

2.2.4 Dynamics

Dynamics in music refer to the range between the loudest and softest notes in a composition. All elements linked to the relative volume or quietness in music come under the overarching category of dynamics. Notable aspects of dynamics encompass the varying degrees of loudness and softness, the contrast between loudness levels, and the emphasis or accent placed on particular notes.

Dynamics annotations in musical scores are invariably relative. For instance, they might suggest a section should be played louder, but they do not stipulate a precise loudness level. Musicians leverage dynamic changes to maintain interest and engage their audience.

Table 2.4 provides an overview of the main attributes tied to dynamics in music.

Name	Description
Dynamic Levels	Loudness levels in a piece (e.g., forte, piano).
Accents and Changes in Dynamic Levels	Gradual shifts in dynamics (e.g., crescendo). Emphasis on particular notes (e.g., sforzando).

Table 2.4: Summary of the attributes related with dynamics.

2.2.5 Tone Color or Timbre

Tone color, often termed as "timbre", signifies the perceived quality or attributes of a sound, such as a musical note. It is the distinctive tone color of a sound that lets a listener discern between various sound sources. For instance, one could differentiate between two instruments playing similar notes, separate one human voice from another, or even distinguish between instruments of the same category, like telling apart a trumpet from a saxophone.

Table 2.4 provides a breakdown of the main primary attributes influencing tone color in music.

Name	Description
Instrument Materials	Influence of material and shape on sound (e.g., wood, metal, vocal).
Playing Methods	Technique to elicit sound from the instrument (e.g., pluck, hit, blow).
Instruments' and Voices' Types	Categorization of the sound source (e.g., strings or percussion; soprano or tenor).
Combinations and Types of Sounds	Nature of instruments (acoustic or electronic) and their grouping in ensembles (e.g., bands, orchestras, Jazz trio).

Table 2.5: Summary of the elements influencing tone color.

Drawing a parallel to visual arts, sounds are said to have a palette of tone colors. Each musical instrument possesses a unique tone color on this spectrum. Composers utilize and amalgamate these tone colors, forging contrasts and novel combinations, in a manner akin to how an artist paints a scene.

2.2.6 Expressivity

Expressive techniques pertain to the methods a performer employs to play a musical composition, especially the techniques used to articulate a unique style or a particular interpretation of a style. Expressive techniques, combined with dynamics, provide “soul” or emotional depth to music.

Over time, various expressive techniques have emerged. These could be techniques linked with instruments, vocals, ornamentations, pace alterations, or specific methods that tie successive notes. Coupled with other musical elements, these techniques substantially influence musical styles, from classical Western compositions to global genres like Indian ragas, African zouk, or Portuguese fado. Each style has its hallmark expressive techniques and instruments.

Table 2.6 provides a summary of the primary attributes of expressive techniques:

Name	**Description**
Tempo (changes)	Speed of the music and its alterations to influence expressive quality.
Stylistic Indications	Terms guiding the manner of performance (e.g., legato, rubato).
Articulation	Manner of playing specific segments or notes (e.g., staccato, slur).
Ornamentation	Embellishing notes with special features (e.g., glissando, trills).
Instrumental, Vocal, and Electronic Techniques	Methods to produce distinct sounds for a style (e.g., vibrato) or electronic modifications (e.g., vocoders).

Table 2.6: List of expressive techniques attributes.

2.2.7 Musical Texture

Musical texture pertains to the manner in which rhythmic, melodic, and harmonic elements produced by musical instruments and voices intermingle in a musical composition. It primarily deals with how musical lines or layers (which could be one or several instruments serving a similar function) in a song are combined and related. A single musical layer can comprise several performers adhering to the same melody.

The texture of music can be classified based on (Benward and Saker, 2008, p. 146):

1. **Density** - Ranging from thin (as in a song by a solo guitar) to thick (like in an orchestral piece with multiple lines of melody, rhythm, and harmony).

2. **Range** - The distance between the lowest and highest tones in the composition.

The musical texture can also be defined based on the count and relationship of layers, with monophonic, homophonic, and polyphonic being common classifications. There's often a correlation between different musical elements, like an increase in layers (leading to a thicker texture) typically being accompanied by a rise in dynamics.

Table 2.7 summarizes the attributes of musical texture.

Name	**Description**
Number of Layers, Density, and Range	Pertains to the count of musical lines, their density (thin or thick), and their range (narrow to wide).
Texture Types	Different layer combinations, including monophonic (single layer), homophonic (multiple layers with a dominant melody), and polyphonic (several independent melodies).

Table 2.7: Summary of the musical texture attributes.

2.2.8 Musical Form

Musical form, also known as musical structure, describes the layout or structure of a composition, typically divided into various sections (Brandt et al., 2011). These sections in a musical piece are often distinguished by changes in rhythm and texture. A uniform rhythm and texture make the listener perceive the music as a singular section. However, noticeable changes in these elements mark boundaries, transitioning the piece into a new section.

Individual song elements each have unique functions and placements in a composition. Common sections include verses (varying lyrics), chorus (repeated melody and lyrics), introductions, bridges (linking verses and chorus), and outros. In genres like pop/rock or blues, solo sections featuring melodic lines (often improvised) are prevalent. Figure 2.4 showcases the typical pop song structure.

Table 2.8 summarizes the attributes of musical form.

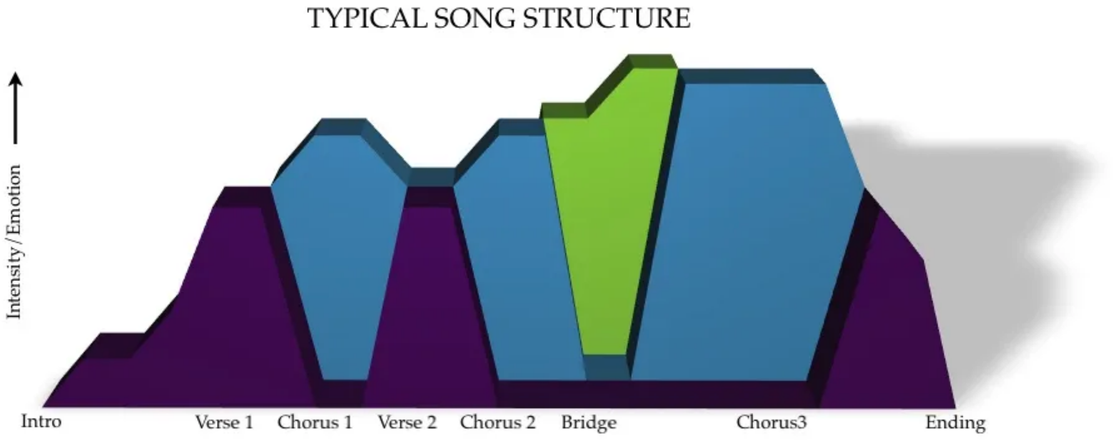


Figure 2.4: Typical pop song structure.

Name	**Description**
Song Elements	Different sections that constitute a musical piece, such as the introduction, verse, chorus, and bridge.
Organization Levels	Rough categorization of musical form into levels like passage (related to musical phrases), piece (pertaining to the whole composition), and cycle (for larger compositions).
Basic Musical Forms	Combining sections results in various forms, e.g., through-composed (no repetition), strophic (verse-repeating), and binary (two contrasting sections).

Table 2.8: Summary of features defining musical form.

This page is intentionally left blank.

Chapter 3

State of the art

This chapter presents an overview of important state of the art concepts, such as review both on the research already performed in this area and the tools used. It also presents an overview of the most used features in MER, as well as the datasets used.

3.1 Datasets used for MER

As previously mentioned, in the MER field there is a severe lack of quality and sizable datasets due to the difficulty in creating them. Thus, it is very hard to compare different approaches and studies due to most of them using different datasets.

The datasets are difficult to create because every audio snippet (in the case of Static MER) or song (in the case of MEVD) needs to be manually annotated. This is a very time consuming task, especially for MEVD. Furthermore, in the case of Static MER snippets of songs are used, typically with only a few dozens of seconds (e.g: 30 seconds), but in the case of MEVD complete songs are annotated. First the whole song has to be split into segments, and only then can those segments be annotated like in the Static MER approach. This is also a very labor-intensive task as protocols needs to be considered as to what constitutes a new segment, if segments such as verse/chorus are also separated, amongst others. Moreover, due to emotion subjectivity leading to inter-listener variability, the snippet cannot be annotated by a single person. Thus, in order to create good quality datasets, snippets are annotated by various annotators and then only the snippets that have high agreement between annotators (for example, being placed in the same quadrant), will be kept and the others will be removed. Finally, it is also important to have diversity in various characteristics of the dataset, such as having a good genre distribution, different artists, songs from different years/eras, amongst many others.

Moreover it is also important to deal with factors relating to emotion subjectivity. Thus it is necessary that in the creation of the datasets a few preemptive steps are taken to minimize this problem. These include selecting songs not in the center of the Russell plane and validating the AllMusic annotations, amongst others.

Due to all these factors, it is not uncommon to have a big set of songs for annotation, and after all the annotation process described above, finishing with a sub-set of the original set with only a small percentage of the songs being kept.

Moreover, some studies also focus on a specific type or genre of music (e.g. only classical music) which makes using those approaches in other scenarios unrealistic. In addition, there is also the problem that, in some of the works, private datasets are used (e.g. (Laurier, 2011)), which makes replicating and verifying the results and approaches nearly impossible.

There is also the problem aforementioned, related to the size of the datasets employed, with some of the works using very small datasets. For example, in the work (Malheiro, 2017), only 133 songs were used. Moreover, there is a problem of the datasets used being low quality, having issues such as low agreement in the annotations, as showcased in (Panda et al., 2020a). One final problem with MER datasets is that, due to working with music fields, there are a lot of problems with copyright infringement. Thus, some datasets will not provide the samples used nor how to obtain them.

In the following sections, a description of the main datasets used for Static MER and MEVD will be provided.

MIREX

This dataset, proposed to the Music Information Retrieval eXchange (MIREX) in 2007 is a private dataset that is used by MIREX to compare various music emotion recognition algorithms that are submitted by researcher for Static MER approaches. With this being a private dataset, it cannot be used by researchers and is only available to the MIREX leaders of this specific task.

It is comprised by 600 audio clips with 30 seconds in length in 22050 HZ mono WAV format, manually annotated. It is also annotated in 5 clusters of 29 adjectives.

Over the years there have been a few issues raised with the emotion taxonomy used, mainly the fact that there is no support from psychology studies that back it up, and the fact that there is acoustic and semantic between the different clusters.

CAL500exp

This dataset, proposed in (Wang et al., 2014), is an expansion of the successful CAL500 dataset. It is comprised of 500 original music clips split into segments ranging from 3 to 16 seconds in length. In order to create these segments, an algorithm was used to divide the audio clips into segments based on their acoustic content. These segments then are given emotion tags. Finally, 11 music experts refined the tags given to each segment, with having the possibility of deleting tags or adding new ones.

As is the case with the original CAL500 dataset, the audio clips are not publicly available, however for CAL500exp they are available upon request to the authors of the dataset. Furthermore, by giving the experts a set of tags as a baseline annotations for each segment, it is possible that the final annotations might have been influenced.

Million Song Dataset

This dataset, proposed in (Bertin-Mahieux et al., 2011), was created with the purpose of solving the problem of small datasets in the Music Information Retrieval (MIR) field. It was created with data from many sources, but mostly from The Echo Nest [1].

It is comprised of a million songs' features and metadata, however, the tags for each song are a byproduct of Last.fm [2] from where the songs were extracted, which may not reflect in quality annotations.

There are also more problems with this dataset, such as it not containing the 30 second audio samples. This means that any attempt to get these samples does not provide any guarantee that the sample extracted will provide the same features as the one in the dataset, as there is no way to make sure the samples are the same. Furthermore, there is also no guarantee that the sample extracted is the same sample used when the listener was annotating it.

For example, one listener can annotate as song with the tag "love", however there is no way to tell if they love the song or if they are annotating that the song is about love.

Bi-Modal

This dataset, proposed in (Malheiro et al., 2016), is a bi-modal dataset with both lyrics and the corresponding audio. This allows for both studies in Static MER with only the audio in mind, only lyrics, or both. To build this dataset 200 songs (lyrics and corresponding 30-second audio clips) were selected with the following criteria: "Several musical genres and eras; Songs distributed uniformly by the 4 quadrants of the Russell emotion model". (Malheiro et al., 2016).

The annotations were done by a total of 39 annotators with different backgrounds, with every person classifying either the audio of a lyric of a particular song. The arousal and valence of each song was obtained by averaging the values of the annotations of every subject, with songs having large standard deviation values being discarded. This dataset also showcases strong agreement between annotators, however, even though it is well constructed, it is very small in size.

In the end, three datasets were created. A lyrical dataset, containing 180 lyrics, an audio dataset, containing 162 audio clips, and a bi-modal, containing 133 song with both audio (30-second snippets) and lyrics.

DEAM

This dataset, proposed in (Aljanaki et al., 2017), consists of 1802 audio files with no royalty, with these files being split into 58 full songs and 1744 snippets of 45

[1] The Echo Nest is a music intelligence and data platform for developers and media companies, it began as a research spin-off from the MIT Media Lab to understand the audio and textual content of recorded music and has since been acquired by Spotify.

[2] Last.fm is an online music service platform. Url: https://www.last.fm/home

seconds with various genres. It is comprised of both static and dynamic annotations in the Russell's model, meaning it can be used both for MEVD and for Static MER. In the dynamic annotations, there is an emotion label every 0.5 seconds. This is a problem as it can possible lead to situations where that specific sample does not represent the emotional sentiment of the song or that specific sample does not have any sound.

Furthermore, there is also the issue of clips with noise (such as clapping, people speaking). The dataset also has a very low agreement rate between annotators of 47% (Sá, 2021), which showcases even further the problems in this dataset.

4QAED

This dataset, proposed in (Panda, 2019), is a dataset comprising 900 music clips. As the name suggests, 4 Quadrant Audio Emotion Dataset, these 900 music clips (30-seconds in length), were mapped onto the Russell's model. The clips were gathered from the AllMusic [3] API with their respective mood tags, and then those mood tags were mapped onto quadrants using Warriner's list of arousal and valence values (Warriner et al., 2013).

Post-processing was then done, such as discarding low quality clips (e.g. clips with clapping, stand up comedy).

The result was a dataset of 900 songs, balanced with 225 songs per quadrant, with considerations like maximizing genre distribution also being taken into account, available for studies regarding Static MER.

In result of work of our group, this dataset has since been revised to be shortened to 893 songs, with the removing of a few duplicate songs and invalid songs. However, as of writing, this updated version has not yet been published.

The 900 song dataset as well as the metadata is available for any researcher to test their emotion detection algorithms.

All of these datasets share similar problems: they are either private, small or were created based on poor sources (e.g. Million Song Dataset). This is why there is a need for creating both high quality and large size datasets for Static MER and MEVD, with this being one of the main objectives of this work.

[3] AllMusic is a music platform that has metadata for songs as well as professional annotations for the songs. https://www.allmusic.com

Databases Review					
Name	Type	Emotion Taxonomy	Audio Duration	Size	Notes/Observations
MIREX	Static MER	5 clusters of 29 adjectives	30 seconds	600 audio clips	No psychology study support for the emotion taxonomy.
CAL500exp	MEVD	Based on the tags of Cal500, totaling 67 tags	3 to 16 seconds audio clips	500 songs divided into segments	
Million Song Dataset	Static MER	User submitted emotional tags	30 seconds song clips	1 million audio clips	Annotations can have poor quality due to ambiguous and uncontrolled annotations.

Table 3.1: Review of the datasets used for Static MER and MEVD.

Databases Review					
Name	Type	Emotion Taxonomy	Audio Duration	Size	Notes/Observations
Bi-Modal	Static MER	Russell's A/V Model	30 seconds	162 audio clips and 133 bi-modal	The size of this dataset is very small even though the annotations were well conducted.
DEAM	Static MER and MEVD	Russell's A/V Model	Static: 45 seconds Dynamic: Full songs	1744 audio clips and 58 full songs	Low agreement rate between annotators, clips with noise and issues where specific sample does not represent the dominant emotion in the song.
4QAED	Static MER	Russell's A/V Model	30 seconds	900 audio clips, later revised to 893	Balanced between quadrants.

Table 3.2: Continuation of the review of the datasets used for Static MER and MEVD.

3.2 Feature engineering in MER

This section provides an overview on the most commonly used features, as well as a review on the features that represent each of the eight musical dimensions related to this work and finally a proposal of new features for this work.

3.2.1 Most commonly used features

In (Panda et al., 2020a) a thorough review of the current state of feature engineering on the MER field was conducted. This work showcased that of the eight musical dimensions (melody, harmony, rhythm, dynamics, tone color, expressivity, texture and form) several are still underrepresented in the state of the art works, while others are very dominant. Moreover, this work also proposed some methods and ideas for future feature engineering, with some of those ideas being the basis for this work.

Table 3.3 summarizes the number of features per musical dimension. As can be observed, there are musical dimensions that are severely underrepresented. Moreover, there has been work that showcases that the underrepresented musical dimensions, particularly texture, are useful specially in identifying songs pertaining to the first and second quadrant. (Panda et al., 2018) Thus, it is important to create new features that represent these musical dimensions.

Musical Dimension	Number of Features	Percentage of total
Melody	9	10.60%
Harmony	10	11.80%
Rhythm	16	18.80%
Dynamics	12	14.10%
Tone Color	25	29.40%
Expressivity	6	7.10%
Texture	3	3.50%
Form	4	4.70%
Total	**85**	**100**%

Table 3.3: Number of features per musical dimension (Panda et al., 2020a).

The following sections provide an overview of the audio features that have been proposed throughout the years for each of the aforementioned eight musical dimensions. Most of these features are extracted from sequential smaller snippets of the audio tracks (commonly referred as windows), with the result being a series of data. These features extracted from the windows are then processed using statistics such as mean, standard deviation, amongst other techniques, and are then used on the machine learning algorithms.

3.2.2 Melody features

Melody can be defined as a horizontal succession of pitches, which is perceived by listeners as a single musical line. There are several features that have been proposed in literature:

1. **Pitch** - Pitch represents the perceived fundamental frequency of a sound (F0). Along with loudness and timbre, it constitutes one of the three major auditory attribute of sound. As an audio feature, pitch usually refers to the fundamental frequency of a monophonic signal (a signal with a single melodic line). Pitch detection algorithms have as an output a sequence of F0s values through time. Several frameworks implement different ways of calculating pitch, such as the YIN algorithm (Cheveigné and Kawahara, 2002) or the algorithm proposed in (Camacho and Harris, 2008).
2. **Pitch Salience** - Pitch salience is a complex concept but it can be explained as how noticeable is the pitch in a sound. Pure tones have an average pitch salience value close to 0 while sounds with various harmonics in the spectrum have higher values of pitch salience.
3. **Predominant Melody F0** - Finding the fundamental frequency of the predominant melody in both monophonic and polyphonic signal is still an open research program, however there have been already a few approaches, such as the MELODIA algorithm (Salamon and Gómez, 2012), an approach proposed by Karin Dressler (Dressler, 2016), and more recently, deep learning approaches, such as (M et al., 2022).
4. **Pitch content** - (Tzanetakis, 2002) proposed a set of features extracted from pitch histograms folded and unfolded (in the folded histograms all the notes are mapped onto a single octave) to describe pitch information:
   (a) **FA0** - Amplitude of the maximum peak of the folded histogram.
   (b) **UP0** - Period of the maximum peak of the unfolded histogram.
   (c) **IP01** - Pitch interval between the two most prominent peaks of the folded histogram.
   (d) **SUM** - The overall sum of the histogram.
5. **MIDI Note Number (MNN) Statistics** - In (Panda et al., 2018), the sequence of predominant melody F0 values were quantised into a sequence of MIDI notes. Based on the MIDI note number of each note, 6 statistics were proposed: textbfMIDIMean, i.e. the average MIDI note number of all the notes, MIDIstd (standard), MIDIskew (skewnewss), MIDIkurt (kurtosis), MIDImax (maximum) and MIDImin (minimum).

These features are extracted from the melody transcription of the original audio waveform. Several methods were employed to estimate the predominant F0 values and pitch saliences, as well as methods to quantise F0 sequence into MIDI notes. Figure 3.1 showcases an example of this features on a specific audio track.

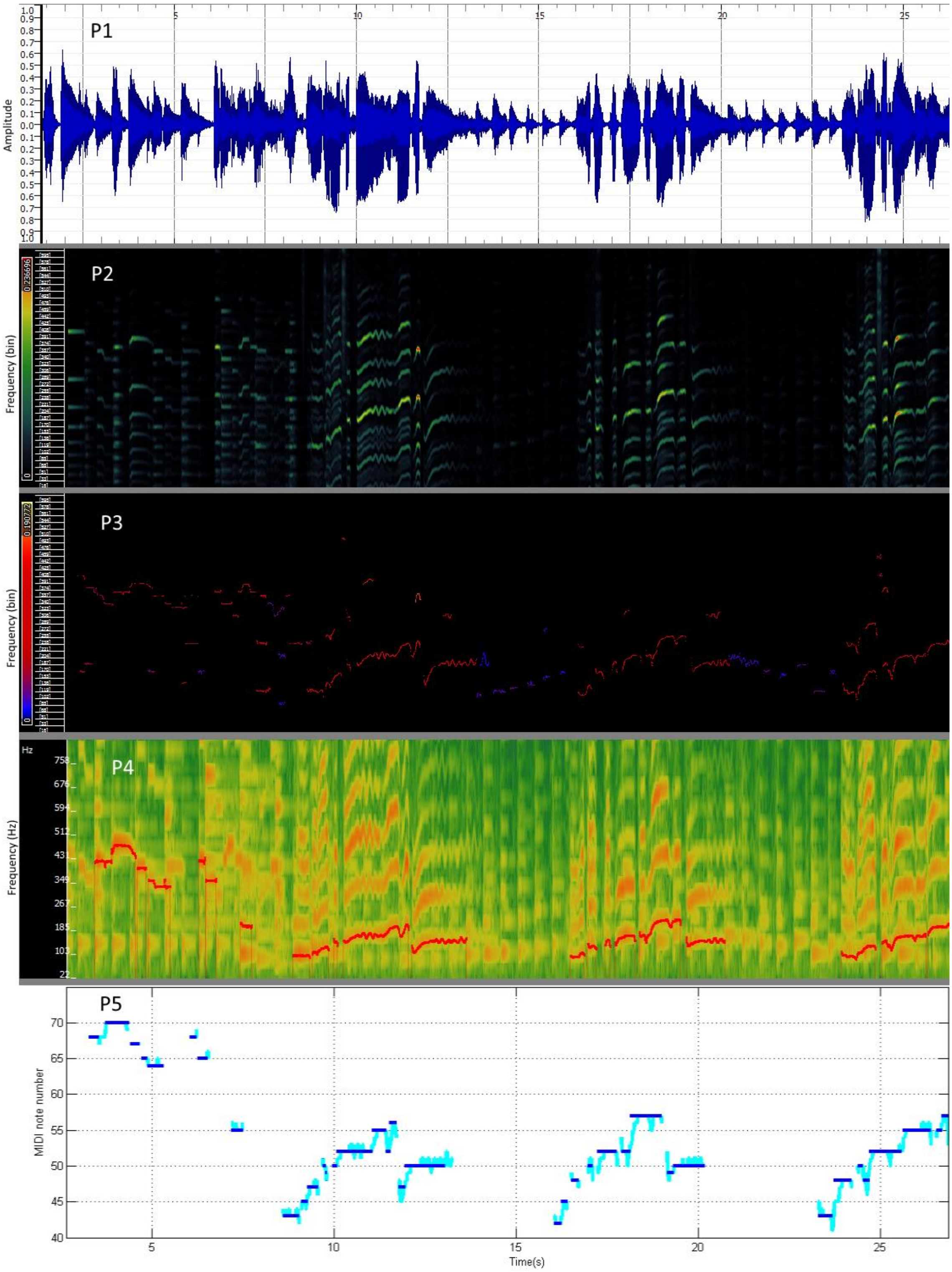


Figure 3.1: Excerpt from "S'posin" by Frank Sinatra, transformed from the audio signal to midi notes using the Melodia plug-in (P1-P4) (Salamon and Gómez, 2012) and (Paiva et al., 2006) work (P5). P1: audio waveform, P2: pitch salience function, P3: pitch contours, P4: extracted melody (in red) with the spectrogram as background, P5: midi notes from (Panda, 2019).

6. **Register distribution** - This set of features proposed in (Panda et al., 2018) represents the distribution of the notes of the predominant melody across the different pitch ranges: Soprano (C4-C6), Mezzo-soprano (A3-A5), Contralto (F3-F5), Tenor (B2-A4), Baritone (G2-F4) and Bass (E2-E4). The resulting metrics are the percentage of the MIDI note numbers in the melody of each of the pitch ranges, as well as the register distribution per second, with this being calculated as the ratio of the sum of the duration of notes from a specific range (e.g. Soprano) to the total duration of all the notes.

7. **Note Smoothness (NS) statistics** - (Panda et al., 2018) proposed also a note smoothness feature that indicates how close consecutive notes are. To calculate this, the difference between the MIDI numbers of consecutive notes is calculated. The 6 statistics aforementioned are also calculated.

8. **Ratios of Pitch Transitions** - Using the MIDI note number, a sequence of the transitions to higher, lower and equal notes is created (Panda et al., 2018). This sequence is summarized by several metrics such as:

    (a) Transitions to Higher Pitch Notes Ratio

    (b) Transitions to Lower Pitch Notes Ratio

    (c) Transitions to Equal Pitch Notes Ratio

    In each of these metrics, the ratio of the number of specific transitions (i.e., from higher to lower) to the total number of transitions is computed.

3.2.3 Harmony features

If melody can be considered the horizontal part of the music, harmony refers to the vertical part of the music, i.e. the sound produced by the combination of various pitches in all the cords.

1. **Inharmonicity** - This feature is based on the number of partials that are not multiples of the fundamental frequency F0. There are several ways to calculate this feature, such as the one found in the timbre toolbox (Peeters et al., 2011). Another implementation by MIR ToolBox calculated inharmonicity by measuring the amount of energy outside the ideal harmonic series, which assumes that there is only one fundamental frequency (Lartillot, 2018).

2. **Chromagram** - The chromagram is a feature used to estimate the energy distribution between pitch classes. It consists of a vector with 12 dimensions, one for each of the 12 semitone pitch classes (A to G#). The respective intensity of each class is calculated based on spectral peaks of the waveform.

3. **Chord Sequence** - With extracting chords from an audio file being a complex task, there are not yet any robust solutions to this problem. There have been experimental methods proposed based on pitch class profiles (Gómez, 2006).

4. **Key Strength** - Key strength is a value between either 0 to 1 or -1 to -1 that represents for each key the strength of the possibility that the key is the key of the given song. The algorithm to calculate this feature is based on the cross-correlation of the chromagram (Gómez, 2006).

5. **Key and Key Clarity** - These two features provide an estimation of the of the tonal centre positions and their respective clarity. This is achieved by peak picking in the key strength curve. The best keys are calculated by getting the peak abscissa value, and the key clarity is the key strength associated with the best keys (Lartillot, 2018).

6. **Tonal Centroid Vector (6 dimensions)** - In MIR ToolBox, the tonal centroid vector is represented by a vector with 6 dimensions corresponding to a projection of the chords along a circle of fifths, minor third and major thirds (Harte et al., 2006). This is based on the Harmonic Network of Tonnetz, which is a planar representation of the pitch relations. In this representation, pitch classes that have closer harmonic relations such as fifths, major and minor thirds, have smaller Euclidean distances on this plane. The algorithm is able to detect harmonic changes by calculating the Euclidean distance between consecutive analysis frames of the tonal centroid vectors.

7. **Harmonic Change Detection Function** - (Harte et al., 2006) proposed a method for detecting changes in the harmonic content of music audio signals. This feature can be interpreted as the flux of the tonal centroid, calculated as the distance between the harmonic regions of consecutive frames (Harte et al., 2006).

8. **Sharpness** - Sharpness is a feature that measures on a subjective scale that ranges from dull to sharp how sharp an audio signal is.

9. **Modality** - There are several algorithms that attempt to estimate modality, i.e. major vs minor (Lartillot, 2018). The typical strategies employed to estimate the strength of each key are:

    (a) The difference between the strength of the strongest major and minor keys

    (b) The sum of all the differences between the strength of each minor/major key pair.

3.2.4 Rhythm features

There have been several works that address the problem of rhythm analysis, leading to the many features being proposed, that will be explained in the following list in more detail. In (Panda et al., 2018), an extra set of features was proposed based on the MIDI notes, with those features being: note duration statistics, note duration distribution and ratio of note duration transitions. The aforementioned features can be described as:

1. **Beat Spectrum** - Beat spectrum was proposed as a way to measure the acoustic self-similarity as a function of time lag. To calculate this feature, a similarity matrix is used, with this matrix being obtained by comparing the spectral similarity between all possible pairs of frames in the original audio signal (Foote et al., 2002).

2. **Beat Location** - The algorithms proposed to calculate this feature estimate the beat location in an input signal in order to track the beat over the course of the audio signal. Several frameworks implement different versions of this principle.

3. **Onset Time** - An alternative away of determining tempo can be achieved by computing an onset detection curve that shows the successive bursts of energy that correspond to successive pulses. Afterwards, by performing peak picking on the detection curve, the positions of the note onsets can be estimated (i.e. the instants where each note starts).

4. **Event Density** - This feature present in MIR ToolBox estimates the "speed" of a song based on the average number of events in a given time window, i.e., the number of note onsets per second (Lartillot, 2018).

5. **Average Duration of Events** - One possible method to estimate the average duration of events was proposed by (Peeters et al., 2011). This method consists of detecting attack and release phases and then measuring the time in seconds between them when the amplitude is at least 40% of the maximum.

6. **Tempo** - Tempo, usually indicated in Beats Per Minute (BPM), refers to the speed of a musical piece. This feature is typically estimated by detecting the periodicities using the onset detection curve (Lartillot, 2018).

7. **Tempo Change** - By computing the difference between successive values of the tempo curve, it is possible to estimate an indicator of the tempo change over time. This feature is computed from the ratio of tempo values between consecutive frames (Lartillot, 2018).

8. **Predominant Local Pulse (PLP) Novelty Curves** - (Grosche and Müller, 2009) presented a mid-level representation aimed at capturing dominant tempo and predominant local pulse even from musical pieces with weak non-percussive note onsets and strongly fluctuating tempo. This PLP curve does not represent high-level information but is used instead as a tool in tasks such as tempo estimation and beat tracking, amongst others.

9. **Harmonically Wrapped Peak Similarity (HWPS)** - (Tzanetakis, 2002) proposed a set of rhythmic features using beat histograms of a song that proved useful in musical genre classification, with those features being:

   (a) **A0 and A1** - Relative amplitude of the first and second histogram peaks (A0 and A1, respectively).

   (b) **RA** - Ratio of the amplitude of the second peak divided by the amplitude of the first peak.

   (c) **P1 and P2** - Period of the first and second peak in BPM.

(d) **SUM** - Histogram sum (used as an indication of beat strength).

HWPS, a feature that follows similar principles has been implemented in frameworks such as Marsyas to calculate harmonicity by using the spectral information (Lagrange et al., 2008).

10. **Pulse/Rhythmic Clarity** - This feature estimates the "rhythmic clarity", which is an indicator of the strength and clarity found in the beats estimated by tempo estimation algorithms.

11. **Metrical Structure** - This feature detects periodicities from the onset detection curve and tracks a broad set of metrical levels, thus providing a detailed description of the metrical structure (Lartillot, 2018).

12. **Metrical Centroid and Strength** - These two features provide two descriptors:

    (a) **Dynamic metrical centroid** - Estimation of the metrical activity, based on computing the centroid of the selected metrical level (Lartillot, 2018).

    (b) **Dynamic metrical strength** - Indicator of the clarity and strength of the pulsation. It estimates whether there is "clear and strong pulsation, or even a strong metrical hierarchy is present", or if there is not: "the pulsation is somewhat hidden, unclear" or a mixture of pulsations (Lartillot, 2018).

13. **Note Duration statistics** - (Panda et al., 2018) proposed a set of note duration statistics based on the duration of each notes (the same ones proposed for the melody section).

14. **Note Duration Distribution** - Furthermore, in (Panda et al., 2018) a set of note distribution features was also proposed: Short Notes Ratio, Medium Length Notes Ratio and Long Notes Ratio.

15. **Ratios of Note Duration Transitions** - Finally (Panda et al., 2018) also proposed ratios of note duration transitions, such as: Transition to Longer Notes Ratio,. Transition to Shorter Notes Ratio and Transition to Equal Notes Ratio, a set of features similar to the melody dimension.

3.2.5 Dynamics features

In this section a description of the features found in literature that represent information related with dynamics and its components is presented.

1. **Root-Mean Square (RMS) Energy** - The RMS energy is used to measure the power of a signal either globally or over a certain window. It is usually calculated by taking the RMS (Tzanetakis, 2002). It is also roughly describes the loudness of a specific audio signal.

2. **Low Energy Rate** - This feature tracks the amount of frames that have less-than-average energy (Tzanetakis, 2002). By estimating the temporal distribution of energy in an audio clip, it allows for understanding if the energy remains constant between frames or if there is contrast.

3. **Sound Level** - This descriptor, represented in decibel, corresponds to the power sum of the spectrum in each time window. At a higher level it represents the unweighted sound pressure level of the signal in each of the analysis windows (Cabrera et al., 2008).

4. **Instantaneous Level, Frequency and Phase** - These three features consist in applying a Hilbert transform to the audio waveform, resulting in the three different outputs. The Instantaneous level can represent as the sound pressure level derived from the Hilbert transform (Cabrera et al., 2008).

5. **Loudness** - The loudness of a sound is a subjective metric that represents how intense a sound is perceived to be. This metric is measured in sones, where a doubling of the sones value represents to a doubling in the value of loudness. Several metrics for loudness have been proposed in the literature over the years and are available in the audio frameworks.

6. **Timbral Width** - Timbral Width was proposed by (Malloch, 1997) as one of the six measures of timbre in a method called loudness distribution analysis. This feature can be regarded as "a measure of the fraction of loudness that lies outside of the loudest band, relative to the total loudness" (Malloch, 1997).

7. **Volume** - Volume represents the perceived "size" of a sound, or the auditory volume of pure tones. (Cabrera, 1999) developed a computational volume model for arbitrary spectra, which was incorporated into audio frameworks. In this work, two diotic volume models were proposed:

   (a) The first model uses a weighted ratio between the binaural loudness and sharpness, which corresponds to the specific loudness centroid in the Bark scale.

   (b) The second model, which performed better, uses a simpler centroid to overcome limitations of the sharpness calculation method that the authors used (Cabrera, 1999).

8. **Sound Balance** - This metric is used to understand how much the peak (maximum amplitude) in a sound is off the center. In order to calculate this, the ratio between the index of the maximum (or minimum) value of the sound envelope of a signal and the total length of the sound envelope. If this amplitude peak is found close to the start, this ratio will approach 0. A ratio of 1 means the peak is close to the end and a value of 0.5 means that the peak is closer to the middle (Shannon, 1948).

9. **Note Intensity statistics** - In (Panda et al., 2018), a set of 6 statistics was proposed based on the median pitch salience of each note (this set of statistics is the same as aforementioned set for the melodic features).

10. **Note Intensity Distribution** - In (Panda et al., 2018), note intensity distribution features were also proposed. These features indicate how the notes of the predominant melody are distributed across three intensity ranges (low, medium and high intensity), leading to the following features:

    - Low Intensity Notes Ratio
    - Medium Intensity Notes Ratio
    - High Intensity Notes Ratio

    The same three features were also computed per second.

11. **Ratios of Note Intensity Transitions** - Also proposed in (Panda et al., 2018) were features that capture information related to the ratios of note intensity features, including Transitions to Higher Intensity Notes Ratio, Transitions to Lower Intensity Notes Ratio and Transitions to Equal Intensity Notes Ratio. Figure 3.4 showcases an example of these features.

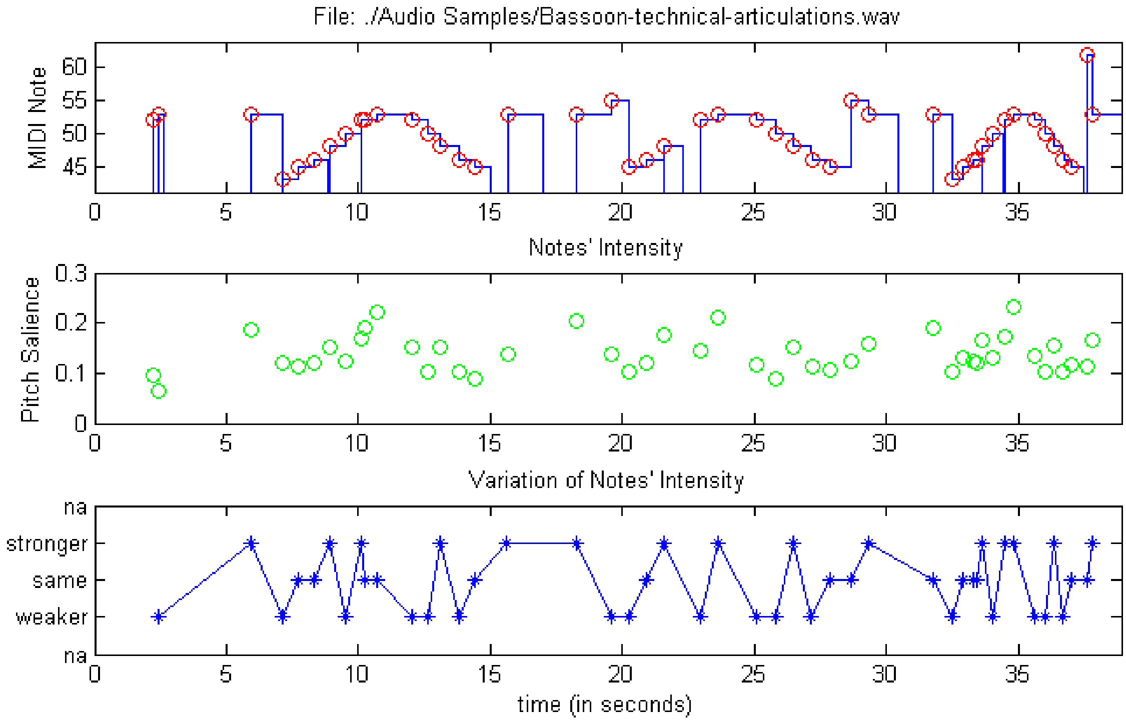


Figure 3.2: Changes of intensity in consecutive notes, from (Panda, 2019), p. 188.

12. **Crescendo and Decrescendo metrics** - Finally, (Panda et al., 2018) proposed a way of identifying notes as having crescendo or decrescendo based on the intensity difference between the first half and the second half of the note. Using this, the number of crescendo and decrescendo notes is computed (both per note and per second). Afterwards, sequences of notes with both increasing and decreasing intensity are computed, computing the number of sequences for both cases (both per note and per second) and also the length of the crescendo sequences in notes and seconds, using the 6 aforementioned statistics.

3.2.6 Tone Color features

In this section a description of the features found in literature that represent information related with tone color and its components is presented. This dimension is the most represented one out of all the eight musical dimensions. Most of the features are low-level temporal features or spectral descriptors that are employed in several audio analysis problems (e.g. zero-crossing rate, spectral moments, amongst others). Furthermore, other common features include attack/decay time, attack slope and leap, amongst others, which will be explained in detail below.

1. **Attack/Decay Time** - One of the aspects that influences tone color is the sound envelope of the audio clip. This sound envelope can be divided into four parts: attack, decay, sustain and release. From this sound envelope several descriptors can be extracted, with most of them being related to the attack phase, i.e., from the starting point of the sound envelope until the amplitude peak is reached. One of these descriptors is the attack time, which consists in estimating the temporal duration of the various attack phases in the audio signal. (Peeters et al., 2011).

2. **Attack/Decay Slope** - The attack slope is another descriptor that is extracted from the attack/decay phase of the sound envelope (Peeters et al., 2011). The attack slope consists on estimating the average slope of the entire attack/decay phase, since its start until its peak/valley.

3. **Attack/Decay Leap** - This simple descriptor consists on estimating the amplitude difference between the beginning s(bottom/top) and the end (peak/valley) of the attack/decay phase (Lartillot, 2018).

4. **Zero Crossing Rate (ZCR)** - Zero Crossing Rate represents the number of times that the audio waveform changes sign in a window (e.g. crosses the x-axis). This can be used as a simple indicator of frequency or noisiness.

5. **Spectral Flatness** - This feature indicates if the spectrum distribution is smooth or spiky, i.e., estimate whether or not the frequencies in the spectrum are uniformly distributed (Lartillot, 2018).

6. **Spectral Crest Factor (SCF)** - The spectral crest factor (Allamanche, 2001) is a measure of the "peakiness" of a spectrum and is thus inversely proportional to the spectral flatness measure. It is commonly used to distinguish noise-like sounds from tone-like sounds because of the different spectral shapes, where noise-like sounds have lower spectral crests.

7. **Irregularity** - This feature, also known as spectral peaks variability, corresponds to the degree of variation of the amplitude values of successive spectral peaks (Lartillot, 2018).

8. **Tristimulus** - The tristimulus feature (Peeters et al., 2011), quantifies the relative energy of partial tones by parameters that measure the energy of certain partials, as such:

- **Tristimulus1** - Corresponds to the energy of the first partial.
- **Tristimulus2** - Corresponds to the energy of the second, third and fourth partials.
- **Tristimulus3** - Corresponds to the energy of the remaining partials.

9. **Odd-to-even Harmonic Energy Ratio** - The odd-to-even harmonic energy ratio ““distinguishes sounds with predominant energy at odd harmonics (such as clarinet sounds) from other sounds with smoother spectral envelopes (such as the trumpet)” (Peeters et al., 2011).

10. **Spectral Moments: Centroid, Spread, Skewness and Kurtosis** - The four spectral moments (implemented in several frameworks) are metrics used to measure spectral shape (Peeters et al., 2011). These moments can be described as such:

    - **Spectral Centroid** - The spectral centroid corresponds to the first moment (mean) of the magnitude spectrum of the Short-Time Fourier Transform (STFT).
    - **Spectral Spread** - The spectral spread represents the standard deviation of the magnitude spectrum, and thus it is a measure of the dispersion (or spread) of the spectrum.
    - **Spectral Skewness** - Spectral skewness is a measure of the symmetry of the spectrum.
    - **Spectral Kurtosis** - Spectral kurtosis captures information about the existing outliers in the magnitude spectrum.

11. **Spectral Entropy** - Spectral entropy is a measure of the spectral power distribution, based on the Shannon entropy (Shannon, 1948) metric from the information theory field.

12. **Spectral Flux** - Spectral flux measures the amount of spectral change in the audio signal, i.e., the distance between the spectra of successive frames (Tzanetakis, 2002). It has shown to be an important attribute in the characterization of timber of musical instruments (Grey, 1975).

13. **Spectral Rolloff** - Spectral flux represents an indicator of the skewness of the frequencies present in a window. According to (Tzanetakis, 2002), the spectral rolloff is defined as the frequency R_t below which 85% of the magnitude distribution is concentrated. This percentage varies between authors, however 85% is the default value in most audio frameworks. Figure 3.3 showcases an example of spectral rolloff.

14. **High-frequency Energy** - Several algorithms have been proposed to try and determine the amount of high-frequency energy in a signal. One of these is algorithms is called brightness, in which a minimum frequency value is set and the amount of energy above that value is measured (Lartillot, 2018).

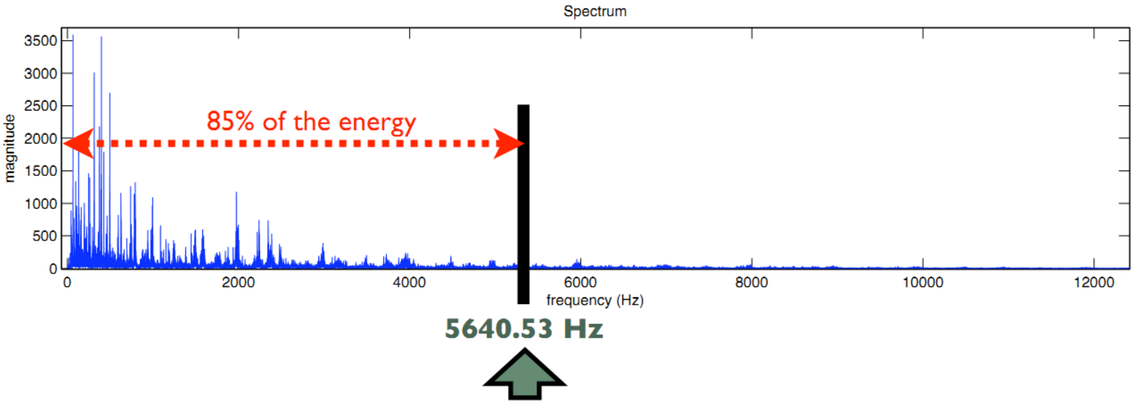


Figure 3.3: Spectral rolloff (Lartillot, 2018).

15. **Cepstrum** - Cepstrum is the result of taking the inverse Fourier transform of the logarithm of the estimated spectrum of an audio signal (Bogert, 1963). It can be described as a measure of the rate of change in different spectral brands. Ceptral analysis is used in several areas such as pitch analysis, human speech processing, amongst others. It provides a simple way to to separate formants from the vocal source.

16. **Energy in Mel/Bark/ERB Bands** - In order to better analyze an audio signal, it is important to decompose the original audio signal into a series of audio signals of different frequencies so that each channel can be studied independently. There are several scales for splitting the frequencies into critical bands, such as the Mel, Bark or the Equivalent Rectangular Bandwith (ERB) scale (Harrington and Cassidy, 1999).

17. **Mel-Frequency Cepstral Coefficients (MFCC)** - MFCCs (Davis and Mermelstein, 1980) are another measure of spectral shape. The frequency bands are positioned logarithmically on the Mel scale and cepstral coefficients are then computed based on the Discrete Cosine Transform of the log magnitude spectrum. In most audio frameworks, only the 13 first coefficients are returned. These 13 coefficients are used mostly for speech representation however (Tzanetakis, 2002) states that “the first five coefficients are adequate for music representation”.

18. **Linear Predictive Coding Coefficients (LPCC)** - Linear predictive coding is used in speech research by a linear predictive model to represent the spectral envelope of a digital speech signal in compressed form (El Ayadi et al., 2011). LPCCs represent the ceptral coefficients derived from linear prediction and are used in many areas regarding speech, such as speech analysis, encoding, amongst others (El Ayadi et al., 2011).

19. **Linear Spectral Pair (LSP)** - Linear Spectral Pair (LSP) are an alternative representation of Linear Prediction Coefficient (LPC) for transmission over a channel. LSPs have many properties that make them superior to LPCs, such as for example smaller sensitivity to quantization noise. Thus, LSPs are useful in speech recognition and encoding (Zheng et al., 2000).

20. **Spectral Contrast** - Spectral contrast is a feature proposed in (Jiang et al., 2002a). It represents the spectral characteristics of the audio signal, specially the relative spectral distribution. According to the authors, this feature was tested in music type classification problems, demonstrating "better discrimination among different music types than mel-frequency cepstral coefficients (MFCC)s" (Jiang et al., 2002a), with MFCCs being one of the features typically used in these types of problems.

21. **Roughness (Sensory Dissonance)** - Sensory dissonance, also known as roughness, relates to the beating phenomenon whenever a pair of sinusoids are close in frequency (Plomp and Levelt, 1965).

22. **Spectral and Tonal Dissonance** - Dissonance measures the harshness or roughness of the acoustic spectrum (Cabrera et al., 2008). Dissonance generally implies a combination of notes that sound harsh or unpleasant to people when played at the same time. The audio framework PsySound3 provides two descriptions of acoustic dissonance:

    - **Spectral Dissonance** - Dissonance that uses all the Fourier components.
    - **Tonal Dissonance** - Dissonance that uses a peak extraction algorithm before calculating dissonance.

3.2.7 Expressivity features

The amount of features that capture information related to the expressiveness of a song is low, thus (Panda et al., 2018) proposed a set of new features to capture this information, such as vibrato [4], tremolo [5], glissando [6] and legato [7]. The features related to expressiveness can be summed up as such:

1. **Average Silence Ratio** - This feature was proposed as an estimation of articulation in (Feng et al., 2003). It is the ratio of silence frames to total frames in time windows of one second. A lower ratio means that there are fewer silence frames in the musical piece, meaning legato occurs (notes being played "smoothly"). A higher ratio mean more silence frames in the musical piece, meaning staccato occurs (notes being short and detached from each other).

2. **Portamento metrics** - Portamento refers to the smooth and monotonic decrease or increase in pitch on consecutive notes. Computational models to calculate this feature were proposed using Hidden Markov Models in a flattened out pitch curve (no vibrato) (Yang et al., 2016).

[4] Periodic changes in the pitch of a tone.

[5] Periodic changes in the intensity of a tone.

[6] Frequency slide in the attack of a note.

[7] Performing style where notes appear to be "connected" without any perceptible break between them.

3. **Articulation metrics** - Articulation is a technique that affects the transition or continuity between notes or sounds. In (Panda et al., 2018) an approach to detect staccato and legato was proposed. The algorithm proposed classified all the transitions between notes in a music piece and from the transitions several metrics were extracted such as the ratio of legato, staccato and other transitions. The longest sequence of each articulation type was also extracted.

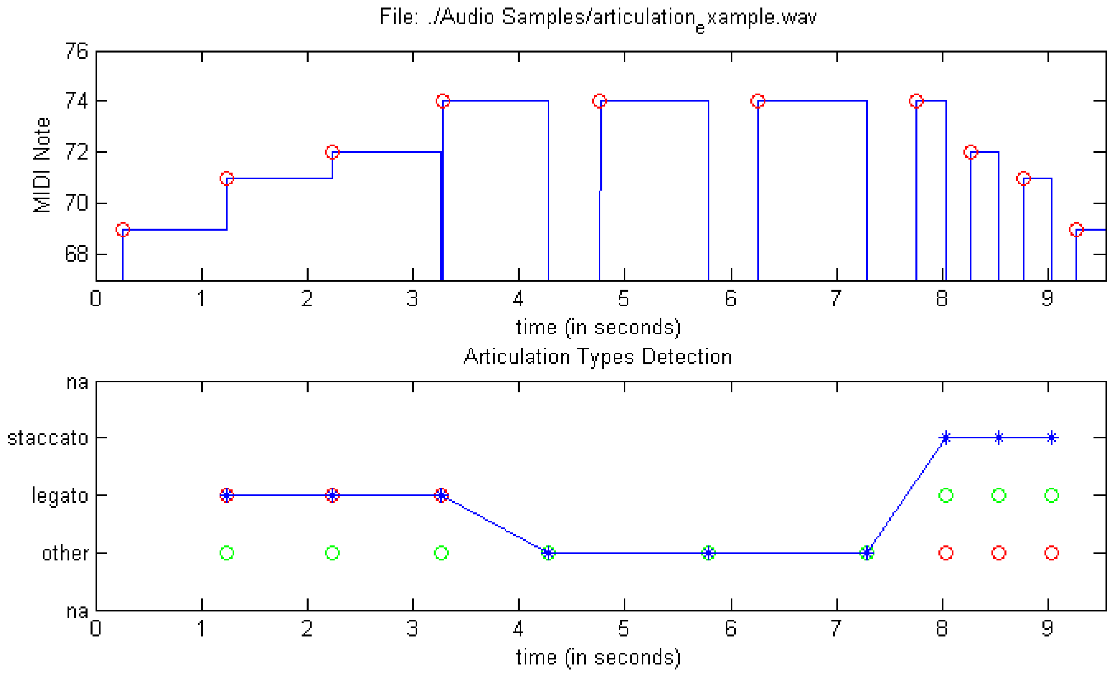


Figure 3.4: Testing articulation extraction with different note durations and intervals, from (Panda, 2019), p. 192.

4. **Glissando metrics** - Glissando is another type of articulation, consisting of a glide from one note to another. It is used as an ornamentation, often to add interest to a piece, and thus may be related to specific emotions. In (Panda et al., 2018) an algorithm to detect glissando as proposed, and based on that algorithm several features are extracted, such as glissando presence, extent, duration, direction, slope and glissando to non-glissando ratio (i.e. the ratio of notes containing glissando compared to the total number of notes.

5. **Vibrato metrics** - Vibrato is a technique used in both vocal and instrumental musical that can be defined as regular oscillation of a pitch. There are two main characteristics regarding vibrato: the amount of pitch variation (i.e. the extent) and the velocity (i.e. the rate) of the pitch variation. In (Panda et al., 2018) an algorithm was proposed to detect vibrato. This algorithm analysis the F0 sequence of each note, and then extracts several features such as vibrato presence, rate, extent, coverage, high-frequency coverage, vibrato to non-vibrato ratio and the base frequency of the vibrato notes. Figure 3.5 showcases an example of these metrics in two different excerpts.

6. **Tremolo metrics** - Tremolo is a trembling effect, similar to vibrato but regarding a change in amplitude, not pitch. It is noteworthy that in (Panda

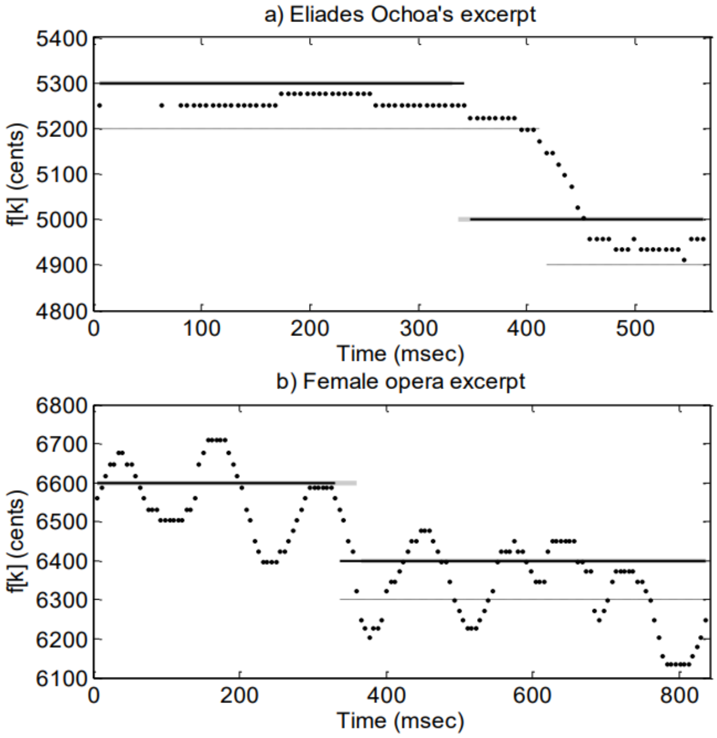


FIGURE 4.4. ILLUSTRATION OF GLISANDO AND VIBRATO.

Figure 3.5: Illustration of glissando and vibrato. Top excerpt from Eliades Ochoa's "Chan Chan" with glissando and bottom excerpt is a recording from opera singer Eith with vibrato.

et al., 2020a) no relation was found between tremolo and emotion. Nonetheless, in (Panda et al., 2018) a tremolo detection algorithm was proposed, similar to the vibrato detection algorithm, and a similar set of features was extracted. In the algorithm, instead of using F0 note sequences, a sequence of pitch saliences of each note is used, because tremolo represents a variation in the intensity or amplitude of a note.

3.2.8 Texture features

According to (Panda et al., 2020a), there were no audio features in the analyzed audio frameworks that are related with musical texture. As such, in (Panda et al., 2020a), a set of novel musical texture features was proposed. These features are based on the sequencing of multiple frequency estimates that are employed to measure the number of simultaneous layers in each of frame of the whole audio signal. This led to the proposal of the following features:

1. **Musical Layer Statistics** - Based on the number of multiple predominant melody fundamental frequency (F0) estimated from each of the frames of the audio clip, 6 statistics are calculated regarding the distribution of the number of musical layers across the frames. The number of layers in a frame is defined as the number of multiple F0s in that frame.

2. **Musical Layer Distribution** - The estimated number of F0s in one frame is divided into four classes:

   (a) No layers

   (b) One layer

   (c) Two layers

   (d) Three or more layers

   For each class the percentage of frames belonging to that class is calculated.

3. **Ratio of Musical Layer Transitions** - To capture information regarding changes from one musical layer sequence to another, consecutive frames with different number of estimated F0s are identified as transitions and the total value of the length of the audio segment is normalized (in seconds). Furthermore, the length in seconds of the longest segment for each musical layer is also computed.

3.2.9 Form features

Despite the relevancy of musical form for emotion classification (Friberg et al., 2014), there are few computational extractors for this group of features. This can also be acquainted to the fact that extracting this information from the audio signal is harder compared to other lower level features (e.g. spectral statistics).

The most commonly used features for musical form are the following:

1. **Structural Change**. The amount of change of each of the basis features at different time intervals, combined into a meta-feature, has correlation with the human perception of complexity in music (Mauch and Levy, 2011). The typical implementation of this feature uses chroma, rhythm and timbre information and aims to exclusively discover the amount of change in a song, illustrating it with a visual audio flower pot (Mauch and Levy, 2011).

2. **Similarity Matrix** - Some approaches estimate the musical structure of a song based on the similarity between adjacent segments or frames (Lartillot, 2018). These similarities can be represented by a inter-frame (or segment) similarity matrix which shows the differences between all possible pairs of frames of a given audio signal. To compute this matrix a set of frame statistics (e.g. spectral features) is used with a distance function, to calculate the proximity between different pairs of frames (Lartillot, 2018).

3. **Novelty Curve** - Based on the specific musical characteristics of a segment or frame, a novelty curve can be obtained by comparing successive frames to estimate the temporal changes in a song (Lartillot, 2018).

4. **Higher Level (HL) Form Analysis** - The best models combine higher-level solutions with low-level features, statistics and machine learning to model the fundamental aspects of musical sections in order to identify song elements in a song (e.g. intro, bridge, chorus). State of the art results in form analysis have been obtained using deep learning models (Wang et al., 2022).

3.3 Proposal of new features

As aforementioned, (Panda et al., 2020a) also proposes a few research directions in order to create new features, and those are the basis of this work, being the following:

1. **Musical Texture** - In terms of musical texture, research was conducted into algorithms that do source and instrument separation for polyphonic music, with the most relevant and most promising of them being MT3, which in short, enables the extraction of the melody lines for each instrument present in the song (Gardner et al., 2021) (See Section 3.5.2 for further information). This framework will enable the extraction of features related to each instrument.
2. **Musical Melody** - (Gardner et al., 2021) also provides a way of performing melody transcription, with efforts already being made in the current phase of this work to test the impact of using Magenta MT3 as the melody transcriptor instead of the current solutions. Besides the transcription of main melody channel, MT3 permits full music transcription, i.e., transcription of all melodic lines. Features from these accompaniment melodic lines are also exploited in this work.
3. **Musical Rhythm** - There is also ongoing work from our research group regarding rhythm, particularly in the percussion realm. Percussion features are not yet used in the MER field, and are going to be explored in this work, such as the existence of percussion, the various types of percussion, amongst others.

Furthermore, with the recent development and performance increase in tools for voice and accompaniment separation, in this work exploration such as extracting features only from the isolated voice or isolated instruments will also be conducted, as those aspects have not yet been fully explored in the MER field.

Thus, there are several possible new features that will be created and explored in the realm of this work.

3.4 Static MER

This section provides an overview of approaches to solve the problem of emotion classification in Static MER.

3.4.1 Classical approaches

With classical approaches in the MER field, the process has remained the same for roughly 20 years. Firstly, a song is selected for annotation. Most of the time, an audio snippet of the song is picked. This audio snippet will be analyzed and annotated by either professionals or people without experience based on an emotional model (e.g. asking the listener to annotate a song with “happy” or “sad”).

Afterwards, a set of audio features is extracted from the samples and these features are used by classifiers to try and classify the samples onto the correct categories (e.g. onto the correct quadrants) and then the classifiers deliver an outcome on the form of a prediction. However, these audio features are picked by humans that try and find patterns between the samples and the annotations, which means that the ML models are only as good as the features given to them.

Moreover, this is not simply a problem of feeding as many features as possible onto the models, as that can have the opposite effect and lead to even worse results. Thus, finding adequate and high quality features is paramount, and this has proven to be a very hard task as it requires a good amount of knowledge in the music field.

Over the years, there have been many approaches to the emotion classification problem, starting with (Feng et al., 2003). In this work, 223 songs of modern popular music were extracted. Out of those 223 pieces, 200 were used for training and 23 were used for testing. In terms of annotations, the songs were annotated into four different emotions: happiness, sadness, anger and fear. The results, in terms of classification accuracy, were good, varying from 75% in sadness, 83% for anger, and 86% for happiness, however, fear only got 25% accuracy.

This can be explained by the small dataset in which there is not a balance between all the four classes, with mainly the fear class being poorly represented. Moreover, only three features were extracted.

In another study, (Lu et al., 2006), the Russell’s model emotion taxonomy was used. In this work, there was a total of 200 audio clips per quadrant, totalling 800 audio clips with a duration of 20 seconds. However, this dataset had the problem of only being derived from 250 pieces of classical music, thus not being representative of whole of the genres in music. Nonetheless, an accuracy of 86.3% was obtained. Like (Feng et al., 2003), this approach also had the problem of only extracting three features, with this becoming a recurring problem in MER approaches.

In (Meyers, 2007), an approach was created using both features extracted from the audio files and the lyrics in order to classify the emotional contents of the song using Russell’s model. The approach uses 5 features: mode, harmony, tempo, rhythm and loudness. First, a decision tree algorithm is used for preliminary classification of the song followed by a K-nearest Neighbor (KNN) classification algorithm that will classify the song into one of eight categories. To get the predicted global emotion of a song, the output of the KNN algorithm is combined with the lyrics’ affective value. According to the author, good results were obtained when comparing the obtained tags against the AllMusic expert tags. However, the authors do not provide a statistical way to analyze the performance of the models.

In the approach proposed by (Panda et al., 2015), a balanced dataset of 903 songs was created with 30 seconds in length by mapping the AllMusic mood tags into the five clusters emotion taxonomy used by MIREX, which has been criticized for not having the support of psychological studies. The goal of this approach was to try and combine standard and melodic features. In total, 351 features were extracted. However, the best result, an f-measure value of 64%, was obtained with only 11 features (9 melodic and 2 standard), using a Support Vector Machine (SVM) model. These top 11 features were obtained using the RelieF feature selection algorithm (Robnik-Sikonja and Kononenko, 1997).

More recently, in the approach proposed by (Panda et al., 2020b), a balanced dataset of 900 songs (the 4QAED dataset) was created, previously mentioned in 3.1. Moreover, a number of new acoustic features were created in conjunction with using state of the art features. As previously conducted in (Panda et al., 2015), feature ranking was performed using the RelieF algorithm and classification was done using SVMs. An f1-score of 76.4% was obtained, which was a 9% increase compared to the previous work. This improvement was mainly due to the new proposed novel features, thus proving that there is a lack of emotion relevant features in the MER field.

3.4.2 Deep Learning approaches

With the rise of deep learning approaches not only in the MER field, but in the whole world of machine learning, combined with the stagnation that classical approaches had sustained in the MER field, more and more research has been done with deep learning in the MER field. As aforementioned, deep learning requires huge amount of data compared to classical approaches, which is a big problem in the MER field as there do not exist quality datasets with big enough size that allow researchers to take full advantage of the capabilities of DL approaches, with techniques such as data augmentation having to be employed in order to try and improve the results obtained.

Furthermore, there is the problem of interpretability with deep learning models. With the increasing usage of deep learning models so have the concerns over interpreting the results of these models. It is not enough for a model to classify a certain song with a certain emotion or place it inside a certain quadrant. Knowing why it was classified as such is also important.

One of the advantages that deep learning models have versus the classical approaches is speed, as most of the classical approaches use frameworks for extracting audio features (e.g. Essentia, Marsyas, MIRtoolbox), and some of these approaches are old and are built on older platforms like MATLAB, making the feature extraction process very time consuming, sometimes taking tens of minutes per song.

In (Cañón et al., 2021), the idea of transfer learning in the MER field was explored. Transfer learning, as the name suggests, takes advantage of trying to take knowledge gained by a machine learning model in one domain and applying it to a different but related domain. This was explored in this work by using unsupervised feature learning trained in speech from English and Mandarin and transferring that concept onto emotion classification with some fine tuning. The work was conducted using the Russell model of emotion and the 4QAED database aforementioned. The results obtained were around 48% f1-score, while not impressive, do confirm that there is cross-domain transferability.

In another approach by (Grekow, 2021), RNNs were used for emotion detection. These neural networks NN were used to predict continuous emotion values in the Russell model. The training dataset consists of 324 6-second segments of different genres of music (classical, jazz, blues, country, disco, hip-hop, metal, pop, reggae, and rock). The tracks were taken from the publicly available GTZAN dataset (Tzanetakis and Cook, 2002). Five experts in music were asked to anotate every clip in the dataset in the Russell model of emotion on a range of [-10,10] for arousal and for valence. The final dataset has very good agreement between the various annotators and is also publicly available.

The features for these NN were obtained using audio analysis and music information retrieval tools for audio, such as Marsyas and Essentia. In total, 529 features were extracted. The obtained results using the features from Marsyas achieved a mean squared error of 0.67 and a Mean Absolute Error (MAE) of 0.12 for arousal. For valence, the mean squared error was 0.17 and the obtained MAE was 0.12. The results for Essentia are very similar, with only a few percentage points of difference.

Moreover, another approach was tested using the Essentia features. The idea is to use a pre-trained model to process the Essentia features in order to select the most relevant features, and use the output of this model as input of the RNN. The results obtained using this method were the best results, with a mean squared error of 0.73 and MAE of 0.11 for arousal and 0.46 mean squared error and 0.12 MAE for valence.

One recent approach was done by (Louro, 2022), that tried various deep neural networks, for example CNN and fully connected neural networks amongst other architectures. Both mel-spectrograms and hand-crafted features were tested as inputs. Due to the limited amount of data, data augmentation had to be done in order to increase the dataset, with techniques such as tempo shifting being employed. Furthermore, like in (Cañón et al., 2021), transfer learning from speech emotion recognition was also tested. The tests were conducted on the 4QAED dataset using the Russell model of emotion. The best results, an f1-score of 73.7%, still fall short of the best results obtained using SVMs (76.4%). However, with the increase in dataset a resulting increase in f1-score was observed, with this also being one of the motivating factors for the expansions of the 4QAED dataset conducted in this work (Section A.3).

There are other approaches that used various datasets, including the Million Song Dataset. However, with the problems aforementioned with this particular dataset (e.g. the poor annotation process) the results obtained on these datasets are very lackluster, with very low classification scores compared to smaller but higher quality datasets like the 4QAED. This also reinforces the need for creation of high quality datasets, with this being one of the main objective of this work.

Paper	Approach	Emotion Taxonomy	Datasets	Features and Input	Models	Results	Notes/Observations
(Feng et al., 2003)	Classical ML	Four emotions: Happiness, sadness, anger and fear	223 popular songs	5 features	Feed-forward network	Accuracy values of 75% for sadness, 83% anger, 86% happiness and 25% for fear	Unbalanced dataset - fear class is poorly represented. Only three features extracted.
(Lu et al., 2006)	Classical ML	Russell's A/V Model	800 snippets (20-seconds) of 250 classical music pieces	Standard audio features	Gaussian mixture model	0.863 accuracy	The pieces used are all of the same genre and only three features were extracted.
(Meyers, 2007)	Classical ML	Russell's A/V Model	372 songs	5 features	Decision Tree for classification and KNN for emotion prediction	Good results when compared to All-Music Tags	The authors do not provide a statistical way to analyze performance of the models.
(Panda et al., 2015)	Classical ML	MIREX's task 5 clusters	903 songs	11 features (9 melodic and 2 standard)	SVM (top result)	0.64 f-measure	The emotional taxonomy used does not have any psychological support.

Table 3.4: Review of Static MER approaches.

Paper	Approach	Emotion Taxonomy	Datasets	Features and Input	Models	Results	Notes/Observations
(Panda et al., 2020b)	Classical ML	Russell's A/V Model	4QAED	100 features (29 novel and 71 standard)	SVM	0.76 f-measure	
(Cañón et al., 2021)	Deep Learning	Russell's A/V Model	4QAED	Spectogram	CNN	0.48 f1-score	Results confirm that there is cross-domain transferability .
(Grekow, 2021)	Deep Learning	Russell's A/V Model	324 6-second samples	529 features	Modified CNN and RNN	MAE of 0.12 for arousal and 0.11 for valence	
(Louro, 2022)	Deep Learning	Russell's A/V Model	4QAED	Hand crafted features and spectogram	CNN and fully connected neural networks	0.737 f1-score	Result lower than SVMs however may increase with bigger datasets.

Table 3.5: Continuation of the review of Static MER approaches.

3.5 Automatic Music Transcription (AMT)

This section provides a brief overview of the current automatic music transcription tools, as well as providing a description of Magenta MT3 (Gardner et al., 2021) as it is one of the frameworks used in the scope of this work.

3.5.1 Limitations of current frameworks

In a survey conducted on AMT (Bhagwat et al., 2023), a few of the limitations of the current AMT approaches were presented. The main limitation, which is something that MT3 aims to help solve, is that most of the approaches are either Instrument Music Transcription (IMT) or Instrument Specific Music Transcription (ISMT), meaning that the tools have knowledge of which instruments are present in the audio track. Furthermore, other approaches use source separation before doing the music transcription, which adds another layer of complexity.

Finally, a lot of the approaches presented in (Bhagwat et al., 2023) suffer from similar problems, such as small datasets, datasets with low variety (e.g. only one type of instrument), or have poor results when there are many instruments presented in the track.

MT3 (Gardner et al., 2021) aims to solve all of these problems, by being trained on a mixture of various instrumental datasets, thus solving the problem of the small datasets. These datasets also contain a lot of instrumental variety.

3.5.2 Magenta MT3

This tool, proposed in (Gardner et al., 2021), aims to help solve the problem of Automatic Music Transcription (AMT). It was trained on a mixture of various instrumental datasets and does implicit source separation, followed by instrument classification and then transcription of the notes for each instrument, using sequence-to-sequence transfer learning. Figure 3.6 shows the model architecture and prediction on real 4-second audio clips.

The model was evaluated using a set of already existing metrics, particularly Frame F1 (a binary metric on whether the predicted and final notes match), Onset F1 (a metric that considers a prediction correct if it has the same pitch and is within 50ms of the referenced onset) and finally Onset-Offset F1 (as the name suggests, this metric combines the aforementioned metrics but now notes must also have matching offsets). A final metric is also presented in (Gardner et al., 2021), multi-instrument F1, which combined Onset-Offset F1 with the requirement that the instrument predicted to play a certain note has to match the original instrument of the reference note. This new proposed metric was only calculated for (Gardner et al., 2021), as it was not possible to compute on previous approaches.

Table 3.6 shows the results obtained by MT3 compared to other state of the art approaches.

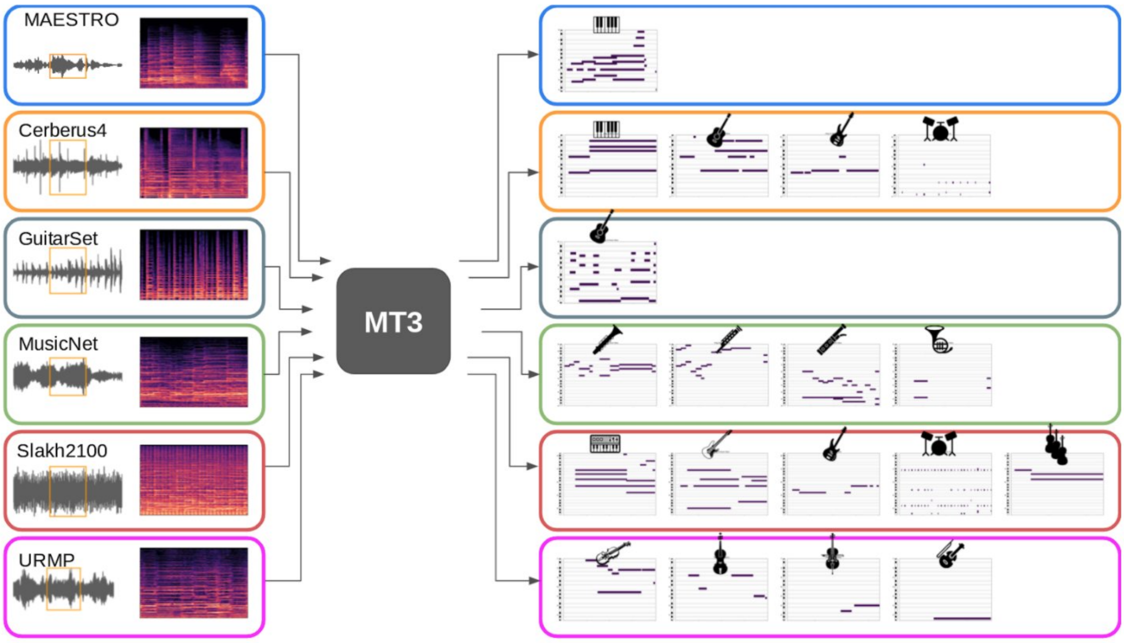


Figure 3.6: MT3 Model Architecture and prediction on real 4-second audio clips (Gardner et al., 2021).

Model	MAESTRO	Cerberus4	GuitarSet	MusicNet	Slakh2100	URMP
			Frame F1			
(Hawthorne et al., 2021)	0.66	–	–	–	–	–
(Manilow et al., 2020)	–	0.63	0.54	–	–	–
(Cheuk et al., 2021)	–	–	–	0.48	–	–
Melodyne	0.41	0.39	0.62	0.13	0.47	0.30
MT3 (single dataset)	**0.88**	0.85	0.82	0.60	0.78	0.49
MT3 (mixture)	0.86	**0.87**	**0.89**	**0.68**	**0.79**	**0.83**
			Onset F1			
(Hawthorne et al., 2021)	0.96	–	–	–	–	–
(Manilow et al., 2020)	–	0.67	0.16	–	–	–
(Cheuk et al., 2021)	–	–	–	0.29	–	–
Melodyne	0.52	0.24	0.28	0.04	0.30	0.09
MT3 (single dataset)	**0.96**	0.89	0.83	0.39	0.76	0.40
MT3 (mixture)	0.95	**0.92**	**0.90**	**0.50**	**0.76**	**0.77**
			Onset+Offset F1			
(Hawthorne et al., 2021)	0.84	–	–	–	–	–
(Manilow et al., 2020)	–	0.37	0.08	–	–	–
(Cheuk et al., 2021)	–	–	–	0.11	–	–
Melodyne	0.06	0.07	0.13	0.01	0.10	0.04
MT3 (single dataset)	**0.84**	0.76	0.65	0.21	0.57	0.16
MT3 (mixture)	0.80	**0.80**	**0.78**	**0.33**	**0.57**	**0.58**
Mixture (Δ%)	-5.3	+5.2	+19.5	+54.0	+0.1	+263

Table 3.6: Transcription F1 scores for Frame, Onset, and Onset+Offset metrics defined previously. Across all metrics and all datasets, MT3 consistently outperforms the baseline systems we compare against. Percent increase over single-dataset training for Onset+Offset F1 is shown in the last row. (Gardner et al., 2021)

The F1-scores obtained in the metrics mentioned are mostly contained in the range of 80% to 95% in the various datasets used, with these F1-scores being better than all the previous approaches as well as a professional-quality Digital Signal Processor (DSP) for polyphonic pitch transcription, Melodyne [8] . Some results on particular datasets show large improvements.

Due to the difficult nature of this task, the implementation for this model is very complex. The authors do not provide a set of instructions on how to run this tool locally, thus a lot of work had to be conducted in order to be able to run the tool locally.

The output of the MT3 model is a MIDI file that contains the channel information (for each of the instruments). It also follows the standard of reserving channel 10 for drum information. Each note has the information on whether or not it is drums, the pitch of the note, and the start and end times of the note. MT3 follows the standard 128 instrument representation for MIDI files as well as the standard drum representation.

Table 3.7 provides a list of the instrument identifier and the represented instrument. Furthermore, Table 3.8 provides a list of the different identifiers that represent each drum instrument.

Moreover, it was important for this work to classify the 128 instruments into groups. There are several types of classification used in literature, with the most common being the one proposed in (Sachs, 1914). This classification splits the instruments into 5 groups. A brief description of these groups can be summarized as such:

1. **Idiophones** - Idiophones produce sound by means of the actual body of the instrument vibrating, rather than a string.
2. **Chordophones** - Chordophones are simply a string or a set of strings and a string bearer.
3. **Membranophones** - Membranophones primarily produce their sounds by means of the vibration of a membrane. This group includes all drums and kazoos.
4. **Aerophones** - Aerophones primarily produce their sounds by means of vibrating air, with the instrument itself not vibrating and not containg any vibrating strings or membrane.
5. **Electrophones** - Electrophones are instruments that either have electric action (e.g. pipe organ with electrically controlled air valves), have electric amplification (e.g. modified piano with microphones inside it) or *radioelectric* instruments, where the sound is produced by electrical means.

There a few instruments in the MIDI that do not fit into any of these groups (e.g. Gunshot), so those were elements were assigned to the "Miscellaneous" group.

[8] https://www.celemony.com/

However, in all the songs, none of these instruments were ever present, so this group and the features related to it ended up being removed.

ID	Instrument	ID	Instrument	ID	Instrument	ID	Instrument
1	Acoustic Grand Piano	33	Acoustic Bass	65	Soprano Sax	97	FX 1 (rain)
2	Bright Acoustic Piano	34	Electric Bass (finger)	66	Alto Sax	98	FX 2 (soundtrack)
3	Electric Grand Piano	35	Electric Bass (pick)	67	Tenor Sax	99	FX 3 (crystal)
4	Honky-tonk Piano	36	Fretless Bass	68	Baritone Sax	100	FX 4 (atmosphere)
5	Electric Piano 1	37	Slap Bass 1	69	Oboe	101	FX 5 (brightness)
6	Electric Piano 2	38	Slap Bass 2	70	English Horn	102	FX 6 (goblins)
7	Harpsichord	39	Synth Bass 1	71	Bassoon	103	FX 7 (echoes)
8	Clavinet	40	Synth Bass 2	72	Clarinet	104	FX 8 (sci-fi)
9	Celesta	41	Violin	73	Piccolo	105	Sitar
10	Glockenspiel	42	Viola	74	Flute	106	Banjo
11	Music Box	43	Cello	75	Recorder	107	Shamisen
12	Vibraphone	44	Contrabass	76	Pan Flute	108	Koto
13	Marimba	45	Tremolo Strings	77	Blown Bottle	109	Kalimba
14	Xylophone	46	Pizzicato Strings	78	Shakuhachi	110	Bagpipe
15	Tubular Bells	47	Orchestral Harp	79	Whistle	111	Fiddle
16	Dulcimer	48	Timpani	80	Ocarina	112	Shanai
17	Drawbar Organ	49	String Ensemble 1	81	Lead 1 (square)	113	Tinkle Bell
18	Percussive Organ	50	String Ensemble 2	82	Lead 2 (sawtooth)	114	Agogo
19	Rock Organ	51	Synth Strings 1	83	Lead 3 (calliope)	115	Steel Drums
20	Church Organ	52	Synth Strings 2	84	Lead 4 (chiff)	116	Woodblock
21	Reed Organ	53	Choir Aahs	85	Lead 5 (charang)	117	Taiko Drum
22	Accordion	54	Voice Oohs	86	Lead 6 (voice)	118	Melodic Tom
23	Harmonica	55	Synth Choir	87	Lead 7 (fifths)	119	Synth Drum
24	Tango Accordion	56	Orchestra Hit	88	Lead 8 (bass + lead)	120	Reverse Cymbal
25	Acoustic Guitar (nylon)	57	Trumpet	89	Pad 1 (new age)	121	Guitar Fret Noise
26	Acoustic Guitar (steel)	58	Trombone	90	Pad 2 (warm)	122	Breath Noise
27	Electric Guitar (jazz)	59	Tuba	91	Pad 3 (polysynth)	123	Seashore
28	Electric Guitar (clean)	60	Muted Trumpet	92	Pad 4 (choir)	124	Bird Tweet
29	Electric Guitar (muted)	61	French Horn	93	Pad 5 (bowed)	125	Telephone Ring
30	Overdriven Guitar	62	Brass Section	94	Pad 6 (metallic)	126	Helicopter
31	Distortion Guitar	63	Synth Brass 1	95	Pad 7 (halo)	127	Applause
32	Guitar Harmonics	64	Synth Brass 2	96	Pad 8 (sweep)	128	Gunshot

Table 3.7: List of instruments using the standard MIDI classification.

ID	Drum type	ID	Drum type
35	Acoustic Bass Drum	58	Vibraslap
36	Bass Drum 1	59	Ride Cymbal 2
37	Side Stick	60	Hi Bongo
38	Acoustic Snare	61	Low Bongo
39	Hand Clap	62	Mute Hi Conga
40	Electric Snare	63	Open Hi Conga
41	Low Floor Tom	64	Low Conga
42	Closed Hi Hat	65	High Timbale
43	High Floor Tom	66	Low Timbale
44	Pedal Hi-Hat	67	High Agogo
45	Low Tom	68	Low Agogo
46	Open Hi-Hat	69	Cabasa
47	Low-Mid Tom	70	Maracas
48	Hi Mid Tom	71	Short Whistle
49	Crash Cymbal 1	72	Long Whistle
50	High Tom	73	Short Guiro
51	Ride Cymbal 1	74	Long Guiro
52	Chinese Cymbal	75	Claves
53	Ride Bell	76	Hi Wood Block
54	Tambourine	77	Low Wood Block
55	Splash Cymbal	78	Mute Cuica
56	Cowbell	79	Open Cuica
57	Crash Cymbal 2	80	Mute Triangle
		81	Open Triangle

Table 3.8: List of percussion instruments using the standard MIDI classification.

Using MT3, several features were extracted, which were not possible to extract before. While it was not possible to use MT3 as a melody extractor and test assess impact of using it in place of the current extractors (due to it not having information such as note velocity (i.e., intensity), amongst others), it still proved a valuable tool to extract features such as detecting the presence of an instrument, the amount of notes a certain instrument produces, amongst many others. A comprehensive list of features can be found in Section 5.2.

3.6 Music Source Separation

This section provides a brief overview of the current state of the art results in music source separation, as well as an overview of the two music source separation tools used in this work, Spleeter (Hennequin et al., 2020) and Demucs (Rouard et al., 2023).

3.6.1 State of the art results

As Spleeter was the framework used in (Panda, 2019) to separate the vocal stem of the audio tracks, Spleeter was once again re-used in this work in order to extract the same feature set on the new MERGE datasets and make a comparison to 4QAED. However, since the release of Spleeter, several other frameworks have appeared that have better results in source separation.

Musical source separation systems are typically evaluated according to four performance metrics (Vincent et al., 2006), namely:

- **Signal to Distortion Ratio (SDR):** Measures the overall quality of the separated source by taking into account both errors in filtering out other sources (interference) and artifacts introduced by the separation process.
- **Signal to Interference Ratio (SIR):** This specifically measures the amount of residual interference left from other sources in the separated source.
- **Signal to Artifacts Ratio (SAR):** This measures the amount of artifacts (or distortions) introduced by the separation process itself. It captures how much the algorithm has modified the original source during the separation, without accounting for interference from other sources.
- **Signal to Noise Ratio (SNR):** This measures the amount of residual noise in the separated source.

In the website papers with code, a list of the benchmarks for various datasets (e.g. MUSDB18 (Rafii et al., 2017), MedleyDB (Bittner et al., 2014), amongst others) using different approaches is compiled (pap, 2023). For example, for the MUSDB18 dataset, the best performer is the Sparse HT Demucs approach (Rouard et al., 2023), with an average Signal to Distortion Ratio (SDR) of 9.20.

Table 3.9 shows the top 10 approaches tested in the MUSDB18 dataset. Multiple Demucs approaches are in the top 10, making it the state of the art framework for source separation. Furthermore the Spleeter result is also included, in order to give a frame of reference.

With Demucs obtaining state of the art results in source separation, it was decided that an experiment would be conducted in this work pertaining to Spleeter and Demucs. This experiment consisted of taking the same set of features that was previously extracted (in (Panda, 2019)) using the vocal stem extracted from Spleeter, but extracting them on the vocal stem now separated with the best Demucs approach in order to see if any improvement on the classification results would happen by using Demucs.

Rank	Model	SDR (avg)	SDR (vocals)	SDR (drums)	SDR (bass)	SDR (other)	Paper
1	Sparse HT Demucs (fine tuned)	9.20	9.37	10.83	10.47	6.41	(Rouard et al., 2023)
2	Hybrid Transformer Demucs (f.t.)	9.00	9.20	10.08	9.78	6.42	(Rouard et al., 2023)
3	Band-Split RNN (semi-sup.)	8.97	10.47	10.15	8.16	7.08	(Luo and Yu, 2022)
4	TFC-TDF-UNet (v3)	8.34	9.59	8.44	8.45	6.86	(Kim et al., 2023)
5	Band-Split RNN	8.23	10.21	8.58	7.51	6.62	(Luo and Yu, 2022)
6	Hybrid Demucs	7.72	8.04	8.58	8.67	5.59	(Défossez, 2022)
7	KUIELab-MDX-Net	7.54	9.00	7.33	7.86	5.95	(Kim et al., 2021)
8	CDE-HTCN	6.89	7.37	7.33	7.92	4.92	(Hu et al., 2022)
9	Attentive-MultiResUNet	6.81	8.57	7.63	5.88	5.14	(Sgouros et al., 2022)
10	DEMUCS (extra)	6.79	7.29	7.58	7.60	4.69	(Défossez et al., 2021)
18	Spleeter (MWF)	5.91	6.86	6.71	5.51	4.02	(Hennequin et al., 2020)

Table 3.9: Top 10 source separation results in the MUSDB18 (Rafii et al., 2017) database, ranked by SDR. Spleeter (Hennequin et al., 2020) is also included as a frame of reference (pap, 2023).

3.6.2 Spleeter

Spleeter (Hennequin et al., 2020) is a source separation library that provides state of the of art models that allow for the separation of the audio signal into certain components, such as:

1. Vocals and Accompaniment (2 stems)
2. Vocals, drum, bass and the other components (4 stems)
3. Vocals, drum, bass, piano and the other components (5 stems)

Spleeter works by using a U-net, which is an encoder-decoder segmentation Convolutional Neural Network (CNN) with skip connections. The specific implementation used by Spleeter uses 12-layer U-nets (6 layers for encoding and 6 for decoding). A U-net is used to estimate a soft mask for each of the stems. Training loss is a L^1-norm, the sum of the absolute values of all the components in a vector, between masked input mix spectrograms and source-target spectrograms. The training was done using Deezer's [9] internal datasets. Finally, the separation is done from the estimated source spectrograms using soft masking or multi-channel Wiener filtering (Hennequin et al., 2020).

It has the advantage of being very fast when using a GPU device, as it is able to separate into the 4 respective stems 100 seconds of stereo sound in 1 second of processing time. The models were tested in the MUSDB18 dataset (Rafii et al., 2017), which is a dataset consisting of 150 full length musical tracks from different genres, as well as their isolated drums, bass, vocals and other stems.

In order to measure performance, four metrics from (Vincent et al., 2006) are used, which are Signal to Distortion Ratio (SDR), Signal to Interference Ratio (SIR), Signal to Artifacts Ratio (SAR), Signal to Noise Ratio (SNR). Comparative results to other frameworks can be found in Table 3.9. Is it noteworthy that all the top 10 approaches have come out since the release of Spleeter.

When compared to other state-of-the art frameworks, Spleeter obtained equivalent or better results than any existing implementation at the time.

Spleeter was used in (Panda, 2019) to separate the vocal track from the accompaniment and then extract an extra set of features from the separated vocal track, and was also used in this work in order to measure the impact of expanding the datasets, as the same set of features previously extracted in (Panda, 2019) had to be extracted.

3.6.3 Demucs

Demucs (Rouard et al., 2023), akin to Spleeter, is a music separation model, which is capable of separating drums, bass, vocals from the rest of the accompaniment. Demucs is based on a U-Net Convolutional architecture, much like Spleeter.

[9] https://deezer.io

There have been many versions of Demucs throughout the years, and at the time of writing the most recent and the best performing one is called Demucs v4, where a hybrid spectrogram/waveform separation model is used using Transformers.

This version is based on Hybrid Demucs (Demucs v3), which uses two U-nets, one in the time domain (with temporal convolutions) and another in the spectrogram domain (with convolutions over the frequency axis). Each of the U-nets has 5 encoder layers and 5 decoder layers. The output of the spectral branch is transformed to a waveform using the Inverse Short-Time Fourier Transform (ISFTF), before being added with the output of the temporal branch, and this sum gives the actual prediction of the model (Rouard et al., 2023).

Demucs v4 replaces the innermost convolutional layers with cross-domain transformer encoders, that uses self-attention and cross-attention to process spectral and temporal informations. This allows treating in parallel the 2D signal from the spectral branch and the 1D signal from the waveform branch. In Demucs v3, precise tuning of the model parameters had to be conducted in order to properly align the time and spectral representation (Short-Time Fourier Transform (STFT) window, hop length, hop size, amongst others). On the other hand, the cross-domain transformer encoder approach is able to work with heterogeneous data shapes, making it a much more flexible architecture when compared to Demucs v3 (Rouard et al., 2023). Figure 3.7 shows the Hybrid Transformer Demucs Architecture.

This model was trained the MUSDB18 (Rafii et al., 2017), as well as an extra private dataset of 800 songs with stems from 200 artists from diverse music genres. Pre-processing was done in order to remove ambiguous songs and songs that had more than 30% silence.

It was then also tested in MUSDB18, in order to facilitate the comparison with other state of the art models. Using Signal to Distortion Ratio (SDR) as the metric, it obtained better results when compared to other state of the art approaches in all stems except "other" and "vocals". This means that it is better at separating drums, bass and the overall stems than any of the state of the art models, including Spleeter. Table 3.9 shows the results obtained in the MUSDB18 dataset when compared to other state of the art approaches.

Due to this model being the state of the art model for source separation, it was the model chosen to do the separation of the percussion track from our expanded datasets, in order to enable the creation for features designed solely for percussion. Furthermore, since Demucs also has the capability to separate the vocal track and since it has obtained better results when compared to Spleeter, the previous features extracted from the vocal component with Spleeter were extracted with the Demucs vocal track in order to compare the two approaches.

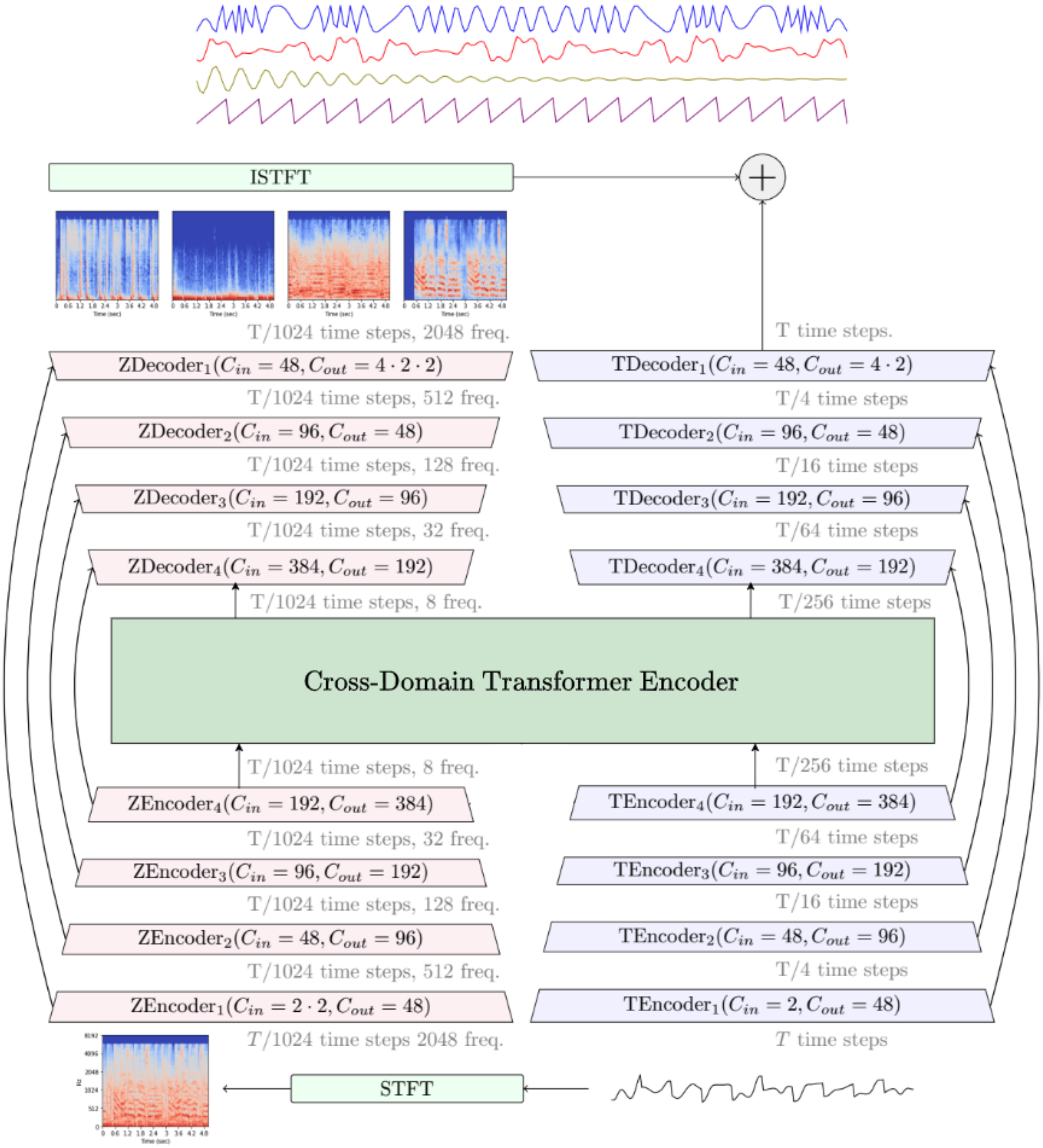

ISTFT
T/1024 time steps, 2048 freq.
T time steps.
ZDecoder$_1(C_{in} = 48, C_{out} = 4 \cdot 2 \cdot 2)$
TDecoder$_1(C_{in} = 48, C_{out} = 4 \cdot 2)$
T/1024 time steps, 512 freq.
T/4 time steps
ZDecoder$_2(C_{in} = 96, C_{out} = 48)$
TDecoder$_2(C_{in} = 96, C_{out} = 48)$
T/1024 time steps, 128 freq.
T/16 time steps
ZDecoder$_3(C_{in} = 192, C_{out} = 96)$
TDecoder$_3(C_{in} = 192, C_{out} = 96)$
T/1024 time steps, 32 freq.
T/64 time steps
ZDecoder$_4(C_{in} = 384, C_{out} = 192)$
TDecoder$_4(C_{in} = 384, C_{out} = 192)$
T/1024 time steps, 8 freq.
T/256 time steps
Cross-Domain Transformer Encoder
T/1024 time steps, 8 freq.
T/256 time steps
ZEncoder$_4(C_{in} = 192, C_{out} = 384)$
TEncoder$_4(C_{in} = 192, C_{out} = 384)$
T/1024 time steps, 32 freq.
T/64 time steps
ZEncoder$_3(C_{in} = 96, C_{out} = 192)$
TEncoder$_3(C_{in} = 96, C_{out} = 192)$
T/1024 time steps, 128 freq.
T/16 time steps
ZEncoder$_2(C_{in} = 48, C_{out} = 96)$
TEncoder$_2(C_{in} = 48, C_{out} = 96)$
T/1024 time steps, 512 freq.
T/4 time steps
ZEncoder$_1(C_{in} = 2 \cdot 2, C_{out} = 48)$
TEncoder$_1(C_{in} = 2, C_{out} = 48)$
T/1024 time steps 2048 freq.
T time steps
STFT

Figure 3.7: Hybrid Transformer Demucs Architecture. (Rouard et al., 2023).

This page is intentionally left blank.

Chapter 4

Datasets

In this chapter, a discussion about the work currently done in the expansion of the datasets will be presented, as well as a overview of two frameworks that will be used in the next phase of this work.

4.1 Expansion of current datasets

4.1.1 Previously conducted expansions

In previous works by our group, such as (Malheiro, 2017), (Malheiro et al., 2018) and (Panda, 2019), different databases were created for the respective works. These datasets have also been rectified since their publication, with the removal of a few songs that were repeated or were not valid (for example, songs that were not totally in English).

Previous work has also been done to create a large bi-modal (audio and lyrics) dataset. This work encompassed taking the datasets that only had one specific type, for example, an audio dataset, acquiring the lyrics of the corresponding songs from various websites such as Genius or lyrics.com, and then proceeding with various annotation tasks. The same work was also done but in reverse, as in having an already existing lyric dataset and grabbing the audio files from the AllMusic platform. Moreover, there was already a small bi-modal dataset of 133 songs created in (Malheiro, 2017) for the respective work.

This expansion method raised a few problems that ultimately led to the need of the expansion of the already existing datasets:

1. Since for the annotations of the dataset AllMusic tags are used, songs that are not listed on AllMusic or that do not have any tags can not be considered for the dataset.
2. Any song that is not totally in English, is an instrumental or does not have lyrics available (old songs, mostly from the 1940's until the 1960's) can not be considered for the dataset.

3. Since this is a bi-modal dataset, if the songs do not have matching quadrants in the Russell model in the audio and the lyrical annotations, the song is discarded, however, they can still be kept in the corresponding dataset (audio or lyrical).

In Table 4.1 the amount of songs present in the dataset prior to the expansion done for this work are shown, as well as their origin.

Dataset of origin	**Q1**	**Q2**	**Q3**	**Q4**	**Total**	**Amount of songs in the original dataset**
(Malheiro, 2017)	37	37	30	29	133	133
4QAED (Panda, 2019)	100	129	103	72	404	900
(Malheiro et al., 2018)	160	170	146	100	576	771
Total	297	336	279	201	1113	

Table 4.1: Amount of songs in the previous bi-modal dataset.

As we can see, due to the aforementioned problems, a lot of songs ended up being removed from the bi-modal dataset, and with the goal of having 2000 bi-modal songs, a new expansion had to be conducted.

Tables 4.2 and 4.3 show the amount of songs in the previous audio and lyrical datasets respectively.

Dataset of origin	**Q1**	**Q2**	**Q3**	**Q4**	**Total**	**Amount of songs in the original dataset**
(Malheiro, 2017)	51	45	30	34	160	133
4QAED (Panda, 2019)	223	225	221	224	893	900
(Malheiro et al., 2018)	160	170	146	100	576	771
Total	434	440	397	358	1629	

Table 4.2: Amount of songs in the previous audio dataset.

Dataset of origin	**Q1**	**Q2**	**Q3**	**Q4**	**Total**	**Amount of songs in the original dataset**
(Malheiro, 2017)	44	41	51	44	180	133
4QAED (Panda, 2019)	100	129	103	72	404	900
(Malheiro et al., 2018)	207	203	204	148	762	771
Total	351	373	358	264	1346	

Table 4.3: Amount of songs in the previous lyrical dataset.

4.1.2 Expansions done in the scope of this work

To conduct this expansion in an organized manner, an updated version of a previous algorithm used by (Panda, 2019) was used. The algorithm used remains almost the same, with only changing a step in order to decrease ambiguity between quadrants. Before, the emotional tags for each song were extracted from

AllMusic and their arousal and valence values were calculated using the Warriner framework (Warriner et al., 2013). Then the arousal and valence values were converted into one of the four quadrants of the Russell model of emotion. The quadrant where the most tags fell upon was chosen as the quadrant for the song, which was then manually validated.

Now, after calculating the arousal and valence values using the Warriner framework, songs that have valence or arousal values between -0.2 and 0.2 (meaning they are close to the center of the plane) are removed.

Figure 4.1 shows the output of this step on this expansion, with songs that were not considered going forward colored in red while songs that were kept are colored in green, as well as a red square where the songs that are in it are removed for easier visualization.

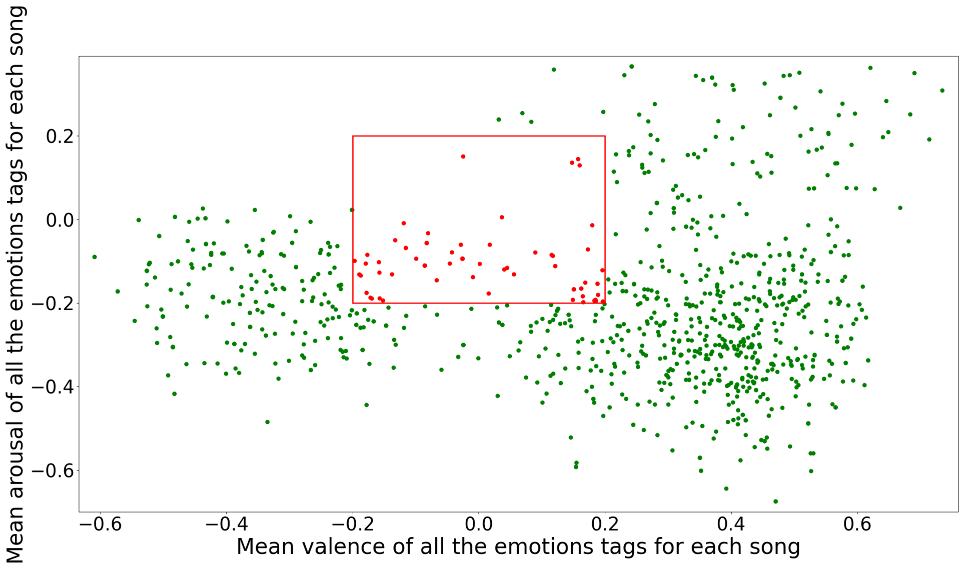


Figure 4.1: Russell's plane with the output of the first step of the expansion.

Thus, the algorithm used to expand the dataset can be summarized as such (see (Panda, 2019) for further details):

1. Gather songs and emotion data from AllMusic services.

   1.1. Retrieve the list of 289 emotion tags, E, using the AllMusic API.

   1.2. For each emotion tag gathered, E_i, query the API for the top 10000 songs related with it, S_i.

2. Bridge the emotional data from AllMusic, based on an unvalidated emotional taxonomy, with Warriner's list.

   2.1. For each emotion tag, E_i, retrieve the associated AVD (arousal, valence and dominance) values from the Warriner's dictionary of English words. If the word is missing, remove it from the set of tags, E.

2.2. Map each emotion tag, E_i, onto one of the four qudrants of the Russell model of emotion using the AV values.

2.3. Attribute a quadrant to each song, S_i, based on the quadrant where the majority of the emotion tags, E_i, fall upon.

3. Data processing and filtering, to reduce the massive amount of gathered data to a more balanced but still sizeable set.

3.1. Filter ambiguous songs, where a dominant emotional quadrant is not present.

3.1.1. For all the songs in the set of songs S_i, calculate the average arousal and valence values of all the emotion tags gathered, E_i, and if the average value of valence or arousal is contained in the range $[-0.2, 0.2]$ remove the song from the dataset.

3.2. Remove duplicated or very similar versions of the same songs by the same artists (e.g., different albums) by using approximate string matching against the combination of artist and title metadata.

3.3. Remove songs without genre information. This ensures that the algorithms that ensure maximum genre diversity can function correctly.

3.4. Remove songs that do not have lyrics (instrumentals) or songs that do not have available lyrics.

4. Generate a subset dataset maximizing genre variability in each quadrant.

5. Manually validate the audio dataset.

5.1. Distribute all the songs in the set S_i for each of the members of the team in an equal manner.

5.2. For each song, perform validation and annotation of the song according to Russell's model of emotion.

5.2.1. Verify that the song is valid (e.g. does not contain clapping or silence) and that the emotion present in the song is not ambiguous.

5.2.2. If the quadrant annotated does not match with the quadrant calculated in step 2.2., remove the song from the bi-modal dataset, else, keep the song.

6. Obtain the lyrics corresponding to the acquired audio clips from platforms such as lyrics.com, ChartLyrics, MaxiLyrics or MusixMatch.

7. Manually validate the obtained song lyrics.

7.1. Distribute all the songs in the set S_i for each of the members of the team in an equal manner.

7.2. For each song, perform validation and annotation of the song according to Russell's model of emotion.

7.2.1. Verify that the lyrical file is well structured, belongs to the correct audio clip, and that the emotion present in the file is not ambiguous.

7.2.2. If the quadrant annotated does not match with the quadrant calculated in step 2.2., remove the song from the bi-modal dataset and add it only to the audio dataset. If it matches with the calculated quadrant, add to both the audio and bi-modal dataset.

This approach, while still involving manual effort, is much lighter than manually annotating every song. Since the base annotations are generated from AllMusic tags which were created by experts, it is not required that there are multiple annotators for each song. Thus, for each song (both audio and lyrics) only one person is require to annotate and validate a song.

The audio annotations were done by 5 members of our group, while 8 members participated in the lyrical annotations. The final datasets are referred to as Music Emotion Recognition - Next Generation (MERGE) followed by the type (audio, lyrics, and bi-modal), and finally by "complete" and "balanced", in order to indicate which type of dataset it is.

The tables below illustrate the final numbers after the expansion for the bi-modal, audio and lyrical datasets, respectively.

Dataset of origin	**Q1**	**Q2**	**Q3**	**Q4**	**Total**	**Amount of songs in the original dataset**
(Malheiro, 2017)	37	37	30	29	133	133
4AQED (Panda, 2019)	100	129	103	72	404	900
(Malheiro et al., 2018)	160	170	146	100	576	771
Current expansion	228	337	221	317	1115	
MERGE_Bimodal_Complete	525	673	500	518	2216	

Table 4.4: Amount of songs in the bi-modal dataset after the expansion.

Dataset of origin	**Q1**	**Q2**	**Q3**	**Q4**	**Total**	**Amount of songs in the original dataset**
(Malheiro, 2017)	51	45	30	34	160	133
4QAED (Panda, 2019)	223	225	221	224	893	900
(Malheiro et al., 2018)	160	170	146	100	576	771
Current expansion	441	475	411	598	1925	
MERGE_Audio_Complete	875	915	808	956	3554	

Table 4.5: Amount of songs in the audio dataset after the final expansion.

With these expansions, the goal set for this work was reached and even surpassed. One important factor to consider is that due to the nature of how the datasets were built straight from the start, with considerations in mind such as balancing between quadrants, maximizing genre presence, it allowed for the creation of bigger datasets while keeping a high standard of quality.

Another important focus of this work was increasing the size of datasets while keeping the question of quadrant balancing in mind, as having an imbalanced

Dataset of origin	Q1	Q2	Q3	Q4	Total	Amount of songs in the original dataset
(Malheiro, 2017)	44	41	51	44	180	133
4QAED (Panda, 2019)	100	129	103	72	404	900
(Malheiro et al., 2018)	207	203	204	148	762	771
Current expansion	249	337	263	373	1222	
MERGE_Lyrics_Complete	600	710	621	637	2568	

Table 4.6: Amount of songs in the lyrical dataset after the final expansion.

can lead to the model becoming biased towards the majority class, thus failing to adequately learn the minority class (He and Garcia, 2009).

Thus, Algorithm 4.2 of (Panda, 2019) (See Section 4.1.4 of (Panda, 2019) for further details) was used to create a balanced set of songs for each quadrant. This algorithm was devised to maintain maximum genre diversity in a new set of songs. This led to the creation of datasets with 600 songs per quadrant for the lyrical experiments and more than 800 per quadrant for the audio experiments which is something that has not been possible before in the MER field. For reference, the dataset used in (Panda, 2019), used 225 songs per quadrant totalling 900 songs.

Furthermore, the bi-modal dataset has also been expanded greatly, now totalling 2216 songs. Moreover, the balanced version of this dataset has 2000 songs in total, with 500 songs per quadrant. Having a dataset of this size with both the audio and lyrics components will allow for new bi-modal approaches to be tested, mostly deep-learning techniques. These techniques were not feasible before either due to the small dataset sizes available or due to their imbalance in the amount of songs per quadrant, with these two factors being major factors taken into consideration in the construction of this dataset.

From this expansion, three datasets were created: MERGE_Audio_Complete (the full audio dataset), MERGE_Lyrics_Complete (the full lyrics dataset) and MERGE-_Bimodal_Complete (the full bi-modal dataset). Furthermore, a balanced version of these datasets was also created (meaning the number of songs per quadrant is equal). These datasets can be referred to as MERGE_Audio_Balanced, MERGE_Lyrics_Balanced and MERGE_Bimodal_Balanced.

Table 4.7 shows the total amount of songs for each of the six aforementioned datasets as well as the amount of songs per quadrant.

Dataset Name	Q1	Q2	Q3	Q4	Total
MERGE_Audio_Complete	875	915	808	956	3554
MERGE_Audio_Balanced	808	808	808	808	3232
MERGE_Lyrics_Complete	600	710	621	637	2568
MERGE_Lyrics_Balanced	600	600	600	600	2400
MERGE_Bimodal_Complete	525	673	500	518	2216
MERGE_Bimodal_Balanced	500	500	500	500	2000

Table 4.7: Number of songs in the novel MERGE datasets.

This page is intentionally left blank.

Chapter 5

Methodology

In this chapter, an overview is provided of the methodology used in the whole pipeline, from feature extraction to the final classification. Figure 5.1 provides an overview of the whole pipeline, including the new additions done to the baseline work.

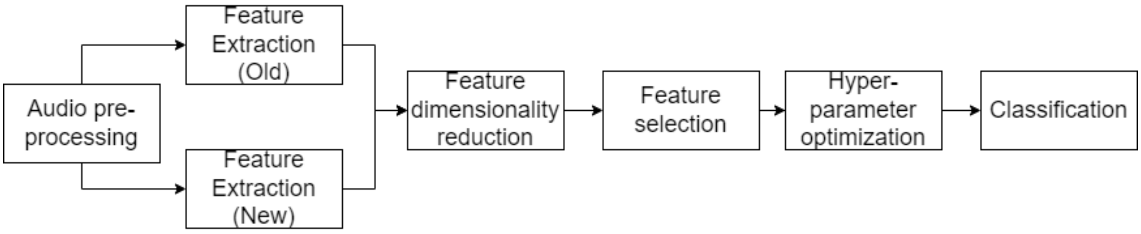


Figure 5.1: Overview of the whole pipeline for emotion classification.

The overall pipeline for this work can be summarized as:

1. Standardize all the audio clips into a certain format - WAV PCM Format, 22050 HZ sampling rate, 16 bits quantization and mono-aural.
2. Extract all the baseline features using the various frameworks - MIR Toolbox (Lartillot, 2018), Marsyas (Tzanetakis, 2002) and PsySound3 (Cabrera et al., 2008).
3. Extract all the newly created features using MT3 and Demucs.
4. Reduce the feature dimensionality.
    4.1. Remove features where no variation was found in all the extracted values.
    4.2. Remove highly correlated pairs of features.
5. Perform feature selection.

5.1. Use the RelieF (Robnik-Sikonja and Kononenko, 1997) algorithm to compute the weight of features and rank them based on their weight and select a set of the top features.

6. Perform SVM hyper-parameter optimization using a Bayesian search.

7. Perform SVM classification with the four quadrants from the Russell model of emotion and extract metrics such as overall F1-Score.

5.1 Preliminary steps

As this work has (Panda, 2019) as basis, the employed methods are based upon the original methods. In terms of feature extraction, there are a few steps that have to be done to extract what is called the baseline features, meaning the features used in (Panda, 2019).

Firstly, the songs are converted onto a standardized audio format using the FFmpeg framework [1], with the following specifications: WAV PCM Format, 22050 HZ sampling rate, 16 bits quantization and mono-aural. From here forward, all the feature extraction steps and anything that uses the audio file is done using these standardized versions.

Afterward this standardization is done, the stem separation of vocal track and the remaining instrumental is done using Spleeter (Hennequin et al., 2020). Furthermore, once again, these audios are standardized to the aforementioned format.

The first step is to extract the melodic lines using the MELODIA framework (Salamon and Gómez, 2012) and the Dressler framework (Dressler, 2016). This is done for the full audio and also for the vocal only component.

After all of this is done, feature extraction can begin.

5.2 Feature engineering

The following section provides an overview of the whole feature engineering process, containing the extracting of both the baseline and newly proposed features, as well as methods used for feature dimensionality reduction and feature selection.

5.2.1 Features obtained from the MIDI file

Using MT3, the MIDI files with the notes for instrument were extracted. As MT3 was trained on files with no vocal track, it was needed that for the input an audio track with no vocal track was given. While separating the vocal track, Demucs

[1] https://ffmpeg.org

also outputs a file where the vocal track is removed from the rest of the track, so those were files used as input for MT3.

In (Panda, 2019), a certain set of features were extracted. These features were described in detail in Section 3.2. In (Panda, 2019) the transcription was only done in the melodic channel of the audio track, and now, using MT3, the transcription of the whole audio track is done. This enables the analysis for all the separated melodic lines of a single track.

One example of this is the music layer information (explained in further detail in Section 3.2.8). Previously, this information was extracted by estimating the number of fundamental frequency (F0) for each frame of the audio track. Now, using the MIDI file outputted by MT3, we are able to estimate the layer information by using the number of instruments playing in each frame. Furthermore, information regarding note duration was also able to be extracted for each instrument, which was something that was not able to be extracted before.

Besides the updated features, we propose the following novel features:

1. **Instrument extent** - Before, it was impossible to measure the presence or not of a certain instrument in an audio track. Now, with the separation done by MT3 this is possible. To encapsulate this, for each of the 128 instruments, a feature was created that has the value of 1 if any note of that instrument is played throughout the audio track and 0 if not. Furthermore, for each of the five instrument groups, the total number of instruments of each specific group that is present in the song is calculated. Finally, the total number of melodic instruments present is also calculated, as well as the amount of drum instruments.

2. **Instrument notes** - Another thing that was not possible before was being able to measure the amount of notes that each instrument plays in a song. Thus, a feature was created that represents the amount of notes that a specific instrument has in a song. The total amount of melodic notes, percussion notes and total amount of notes for each of the five instrument groups is also calculated.

3. **Instrument duration percentage** - As previously mentioned, information regarding note duration is calculated. However one new feature that is calculated for each instrument is the percentage of the song that contains that instrument. For this, the sum of all the duration of all the notes of a certain instrument is calculated. Then, that amount is divided by the length of the song. Once again, this process was repeated for each of the five groups, as well as for the group of percussion instruments and the group of melodic instruments.

5.2.2 Features obtained from the percussion audio track

Using Demucs, the percussion component was separated from the rest of the track, leaving us with two audio tracks: only the percussion component, and the rest of the track. As such, new features were created extracted from the percussion track only, in order to represent the percussion track. These features were extracted using Python, using the librosa [2] and scipy packages.

Prior to any of the features being extracted, the audio is loaded using the *load* function from the librosa package, which returns the audio time series in a form of an array, with that time series being used as the base for the extraction of the percussion information. In all the features that involved having to pick a certain frame length and hop size, the values of 1024 for the frame length and 128 for the hop size were chosen, in order to maintain consistency with previously extracted features in (Panda, 2019).

For the features that involved having to obtain statistics (e.g. volume information), 6 statistics were extracted: mean, standard deviation, skewness, kurtosis, maximum and minimum. This was done in order to have consistency with (Panda, 2019).

In the following the novel proposed features are described:

1. **Drum Extent Percentage** - In order to understand the amount of percussion in a track, this feature aims to represent the amount of drums present in the percussion track. First, the Root-Mean Square (RMS) for each frame is computed, with RMS being used as a measure of the magnitude of the audio signal, representing the energy of said signal. The RMS can be defined as:

   $$\text{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2} \tag{5.1}$$

   where N is the number of samples in the frame, and x_i is the i^{th} sample value.

   After, the number of frames that are above a certain threshold is calculated. The ratio between this number of frames and total amount of frames in the song gives us the drum extent of the audio track. This threshold was defined experimentally as 0.025.

   To choose this threshold, 20 songs that were near silent to the listener were chosen (5 from each quadrant), and 20 songs that had a high extent of drums were also picked. Then, different thresholds were tested until the drum extent calculated accurately represented the amount of drum that is audible to the listener of the track.

   It was chosen to not use decibels because using that approach led to situations where the track was near silent to the human ear but still had high values of drum extent.

2. **Amplitude and intensity information** - Using the audio time series, amplitude and intensity was calculated. First, we start by taking the absolute

[2] Librosa is a python package for music and audio analysis. Url: https://librosa.org/doc/latest/index.html.

value of the audio time series, to get the amplitude of the signal. Then using this amplitude, the six aforementioned statistics were computed.

For intensity, the frame based intensity is calculated, using the same method for drum extent explained above. The usual six statistics were also computed.

3. **Spectral features** - A set of spectral features was extracted from the signal. For the features that return an array with information for each frame, the same usual six statistics are computed. The spectral features extracted are:

   (a) **Spectral centroid** - For each frame of the audio time series, the frame is normalized and treated as a distribution over frequency bins, from which the main (the centroid) is extracted per frame.

   The centroid at frame *t* can be defined as (Klapuri and Davy, 2006):

   $$\text{Centroid}[t] = \frac{\sum_k S[k,t] \times \text{freq}[k]}{\sum_j S[j,t]}$$

   where *S* is a magnitude spectrogram, and *freq* is the array of frequencies (e.g., FFT frequencies in Hz) of the rows of *S*.

   (b) **Spectral bandwidth** - Bandwidth is the difference between the upper and lower frequencies in a continuous band of frequencies.

   The spectral bandwith at frame *t* can defined as (Klapuri and Davy, 2006):

   $$\text{Spectral Bandwidth}[t] = \left(\sum_k S[k,t] \times (\text{freq}[k,t] - \text{Centroid}[t])^p \right)^{\frac{1}{p}}$$

   (c) **Spectral contrast** - First, the spectrogram is computed if none is provided. After, each frame of the spectrogram *S*, is divided into sub-bands. For each sub-band, the energy contrast is estimated by comparing the average energy in the top quartile (peak energy) to the one in the bottom quartile (valley energy). Having high energy contrast values generally correspond to clear, narrow-band signals, while low contrast values correspond to broad-band noise. (Jiang et al., 2002b)

   The function returns an array where each row of spectral contrast values corresponds to a octave-based frequency. Afterwards, the six usual statistics are extracted from this array.

   (d) **Spectral flatness** - Spectral flatness (also referred as tonality coefficient) is a measure to quantify how much noise-like a sound is, as opposed to being tone-like (Dubnov, 2004). Having a high spectral flatness (closer to 1) indicates the spectrum is similar to white noise. The formula for spectral flatness can be summarized as (Dubnov, 2004):

   $$\text{SFM} = \frac{\left(\prod_{n=0}^{N-1} X[n] \right)^{1/N}}{\frac{1}{N} \sum_{n=0}^{N-1} X[n]}$$

   Where:

- $X[n]$ is the power spectrum (or magnitude squared) at frequency bin n.
- N is the number of frequency bins.
- $\prod$ is the product over all n.
- $\sum$ is the sum over all n.

(e) **Spectral rolloff** - The roll-off frequency is defined for each frame as the center frequency for a spectrogram bin such that at least 85% percent (by default) of the energy of the spectrum in this frame is contained in this bin and the bins below (Tzanetakis, 2002). The exact percentage varies between authors but the majority of authors use 85%. The formula can be summarized as (Tzanetakis, 2002):

$$R = f_i \quad \text{where} \quad \sum_{n=0}^{i} X[n] \geq \alpha \sum_{n=0}^{N-1} X[n]$$

Where:

- R is the spectral rolloff frequency.
- f_i is the frequency corresponding to the i-th bin.
- $X[n]$ is the power or magnitude at frequency bin n.
- N is the total number of frequency bins.
- α is a threshold (e.g., 0.85)
- $\sum$ is the sum over all n.

(f) **Spectral entropy** - Spectral entropy is a measure that is used to characterize the complexity or randomness of a signal in the frequency domain. A lower entropy value suggests a less complex signal while higher values suggest a more complex signal.

First, the Power Spectral Density (PSD) is calculated. The PSD is given by calculating the Short-Time Fourier Transform (STFT) of the audio signal. Afterwards, the PSD values are normalized across frequency bins for each frame, which ends up turning the values into probabilities. This also makes it so that the sum of the values in all the values equals 1, which is required for entropy calculation to be meaningful. The spectral entropy is calculated for each frame using the formula for Shannon entropy.

This entropy calculation results in an array of entropy values, one for each frame. Then, the six usual statistics are computed.

4. **Self-Similarity Matrix (SSM) features** - Using Mel-Frequency Cepstral Coefficients (MFCC) derived from the audio signal, a set of features that aim to capture information regarding the similarity between each pair of frames in the audio signal are computed. First, the Mel-Frequency Cepstral Coefficients (MFCC) are calculated, using the same hop size of 128. This will produce a feature matrix that is used to generate the SSM. Then, the SSM is computed using cosine similarity as the distance metric (Tzanetakis, 2002). Finally, the SSM is thresholded to create a binary matrix:

$$\text{Thresholded\_SSM} = \begin{cases} 1 & \text{if SSM} > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

This SSM serves as a base from which the following features are proposed and extracted:

(a) **Total number of patterns** - Corresponds to the total number of patterns where there are similarities.

(b) **Pattern duration** - An array of pattern durations is computed, where all the durations of the patterns that are repeated are kept. From this array, the six normal statistics are computed.

(c) **Total number of repetitions per second** - The total number of repetitions is divided by the total number of seconds in the song in order to get the total number of repetitions per second.

As such, the process used to extract the new proposed features can be summarized as:

1. Separate the vocal track and percussion track using Demucs (this will in turn also create the non-vocal and non-percussion tracks).

   1.1. Standardize the non-vocal track, the vocal track, the percussion track, and the non-percussion track (called the melodic track) to the aforementioned specifications.

2. Extract the proposed percussion features from the standardized percussion track audio.

3. Extract the MIDI files using MT3 using the standardized non-vocal track.

   3.1. Extract all the proposed features related to the MIDI file.

5.2.3 Baseline feature extraction

A summary of the features extracted for the baseline approach can be found in Section 3.2. These features are extracted using various audio frameworks, such as MIR Toolbox (Lartillot, 2018), Marsyas (Tzanetakis, 2002) and PsySound3 (Cabrera et al., 2008). The process of extracting these baseline features is not the main focus on this work, and as such it will not be further detailed.

5.2.4 Feature dimensionality reduction

After this feature extraction is complete, there are over 4500 features. Since this feature set will more than likely contain features that represent duplicate information or that are heavily correlated, feature dimensionality reduction is used in order to reduce the amount of features used by the classifier.

In order to reduce the feature dimensionality, first, features where the standard deviation of the observed data is zero are removed. Secondly, features that are heavily correlated were removed. In order to accomplish this, the features were ranked based on their importance using the RelieF algorithm (Robnik-Sikonja and Kononenko, 1997). Afterwards, the outliers for each feature were computed. Excluding these outliers, the correlation between every pair of features is computed. If the correlation factor between two features is greater than a threshold (set experimentally at 0.9), then the feature is removed.

After this step, from the original set of around 4500 features, only about 3500 remained.

5.2.5 Feature selection

For feature selection, once again the RelieF algorithm was used to order features based on their importance. Here is where the methodology from this work diverges from the one used in (Panda, 2019). We take each set of top X features ranked by the RelieF algorithm, where X is an integer that can have the values 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200, 300, 400, 500, 600, 700, 800, 900. Afterwards, perform hyper parameter optimization on this set of features. This ensures that for each of the datasets, the best possible F1-Score is obtained, as hyper parameter optimization is done separately for each set of top features.

5.3 Hyper-parameter optimization

In order to achieve optimal performance, a Bayesian search was performed to find the best hyper-parameters for the SVM, which included searching for the optimal kernel type, as well as the optimal gamma, cost, and the degree of the polynomial kernel (where applicable). For the cost and gamma values, the search range consisted of values between 1e-6 and 100, and 1 through 5 for the degree of the kernel. Finally, for the kernel type, Linear, Polynomial, Radial Basis Function (RBF) and Sigmoid were the options considered (Brownlee, 2019).

To find these hyper-parameters, repeated stratified 10-fold cross validation (Duda et al., 2001) was used (10 repetitions were applied), since, according to the literature, “there are more performance estimates, and the training set size is closer to the full data size, thus increasing the possibility that any conclusion made about the learning algorithm(s) under test will generalize to the case where all the data is used to train the learning model” (Refaeilzadeh et al., 2009).

The Bayesian approach was chosen in detriment of the standard grid search approach used in (Panda, 2019) as it achieved either equal or comparable results when compared to the grid search method while taking a fraction of the time to find the an optimal set of parameters.

5.4 Classification and evaluation metrics

For classification, an SVM was used as that is also the method used in (Panda, 2019). The SVM parameters used are the ones found in the previous step, and then there are several metrics reported.

The first metric calculated is the average F1-Score obtained overall for all the folds, and also for each quadrant. The standard deviation overall and for each quadrant is also calculated.

The second metric calculated is the confusion matrix, but in percentage instead of whole values. This means that for each fold and for each combination of true and predicted values, the percentage of the correctly classified values for each quadrant is calculated. Then, the average percentage for each cell of the confusion matrix throughout all folds is calculated, as well as the standard deviation.

This provides a better understanding of the amount of songs that are being classified correctly throughout all folds.

After training and assessing the model, the postulated hypotheses are either confirmed or dis-proven based on inferential analysis of the F1-Scores determined for each fold for the approach with baseline features and approaches with new feature sets. This validation process involves a significance test, with a 5% confidence interval. If the p-value, which is the outcome of the significance test, falls below this interval, then the findings are considered statistically significant, and the hypothesis is affirmed.

This page is intentionally left blank.

Chapter 6

Results and discussion

In this chapter, an overview of the results obtained for all the experiments done will be presented. In the scope of this work many extra experiments were done, such as for example replacing the original features extracted from the vocal track from Spleeter with the features extracted with the vocal track from Demucs in order to measure the impact of using one framework versus the other.

6.1 Comparison with 4QAED

With the large increase in size in the three datasets (audio, lyrical and bi-modal), it is paramount to replicate previously done experiments but applied to the bigger datasets, in order to measure the expansion impact in the emotion classification results. Only the audio and bi-modal dataset pertain to the scope of this work so those were the two datasets tested.

Previously in (Panda, 2019), the best results obtained were an F1-score of 76.4%, obtained using a SVM classifier approach using the top 100 features in the 4QAED dataset, containing 900 songs. As (Panda, 2019) serves as foundation to this work, it is paramount that the same approach is applied to the newly constructed datasets in order to better measure the impact of the dataset expansions.

An explanation of the whole pipeline for emotion classification into the four quadrants of the Russell model of emotion can be found in Chapter 5.

In order to test the impact of the expansion, the two main datasets considered are MERGE_Audio_Complete, totalling 3554 songs, and MERGE_Bimodal_Complete (only the audio component was considered), totalling 2216 songs. Furthermore, the balanced by quadrant versions of the two aforementioned datasets were also tested (MERGE_Audio_Balanced and MERGE_Bimodal_Balanced), totalling 3232 and 2000 songs, respectively.

The top 100 features for each dataset were used in the classification, in order to obtain a comparison with the 4QAED dataset, as this was also the method used in (Panda, 2019).

6.1.1 Results

It is noteworthy that in (Panda, 2019) an overall F1-score of 76.4% was achieved, but this was not able to be replicated using the newly extracted features. The set of features extracted in (Panda, 2019) were extracted in certain circumstances (e.g. certain Matlab or framework versions) that have been updated throughout the years. This led to a result of 72.8% overall F1-score for 4QAED instead of the 76.4% obtained earlier. As this environment is the same one where the features for the novel datasets are extracted, the comparisons done with 4QAED are done with the newly obtained F1-score.

Table 6.1 shows the results obtained on the new datasets using the aforementioned algorithm, using the weighted F1-score metric. Furthermore, Table 6.2 shows the weighted F1-score metric obtained for each quadrant for the datasets.

Dataset Name	Overall F1-Score
4QAED	72.7 ± 4.0
MERGE_Audio_Complete	71.0 ± 2.3
MERGE_Audio_Balanced	70.9 ± 2.3
MERGE_Bimodal_Complete	71.0 ± 2.6
MERGE_Bimodal_Balanced	71.0 ± 2.8

Table 6.1: Results obtained for all the tested datasets using the MERGE_Panda feature set.

Dataset Name	F1 Q1	F1 Q2	F1 Q3	F1 Q4
4QAED	74.4 ± 6.5	84.5 ± 5.9	65.9 ± 7.1	67.2 ± 7.3
MERGE_Audio_Complete	73.2 ± 3.2	86.6 ± 2.5	58.8 ± 4.1	63.9 ± 3.9
MERGE_Audio_Balanced	74.3 ± 3.1	86.4 ± 2.5	62.2 ± 4.7	60.9 ± 4.3
MERGE_Bimodal_Complete	73.4 ± 3.6	89.4 ± 2.7	58.4 ± 5.1	57.0 ± 5.1
MERGE_Bimodal_Balanced	74.4 ± 4.0	88.1 ± 3.1	63.1 ± 4.9	58.6 ± 5.4

Table 6.2: F1-score obtained for each quadrant for all the tested datasets using the MERGE_Panda feature set.

The results show a drop between 2% and 3% in F1-Score for the various datasets. Looking at the F1-Score per quadrant, we can see that the fourth and the third quadrant are the ones where the results drop, with the other two quadrants achieving equal or better results when compared to 4QAED. This can partly be explained in the complete datasets due to the dataset imbalance, for example, in MERGE_Bimodal_Complete, there are 673 2nd quadrant songs compared to only 500 3rd quadrant songs.

Furthermore, having a F1-score of 71% with a much larger dataset size proves that the datasets constructed are still robust and can be used for future experiments. As previously mentioned, having a dataset of this quality with almost 4 times the size of previous dataset is very important for deep learning experiments, where the amount of data required to get better results is very important.

Finally, it is also important to look at the confusion matrix, in order to see which of the quadrants is most affected by the expansion.

	Predicted			
True	**Q1**	**Q2**	**Q3**	**Q4**
	76.32 ± 5.22	10.45 ± 4.36	4.73 ± 3.02	8.50 ± 3.43
	8.70 ± 3.36	89.51 ± 3.44	1.16 ± 1.23	0.64 ± 1.01
	7.86 ± 3.41	2.76 ± 2.22	56.78 ± 6.36	32.60 ± 5.85
	13.15 ± 4.71	0.68 ± 1.10	30.03 ± 6.48	56.15 ± 6.90

Table 6.3: Confusion matrix for MERGE_Bimodal_Complete using the MERGE_Panda feature set.

As Table 6.3 shows, there is a lot of confusion between the third and fourth quadrant, which was also one of the problems found in (Panda, 2019). In sum, these results show that the expansion was successful, even though the F1-score decreased. The outcome of the expansions are still robust datasets that can be used in future experiments, mainly with the focus of deep learning.

6.2 Results obtained on novel MERGE datasets

In order to better understand the feature sets considered, a nomenclature is proposed to each of the feature sets. Table 6.4 has an overview of all the proposed feature sets and their names.

Feature Set Name	Feature set description	Feature amount
MERGE_Panda	Features originally extracted in (Panda, 2019). It consists of a set of features extracted on the whole signal, as well as additional features extracted from the vocal stem extracted by Spleeter.	2701
MERGE_Percussion	Novel features extracted from solely the percussion track separated by Demucs, as described in 5.2.2.	57
MERGE_MIDI	Novel features extracted from solely the MIDI files outputted by MT3, as described in 5.2.1.	712
MERGE_MIDI_Percussion	Combination of MERGE_Percussion and MERGE_MIDI.	769
MERGE_Panda_MIDI_Percussion	Combination of MERGE_Panda and MERGE_MIDI_Percussion.	3470
MERGE_Panda_DMCS_Vocals	Similar to MERGE_Panda, but now instead of having the features extracted using the vocal stem separated by Spleeter, Demucs is used for the vocal stem separation.	2701
MERGE_Panda_DMCS_Drums	MERGE_Panda combined with the features extracted from the percussion track separated by Demucs.	3244
MERGE_Panda_DMCS_NoDrums	Similar to MERGE_Panda_DMCS_Drums but using the non drums stem of the track separated by Demucs instead of the drum stem.	3244
MERGE_All	Combination of MERGE_MIDI_Percussion with the extra features from MERGE_Panda_DMCS_Drums and MERGE_Panda_DMCS_NoDrums. Aimed to test all the new features.	4571

Table 6.4: Proposed feature set names for the proposed novel features.

6.2.1 Novel features only

The first step taken was to go through the same process of feature extraction, parameter optimization as previously, but now with only the newly extracted features. Likewise, this process was repeated for only the percussion features, the MIDI features, and for the combination of both in order to measure the impact of each set of features individually.

Features extracted from the percussion audio signal only

The first test done was extracting solely the features based on the percussion audio signal (this means that there are MT3 features related to percussion that are not present here) and see what results could be achieved. Table 6.5 shows the results obtained for all the datasets tested. The feature set used was MERGE_Percussion.

Dataset Name	Num. of Features	F1-Score
MERGE_Audio_Complete	50	56.6 ± 2.4
MERGE_Audio_Balanced	40	55.8 ± 2.6
MERGE_Bimodal_Complete	40	58.7 ± 3.3
MERGE_Bimodal_Balanced	50	56.6 ± 3.6

Table 6.5: Results obtained using the MERGE_Percussion feature set.

As Table 6.6 shows, there is also a lot of confusion between the 3rd and 4th quadrants. However, there is also confusion between Q1 and Q2, with having just these percussion features not being enough to distinguish well between the two quadrants.

	Predicted			
	Q1	Q2	Q3	Q4
True	53.69 ± 6.54	24.12 ± 5.37	9.66 ± 4.15	12.53 ± 4.37
	17.07 ± 4.69	78.61 ± 5.23	2.15 ± 1.73	2.17 ± 1.63
	11.84 ± 4.94	4.70 ± 2.97	46.02 ± 6.32	37.44 ± 7.04
	15.04 ± 5.15	3.22 ± 2.32	31.19 ± 7.02	50.55 ± 6.52

Table 6.6: Confusion matrix for MERGE_Bimodal_Complete with 40 features from the MERGE_Percussion feature set.

Dataset Name	F1 Q1	F1 Q2	F1 Q3	F1 Q4
MERGE_Audio_Complete	54.2 ± 4.1	74.4 ± 3.3	44.2 ± 4.8	52.8 ± 3.3
MERGE_Audio_Balanced	54.4 ± 4.1	72.9 ± 3.7	49.2 ± 4.4	47.0 ± 4.1
MERGE_Bimodal_Complete	53.2 ± 5.4	77.3 ± 3.6	48.1 ± 5.5	50.0 ± 5.4
MERGE_Bimodal_Balanced	55.5 ± 5.2	73.2 ± 5.0	49.8 ± 5.8	47.5 ± 5.6

Table 6.7: F1-score with standard deviation obtained for each quadrant for all the tested datasets using SVM with features from the MERGE_Percussion feature set.

In Table 6.7 we can see that the F1-score obtained for Q2 is good, but for the remaining quadrants is low when compared to the other set of features.

Features extracted from the MIDI file only

The second test was following the same principle as before but now only considering the features extracted from the MIDI files. Table 6.8 shows the results obtained from the novel features extracted from the MIDI files outputted by MT3. The feature set used was MERGE_MIDI.

Dataset Name	Num. of Features	F1-Score
MERGE_Audio_Complete	300	62.2 ± 2.3
MERGE_Audio_Balanced	200	62.0 ± 2.6
MERGE_Bimodal_Complete	200	64.5 ± 2.6
MERGE_Bimodal_Balanced	200	61.5 ± 3.4

Table 6.8: Results obtained using only the MERGE_MIDI feature set.

Compared to the results obtained using only the features extracted from the percussion track, we can see that there is less confusion between the first and second quadrant as shown in Table 6.9, indicating that the features from the MIDI are better at helping to distinguish between the two quadrants. Furthermore, the problem of the confusion between the third and fourth quadrant is also present.

However, the F1-score obtained for the second quadrant is already in line with the baseline approach, as can be seen in Table 6.10.

	Predicted			
True	Q1	Q2	Q3	Q4
	67.55 ± 5.03	15.96 ± 4.78	7.64 ± 3.35	8.85 ± 3.67
	10.82 ± 3.94	84.28 ± 4.52	3.27 ± 2.00	1.63 ± 1.54
	11.74 ± 4.33	3.96 ± 2.48	51.40 ± 6.60	32.90 ± 7.09
	12.55 ± 4.40	2.80 ± 2.10	33.40 ± 6.53	51.26 ± 6.47

Table 6.9: Confusion matrix for MERGE_Bimodal_Complete with the top 200 features from MERGE_MIDI feature set.

Dataset Name	F1 Q1	F1 Q2	F1 Q3	F1 Q4
MERGE_Audio_Complete	65.7 ± 4.0	80.1 ± 3.0	43.8 ± 4.3	56.9 ± 3.7
MERGE_Audio_Balanced	65.8 ± 4.1	78.7 ± 3.4	50.2 ± 4.2	53.1 ± 4.3
MERGE_Bimodal_Complete	65.9 ± 4.1	83.5 ± 3.6	51.8 ± 4.9	52.8 ± 5.2
MERGE_Bimodal_Balanced	65.9 ± 4.8	79.2 ± 4.0	53.8 ± 5.6	48.1 ± 5.5

Table 6.10: F1-score with standard deviation obtained for each quadrant for all the tested datasets using SVM with features from the MERGE_MIDI feature set.

All the new features combined

Finally, both sets of features were merged. In total, 769 features were extracted when combining both sets of features, 57 from the percussion track, and 512 from the MIDI file. The feature set used was MERGE_MIDI_Percussion.

Dataset Name	Num. of Features	F1-Score
MERGE_Audio_Complete	300	63.4 ± 2.4
MERGE_Audio_Balanced	300	63.5 ± 2.4
MERGE_Bimodal_Complete	200	64.2 ± 3.0
MERGE_Bimodal_Balanced	200	62.6 ± 3.0

Table 6.11: Results obtained using the MERGE_MIDI_Percussion feature set.

When joining the two datasets together, the results are very similar to what was previously obtained. As Table 6.11 shows, the results are either equal or only a bit above the results obtained using only the MIDI files. Table 6.12 shows the top 10 features for the MERGE_Bimodal_Complete dataset.

Rank	Feature Name
1	Drum Extent Percentage
2	Average Frame-Based Intensity
3	Average Amplitude
4	Standard Deviation of Amplitude
5	MIDI Drum Standard Deviation of Layers in a frame
6	Standard Deviation of Frame-Based Intensity
7	MIDI Drum Ratio of Transitions Musical Layers (0:1)
8	MIDI Drum Ratio of Transitions Musical Layers (1:0)
9	MIDI Melodic Ratio of Transitions Musical Layers (2:1)
10	Maximum Amplitude

Table 6.12: Ranked features and their names the MERGE_MIDI_Percussion feature set.

In the top 100 features, 22 of the features are from the percussion track, while the other 78 belong to the MIDI file. For the top 200, there are 166 features originating from the MIDI files and 34 from the percussion track. While this is the case, there are a few percussion features at the top of the ranking, such as the Drum Extent feature. Furthermore, 45 features related to the percussion track but originating from the MIDI files are also in the top 100, signifying their importance.

6.2.2 Combination of old features with the newly extracted features

Previously, a set of features using the audio track with all the stems was extracted, as well as a set of features using the vocal track extracted by Spleeter. The goal of this set of experiments is to measure the impact of adding the newly extracted features with the old ones, and seeing the classification results obtained. Table 6.13 shows the results obtained for the four tested datasets. The feature set used was MERGE_Panda_MIDI_Percussion.

When compared to the results obtained in Section 6.1.1 using only the baseline features, we see a improvement across the board, particularly in the bimodal

Dataset Name	Num. of Features	F1-Score
MERGE_Audio_Complete	700	72.7 ± 2.0
MERGE_Audio_Balanced	300	72.5 ± 2.4
MERGE_Bimodal_Complete	400	73.6 ± 2.9
MERGE_Bimodal_Balanced	500	72.3 ± 2.8

Table 6.13: Results obtained using the MERGE_Panda_MIDI_Percussion feature set.

complete dataset, going from 70.6% to 73.6% F1-Score. However, looking at the confusion matrix and the F1-scores obtained per quadrant, we can see that this rise was mostly due to better classification of the first and second quadrants, while the third and fourth quadrant remain with a much lower F1-Score when compared to the other two. Table 6.14 shows the confusion matrix obtained for MERGE_Bimodal_Complete.

	Predicted			
	Q1	**Q2**	**Q3**	**Q4**
True	78.49 ± 5.88	8.63 ± 3.90	4.29 ± 2.93	8.59 ± 3.56
	6.99 ± 2.68	91.32 ± 2.87	1.38 ± 1.30	0.31 ± 0.61
	7.84 ± 3.60	2.48 ± 2.21	59.86 ± 7.71	29.82 ± 6.86
	10.64 ± 3.89	0.79 ± 1.19	28.55 ± 5.95	60.03 ± 5.67

Table 6.14: Confusion matrix for MERGE_Bimodal_Complete with the top 400 features from MERGE_Panda_MIDI_Percussion feature set.

Dataset Name	F1 Q1	F1 Q2	F1 Q3	F1 Q4
MERGE_Audio_Complete	75.9 ± 2.6	87.6 ± 2.5	60.1 ± 4.4	65.9 ± 3.1
MERGE_Audio_Balanced	75.7 ± 3.5	87.4 ± 2.6	63.3 ± 4.1	63.8 ± 4.3
MERGE_Bimodal_Complete	76.4 ± 4.4	91.1 ± 2.7	61.1 ± 5.9	60.7 ± 4.7
MERGE_Bimodal_Balanced	76.5 ± 4.1	89.4 ± 2.9	63.9 ± 4.6	60.3 ± 5.2

Table 6.15: F1-score with standard deviation obtained for each quadrant for all the tested datasets using SVM with features from the MERGE_Panda_MIDI_Percussion feature set.

As Table 6.15 shows, there is an improvement across the board on the F1-scores obtained, even surpassing 91% for the Q2, thus proving that new proposed features regarding percussion and song texture are important for MER problems.

Table 6.16 showcases the statistical significance test results when compared to the baseline approach (MERGE_Panda), showcasing that the new features have a statistical significance.

Finally, looking at the feature ranking for MERGE_Bimodal_Complete as this was the one that obtained better results, we see that in the top 200, 32 are features originating from the MT3 MIDI file, and 7 are from the percussion track. However, 5 of these 7 are highly ranked, being located in positions 6, 9, 10, 20 and 31.

Audio_Complete	Audio_Balanced	Bimodal_Complete	Bimodal_Balanced
Significant (2.3756×10^{-8})	Significant (3.1317×10^{-6})	Significant (2.0067×10^{-11})	Significant (1.7170×10^{-4})

Table 6.16: Statistical Significance test against the MERGE_Panda feature set on the new MERGE datasets using the MERGE_Panda_MIDI_Percussion feature set.

6.2.3 Replacing Spleeter features with Demucs

As previously mentioned, a set of features was previously extracted using the vocal stem extracted by Spleeter. As Demucs boasted better results in source separation, tests were conducted in order to measure the impact of using Demucs to separate the voice instead of Spleeter. The feature set used was MERGE_Panda_-DMCS_Vocals.

As such, instead of using the old features extracted with Spleeter, those features were replaced with the ones extracted from Demucs and the same process of feature ranking, and hyper-parameter optimization were done. Table 6.17 shows the results obtained.

Dataset Name	**Num. of Features**	**F1-Score**
MERGE_Audio_Complete	200	72.2 ± 2.4
MERGE_Audio_Balanced	200	71.4 ± 2.3
MERGE_Bimodal_Complete	200	72.6 ± 2.4
MERGE_Bimodal_Balanced	150	70.9 ± 2.9

Table 6.17: Results obtained using the MERGE_Panda_DMCS_Vocals feature set.

When compared to the other approaches, we do not see a huge increase in F1-score when replacing Demucs with Spleeter. But analyzing the feature ranking helps understand the results a lot better. In the baseline approach, the features extracted solely for the voice signal were found to not be very important. In the baseline features, for MERGE_Audio_Complete, there are 2 features related to the voice in the top 100, and 9 in the top 200, with this being similar for the other datasets.

Table 6.18 helps further prove this assumption, as two of the four datasets were found to not have significant statistical differences when compared to the MERGE-_Panda feature set.

Audio_Complete	Audio_Balanced	Bimodal_Complete	Bimodal_Balanced
Significant 0.0048	Not Significant	Significant 1.3577×10^{-6}	Not Significant

Table 6.18: Statistical Significance test against the MERGE_Panda feature set on the new MERGE datasets using the MERGE_Panda_DMCS_Vocals feature set.

When extracting this new set of features, in the top 100 features for MERGE_Audio-

_Complete, we have two from Demucs, and in the top 200 we have 13. Thus, effectively, the features being used for classification are in the grand majority just the baseline features, thus explaining having such similar results when compared to the baseline results.

6.2.4 Combination of old features with features extracted from vocals and percussion tracks

As aforementioned, the previous feature set contained features extracted only from the vocal track, and this same set of features was extracted but now using the percussion track. The goal of this experiment is to see if there is any advantage to using the percussion track for emotion classification. The feature set used was MERGE_Panda_DMCS_Drums. Table 6.21 shows the best results obtained.

Dataset Name	Num. of Features	F1-Score
MERGE_Audio_Complete	200	72.2 ± 2.2
MERGE_Audio_Balanced	200	71.0 ± 2.0
MERGE_Bimodal_Complete	200	74.1 ± 2.6
MERGE_Bimodal_Balanced	150	72.1 ± 2.8

Table 6.19: Results obtained using the top features using the MERGE_Panda_DMCS_Drums feature set.

The results are still an improvement over baseline, but they are not as great as other feature combinations. This can possibly be attributed to the fact that there are already features that represent the same types of songs that the features extracted from the percussion signal do, meaning, that the main problem is as previously mentioned, to distinguish between the third and fourth quadrant, which is a problem that also plagues this combination of features. Furthermore, there are only 13 features from the percussion signal in the top 200 features.

Table 6.20 showcases the statistical significance results when compared to the MERGE_Panda approach. Just like the MERGE_Panda_DMCS_Vocals approach, extracting the old features in the new percussion track did not lead to drastically best results when compared to the other approaches.

Audio_Complete	Audio_Balanced	Bimodal_Complete	Bimodal_Balanced
Significant 1.5260×10^{-4}	Not Significant	Significant 1.9825×10^{-15}	Significant 0.0015

Table 6.20: Statistical Significance test against the MERGE_Panda feature set on the new MERGE datasets using the MERGE_Panda_DMCS_Drums feature set.

6.2.5 Combination of old features with features extracted from vocals and melodic tracks

Following the same principle as Section 6.2.4, a test was also done using the separated track with no drums, in it, keeping only the melodic part. The idea of this experiment is to see if removing the drums aids in the emotion classification results. The feature set used was MERGE_Panda_DMCS_NoDrums. Table 6.21 shows the results obtained using this feature set.

Dataset Name	Num. of Features	F1-Score
MERGE_Audio_Complete	200	72.4 ± 2.2
MERGE_Audio_Balanced	150	72.4 ± 2.5
MERGE_Bimodal_Complete	200	74.1 ± 2.5
MERGE_Bimodal_Balanced	200	72.1 ± 3.0

Table 6.21: Results obtained using the top features using the MERGE_Panda-_DMCS_NoDrums feature set.

The results obtained do show an increase when compared to the baseline results, achieving the best F1-score obtained in this work at 74.1%.

This indicates that the isolated melodic part of the track can be helpful in MER. This is also helped by the fact that in the top 150 features, 22 are features extracted from the isolated melodic signal.

This assumption is further cemented by the statistical significance test results found in Table 6.22, where when compared to the baseline approach, the new features proved to have significant statistical significance.

Audio_Complete	Audio_Balanced	Bimodal_Complete	Bimodal_Balanced
Significant 2.6725×10^{-6}	Significant 1.4435×10^{-5}	Significant 2.3236×10^{-16}	Significant 0.0085

Table 6.22: Statistical Significance test against the MERGE_Panda feature set on the new MERGE datasets using the MERGE_Panda_DMCS_NoDrums feature set.

6.2.6 Combination of all the extracted features

Finally, an experiment was done were all the extracted features were combined. These features include the previously extracted features from the vocal track, percussion and non-melodic tracks. Finally, the newly extracted features from the percussion track and the features extracted from the MIDI tracks are also included. The combined feature set is called MERGE_All. Table 6.23 shows the result obtained for each of the datasets.

As can be seen, the overall results show an improvement when compared to the baseline approach across the board. Nonetheless, analysing the confusion matrix

Dataset Name	Num. of Features	F1-Score
MERGE_Audio_Complete	250	72.6 ± 2.4
MERGE_Audio_Balanced	250	72.8 ± 2.6
MERGE_Bimodal_Complete	250	74.1 ± 2.5
MERGE_Bimodal_Balanced	300	72.2 ± 3.0

Table 6.23: Results obtained using the top features using the MERGE_All feature set.

and the F1-score obtained per quadrant will allow us to further understand the results.

Looking particularly MERGE_Bimodal_Complete, a graph consisting of the number of features on the X axis and the overall F1-score in the Y axis was done, in order to understand how much the amount of features influenced the final result. As Figure 6.1 shows, the best result obtained was with 250 features, while containing to add features led to worse results, except in a few cases where the results rose a bit.

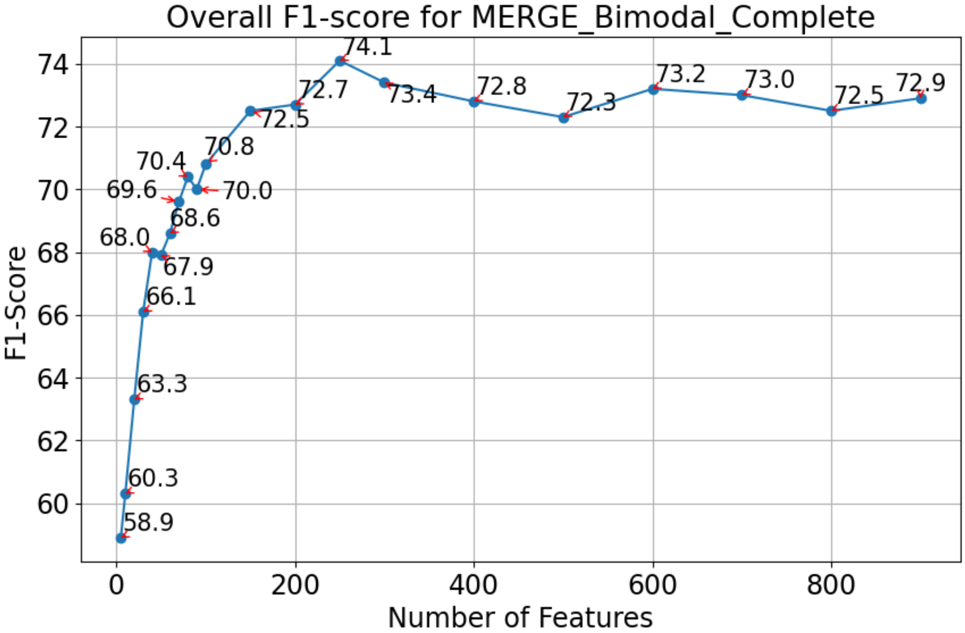


Figure 6.1: Graph showcasing the F1-score evolution based on the number of top features chosen for MERGE_Bimodal_Complete using the MERGE_All feature set.

In order to best understand the overall results, Table 6.24 and 6.25 show the confusion matrix for MERGE_Bimodal_Complete using the baseline features (MERGE_-Panda) and another confusion matrix using the MERGE_All feature set.

As we can see, across the board the values obtained increase when using the MERGE_All feature set when compared to the baseline results. The major prob-

	Predicted			
	Q1	Q2	Q3	Q4
True	76.32 ± 5.22	10.45 ± 4.36	4.73 ± 3.02	8.50 ± 3.43
	8.70 ± 3.36	89.51 ± 3.44	1.16 ± 1.23	0.64 ± 1.01
	7.86 ± 3.41	2.76 ± 2.22	56.78 ± 6.36	32.60 ± 5.85
	13.15 ± 4.71	0.68 ± 1.10	30.03 ± 6.48	56.15 ± 6.90

Table 6.24: Confusion matrix for MERGE_Bimodal_Complete using the MERGE_Panda feature set.

	Predicted			
	Q1	Q2	Q3	Q4
True	78.64 ± 5.92	8.31 ± 3.99	4.21 ± 2.31	8.84 ± 4.21
	7.04 ± 2.80	91.47 ± 3.24	1.16 ± 1.29	0.33 ± 0.61
	7.88 ± 3.65	3.32 ± 2.61	60.14 ± 6.14	28.66 ± 5.80
	11.43 ± 4.04	0.52 ± 0.94	27.86 ± 6.14	60.19 ± 7.07

Table 6.25: Confusion matrix for MERGE_Bimodal_Complete with 300 features from MERGE_All feature set.

lem still continues to be the confusion between Q3 and Q4, but the new features helped lessen that confusion. The second major contributors of bad results is the confusion between Q1 and Q4, and finally Q1 and Q2. These both got improved with the new features, with the main improvement coming in the confusion between Q1 and Q2.

It is also important to take a look at the F1-score per quadrant obtained, shown on Table 6.26.

Dataset Name	F1 Q1	F1 Q2	F1 Q3	F1 Q4
MERGE_Audio_Complete	74.5 ± 3.0	87.2 ± 2.6	61.4 ± 4.1	66.3 ± 3.4
MERGE_Audio_Balanced	76.1 ± 3.1	87.1 ± 2.8	64.1 ± 4.7	64.0 ± 4.2
MERGE_Bimodal_Complete	76.2 ± 4.1	91.1 ± 2.6	61.7 ± 5.1	61.0 ± 5.0
MERGE_Bimodal_Balanced	75.9 ± 4.4	88.8 ± 2.9	64.5 ± 5.3	59.9 ± 5.0

Table 6.26: F1-score with standard deviation obtained for each quadrant for all the tested datasets using SVM with features from the MERGE_All feature set.

Overall, using this new set of features shows improvement for almost all the quadrants when compared to the baseline, as shown in Table 6.27. This table shows the difference in F1-score between the MERGE_All feature sets and the baseline results, for each quadrant for each of the four datasets. The F1-scores that increased are colored in green, while the ones that decreased are colored in red.

We can see that there is an increase across the board in all the quadrants, further signifying that the newly proposed features are helpful for MER studies. The quadrant with the biggest increases are the third and the fourth, as these were the quadrants that previously had the most confusion. The second quadrant already had very high results (between 87 and 89% F1-score) but now has risen even

Dataset Name	Q1	Q2	Q3	Q4
MERGE_Audio_Complete	1.3	0.6	2.6	2.4
MERGE_Audio_Balanced	1.8	0.7	1.9	3.1
MERGE_Bimodal_Complete	2.8	1.7	3.3	4
MERGE_Bimodal_Balanced	1.5	0.7	1.4	1.3

Table 6.27: Table with the difference in F1-Scores on MERGE_Bimodal_Complete using the MERGE_Panda and MERGE_All feature sets.

more, albeit slightly than the other quadrants.

Audio_Complete	Audio_Balanced	Bimodal_Complete	Bimodal_Balanced
Significant 6.8288×10^{-7}	Significant 1.9628×10^{-7}	Significant 4.2534×10^{-12}	Significant 0.0026

Table 6.28: Statistical Significance test against the MERGE_Panda feature set on the new MERGE datasets using the MERGE_All feature set.

Furthermore, analyzing the results of the statistical significance tests will also allow us to understand better if the new features are indeed helpful for MER. Table 6.28 showcases the results obtained for the four datasets. With these results, we can conclude that the new features are indeed helpful to aid in solving the problem of MER.

Nonetheless, it is also important to analyze the features present in the top 200 features for MERGE_Bimodal_Complete in order to understand if the proposed features are useful for MER. In Table 6.29, for each of the new feature set, the number of features present in the top 200 features for MERGE_Bimodal_Complete is shown.

Feature origin	Num. of Features
MERGE_MIDI	21
MERGE_Percussion	7
MERGE_Panda_DMCS_Drums (Only new)	14
MERGE_Panda_DMCS_NoDrums (Only new)	20

Table 6.29: Number of features from each of the corresponding newly proposed feature sets in the top 200 of MERGE_Bimodal_Complete using the MERGE_All feature set.

Table 6.30 and Table 6.31 showcase in the detail the novel features present in the top 200. As the tables show, drum related features proved to be important in MER. Furthermore, the features extracted from the MIDI files, particularly instrument presence and layer information, also helped increase the results. Finally, information extracted from the melodic and percussion tracks extracted by Demucs also proved important. Thus, in all the proposed feature sets, there were features that contributed to improving the overall results.

Overall, 62 of the top 200 features are novel features, which is a good representation and further indicates the usefulness of the features in MER problems.

In Table 6.30 and Table 6.31, features extracted from the percussion track are colored in light blue, features related to Demucs drum information are colored in light pink, features originated from the MIDI file are colored in green, and finally features originating from the melodic track separated by Demucs are colored in orange.

H

Rank	Feature Name
5	DEMUCS NODRUMS TEXTURE Musical Layers Mean
6	Drum Extent Percentage
7	DEMUCS NODRUMS TEXTURE ML3 Thicker Texture Percentage
10	Average Frame Based Intensity
12	Average Amplitude
13	DEMUCS DRUMS DYNAMICS Notes Intensity Mean
14	DEMUCS NODRUMS TEXTURE Musical Layers Std
20	DEMUCS DRUMS EXPRESSIVE TECHNIQUES Tremolo Salience Mean
23	Std Amplitude
25	DEMUCS DRUMS INTENSITY CONTOURS Intensity Contour Polynomial of degree 1 coefficient 0
28	Std Frame Based Intensity
32	DEMUCS DRUMS DYNAMICS Notes Intensity Std
35	DEMUCS DRUMS EXPRESSIVE TECHNIQUES Tremolo Salience Max
36	MIDI MELODIC RATIO OF TRANSITIONS MUSICAL LAYERS THREE OR MORE TO TWO
40	DEMUCS DRUMS DYNAMICS Notes Intensity Max
41	DEMUCS DRUMS INTENSITY CONTOURS Intensity Contour to Polynomial degree 7 RMSE
58	Max Amplitude
61	MIDI MELODIC RATIO OF TRANSITIONS MUSICAL LAYERS TWO TO THREE OR MORE
63	DEMUCS DRUMS EXPRESSIVE TECHNIQUES Tremolo Extent Weighted Mean
71	MIDI MELODIC RATIO OF TRANSITIONS MUSICAL LAYERS TWO TO ONE
73	Max Frame Based Intensity
81	DEMUCS DRUMS EXPRESSIVE TECHNIQUES Tremolo Salience Std
82	DEMUCS NODRUMS TEXTURE State Transitions ML2 ML3 Per Sec
93	DEMUCS NODRUMS TEXTURE State Transitions ML3 ML2 Per Sec

Table 6.30: Novel features in the top 100 of the MERGE Bimodal Complete dataset using the MERGE All feature set. Features extracted from the percussion track are colored in light blue, features related to Demucs drum information are colored in light pink, features originated from the MIDI file are colored in green, and features originating from the melodic track separated by Demucs are colored in orange.

Rank	Feature Name
101	MIDI BASS DRUM 1 PERCUSSION INSTRUMENT PERCENTAGE NOTES
103	DEMUCS NODRUMS TEXTURE State Transitions ML1 ML2 Per Sec
106	MIDI BASS DRUM 1 PERCUSSION INSTRUMENT TOTAL NOTES
107	MIDI CLOSED HI HAT PERCUSSION INSTRUMENT TOTAL NOTES
108	DEMUCS DRUMS EXPRESSIVE TECHNIQUES Tremolo Salience Min
111	DEMUCS NODRUMS TEXTURE State Transitions ML2 ML1 Per Sec
113	MIDI MELODIC RATIO OF TRANSITIONS MUSICAL LAYERS ONE TO TWO
115	MIDI CLOSED HI HAT PERCUSSION INSTRUMENT PERCENTAGE NOTES
121	DEMUCS NODRUMS TEXTURE Musical Layers Max
123	MIDI CHINESE CYMBAL PERCUSSION INSTRUMENT PRESENT
124	DEMUCS NODRUMS EXPRESSIVE TECHNIQUES Vibrato Base Freq Std
128	DEMUCS NODRUMS TEXTURE ML1 Monophonic Texture Percentage
130	DEMUCS NODRUMS FRACTAL Fractal Dimension Global
136	DEMUCS NODRUMS EXPRESSIVE TECHNIQUES Vibrato Rate Std
142	DEMUCS DRUMS FRACTAL Fractal Dimension Max
143	DEMUCS NODRUMS VAT Number of Silence Sections
144	DEMUCS NODRUMS VAT Number of Voice Sections
147	DEMUCS NODRUMS VAT Voice Segments Per Interval Mean
149	DEMUCS NODRUMS VAT Voice Segments Per Second
150	MIDI CRASH CYMBAL 2 PERCUSSION INSTRUMENT PRESENT
155	DEMUCS DRUMS FRACTAL Fractal Dimension Std
156	MIDI PERCUSSION INSTRUMENT PERCENTAGE NOTES
160	PERCUSSION TOTAL NOTES
166	DEMUCS NODRUMS EXPRESSIVE TECHNIQUES Vibrato Extent Std
167	DEMUCS DRUMS EXPRESSIVE TECHNIQUES Tremolo Extent Max
168	MIDI CHORDOPHONE RATIO OF TRANSITIONS MUSICAL LAYERS THREE OR MORE TO TWO
170	MIDI STRING ENSEMBLE 2 INSTRUMENT PRESENT
173	DEMUCS NODRUMS EXPRESSIVE TECHNIQUES Vibrato Rate Kurtosis
181	MIDI DRUM PERCENTAGE FRAMES WITH NO LAYER
182	MIDI DRUM AVERAGE LAYERS IN FRAME
183	MIDI DRUM PERCENTAGE FRAMES WITH ONE LAYER
186	MIDI SPLASH CYMBAL PERCUSSION INSTRUMENT PRESENT
187	DEMUCS NODRUMS EXPRESSIVE TECHNIQUES Vibrato Base Freq Kurtosis
191	MIDI DRUM RATIO OF TRANSITIONS MUSICAL LAYERS ZERO TO ONE
193	MIDI DRUM RATIO OF TRANSITIONS MUSICAL LAYERS ONE TO ZERO
195	DEMUCS DRUMS FRACTAL Fractal Dimension Skewness
197	MIDI ACOUSTIC BASS INSTRUMENT TOTAL NOTES
198	DEMUCS NODRUMS EXPRESSIVE TECHNIQUES Tremolo Notes in Cents Max
199	MIDI MEMBRANOPHONE RATIO OF TRANSITIONS MUSICAL LAYERS ONE TO THREE OR MORE

Table 6.31: Novel features in the top 100 to 200 of the MERGE Bimodal Complete dataset using the MERGE All feature set. Features extracted from the percussion track are colored in light blue, features related to Demucs drum information are colored in light pink, features originated from the MIDI file are colored in green, and features originating from the melodic track separated by Demucs are colored in orange.

6.2.7 Results summary

Table 6.34 presents a summary of all the best results obtained. It contains for each combination the dataset used, the feature set used, the number of top features used (e.g. top 200) and the hyper parameters used to achieve that result. Furthermore, it also contains the time in minutes that it took to train those hyperparameters, and finally, it also contains the overall F1-score obtained.

Feature set origin	Dataset	Num. of Features	Cost	Kernel	Gamma	Hyper parameter Optimization Time (Min)	Best Overall F1-Score
MERGE Panda	Merge Audio Complete	100	2.299	rbf	0.01	47.82	71
MERGE Panda	Merge Audio Balanced	100	2.46	rbf	0.006	41.72	70.8
MERGE Panda	Merge Bimodal Complete	100	2.798	rbf	0.008	44.50	71.2
MERGE Panda	Merge Bimodal Balanced	100	2.574	rbf	0.009	33.08	71.1
MERGE MIDI	Merge Audio Complete	150	9.899	rbf	0.0004	171.90	62.2
MERGE MIDI	Merge Audio Balanced	150	1.026	rbf	0.006	170.98	62
MERGE MIDI	Merge Bimodal Complete	150	2.86	rbf	0.005	178.85	64.5
MERGE MIDI	Merge Bimodal Balanced	150	1.207	rbf	0.004	90.07	61.5
MERGE Percussion	Merge Audio Complete	50	33.286	rbf	0.003	154.50	56.6
MERGE Percussion	Merge Audio Balanced	50	100	rbf	0.002	160.58	55.8
MERGE Percussion	Merge Bimodal Complete	40	100	rbf	0.002	41.88	58.7
MERGE Percussion	Merge Bimodal Balanced	50	81.502	rbf	0.001	71.77	56.6

Table 6.32: Summary of the best results obtained.

Feature set origin	Dataset	Num. of Features	Cost	Kernel	Gamma	Hyper parameter Optimization Time (Min)	Best Overall F1-Score
MERGE MIDI Percussion	Merge Audio Complete	150	9.129	rbf	0.0002	177.12	63.4
MERGE MIDI Percussion	Merge Audio Balanced	150	9.625	rbf	0.0004	156.15	63.5
MERGE MIDI Percussion	Merge Bimodal Complete	200	8.664	rbf	0.001	53.55	64.2
MERGE MIDI Percussion	Merge Bimodal Balanced	200	1.648	rbf	0.004	74.65	62.6
MERGE Panda MIDI Percussion	Merge Audio Complete	700	10.936	rbf	0.0002	183.23	72.7
MERGE Panda MIDI Percussion	Merge Audio Balanced	300	1.481	rbf	0.003	141.03	72.5
MERGE Panda MIDI Percussion	Merge Bimodal Complete	400	16.805	rbf	0.005	118.45	73.6
MERGE Panda MIDI Percussion	Merge Bimodal Balanced	500	2.444	rbf	0.002	66.75	72.3
MERGE Panda DMCS Vocals	Merge Audio Complete	200	1.916	rbf	0.003	112.12	72.2
MERGE Panda DMCS Vocals	Merge Audio Balanced	200	9.565	rbf	0.001	179.83	71.4
MERGE Panda DMCS Vocals	Merge Bimodal Complete	200	8.321	rbf	0.0001	192.38	72.6
MERGE Panda DMCS Vocals	Merge Bimodal Balanced	150	1.17	rbf	0.007	62.28	70.9

Table 6.33: Continuation of Summary of the best results obtained.

Feature set origin	Dataset	Num. of Features	Cost	Kernel	Gamma	Hyper parameter Optimization Time (Min)	Best Overall F1-Score
MERGE Panda DMCS NoDrums	Merge Audio Complete	200	2.3	rbf	0.003	187.75	72.3
MERGE Panda DMCS NoDrums	Merge Audio Balanced	150	2.393	rbf	0.006	156.40	72.5
MERGE Panda DMCS NoDrums	Merge Bimodal Complete	200	2.157	rbf	0.008	97.80	74.1
MERGE Panda DMCS NoDrums	Merge Bimodal Balanced	200	1.547	rbf	0.006	80.33	71.9
MERGE Panda DMCS Drums	Merge Audio Complete	200	2.134	rbf	0.004	202.92	69.6
MERGE Panda DMCS Drums	Merge Audio Balanced	200	71.093	rbf	0.00008	177.30	71
MERGE Panda DMCS Drums	Merge Bimodal Complete	200	1.786	rbf	0.01	128.92	73.9
MERGE Panda DMCS Drums	Merge Bimodal Balanced	150	2.719	rbf	0.007	76.63	72.1
MERGE All	Merge Audio Complete	200	2.548	rbf	0.005	83.33	72.6
MERGE All	Merge Audio Balanced	200	1.803	rbf	0.01	79.30	72.7
MERGE All	Merge Bimodal Complete	250	1.677	rbf	0.006	31.31	74.1
MERGE All	Merge Bimodal Balanced	300	1.871	rbf	0.003	42.87	72.4

Table 6.34: Continuation of Summary of the best results obtained.

6.2.8 Computational cost

In this subsection, an overview of time it takes to complete all the tasks that are required to complete the emotion classification is given. It is noteworthy that processes such as file moving amongst similar processes is not counted, only the specific tasks are counted.

Feature Extraction

There are several feature extraction processes (vocal track only, drums track, no-drums track), so an overview of the average for all three is presented. Thus, in order to obtain the total time for feature extraction, the final time value should be multiplied by three. There are also several types of features that need to be extracted. A full breakdown is presented in Table 6.36.

Note: Dressler (Dressler, 2016) and Melodia (Salamon and Gómez, 2012) are two frameworks used in (Panda, 2019) to estimate predominant fundamental frequencies (F0) and saliences as explained in 3.2.2.

Name	Time Per Song (seconds)
Standardization	0.491
Melodia Melodic lines	4.619
Dressler Melodic lines	6.549
Melodia Features	2.582
Dressler Features	0.494
Fractal Dimension	1.999
Voice analysis Toolkit	6.580
New Percussion Features based on audio	2.154
New Features based on MT3	1.072
Total	26.049

Table 6.35: Computational cost for feature extraction steps.

As Table 6.36 shows, it takes around 26 seconds to extract all the features for a single 30-second audio track, multiplied by three for all the three types of tracks, making it take around 75 seconds to extract all the 3 sets of features.

Source Separation

Table 6.36 provides an overview of the computational cost of separating the source tracks.

All of this combined makes it so that it takes nearly two minutes to extract the features for one 30-second song, which one of the main problems of classical machine learning approaches. In Deep Learning approaches, the training time for the neural networks might be greater for the same amount of songs, but processing new songs is almost instant, while in classical approaches all the features need to be extracted, leading to major time losses.

Name	Time Per Song (seconds)
Separate tracks using Spleeter	13.315
Separate tracks using Demucs (Vocals)	23.733
Separate tracks using Demucs (Drums)	22.569
Total	59.617

Table 6.36: Computational cost for source separation steps.

Although this can be somewhat combated by using faster languages (e.g. C) instead of slower languages, like MATLAB which is the language where majority of the features are extracted, it will still take longer when compared to DL approaches. Having this huge processing time also makes it unfeasible to use classical machine learning approaches in commercial products to solve emotional classification problems.

This page is intentionally left blank.

Chapter 7

Conclusions and future work

This chapter summarizes the main contributions of this work, and proposes possible directions that might be promising for future work.

7.1 Conclusion

As mentioned in this work, the biggest bottleneck of current Deep Learning approaches is the lack of data, and that is something that also pertains to the MER field. As (Louro, 2022) concludes in his thesis, “Beginning with the most pressing matter that severely limits the performance of the presented methodologies, is the lack of data in reasonable amounts to truly take advantage of DL.”.

In this work, a database expansion was proposed that takes the current database and increases the size four-fold, as well as increasing by nearly ten-fold the size of the bi-modal database. These databases will not only allow for the creation of DL approaches for one specific type (e.g. audio) but also allow for the development of networks that take both the audio and lyrics as input, as this is something that can help solve the confusion between Q3 and Q4 using lyrics information (Malheiro, 2017).

Furthermore, in this work, several novel features were proposed and frameworks were explored. Overall, the newly proposed features helped increase the F1-Score obtained, achieving the best result of F1-Score of 74.1% with 250 features on MERGE_Bimodal_Complete with the MERGE_All feature set. The newly proposed features helped mostly to decrease the confusion between the first and fourth quadrants, while also helping to decrease, albeit less, the main problem of current approaches, the confusion between the third and fourth quadrants.

Furthermore, with these newly built expanded datasets, Deep Learning approaches should be attempted to improve the ceiling in MER, as these approaches are the one that have shown the most promise. However, the frameworks used, mainly MT3, proved useful for the MER problem, and are certainly worthy of being explored for other approaches in the area.

As such, in this work the viability of the dataset expansions done by our team was proved, with achieving reasonable F1-Scores when compared to 4QAED, while having a much larger size. Finally, the viability of the proposed features for MER problems was also demonstrated.

7.2 Future Work

In terms of future work, there a few directions that can still be explored. A summary of these options is presented:

- Use the previously explored Deep Learning approaches in the much larger MERGE datasets, as the lack of data was one of the biggest drawback of these approaches before;
- Test new Deep Learning approaches that were not possible to use before due to the smaller dataset sizes;
- Explore bi-modal approaches, combining audio and lyrics. This is a promising approach to reduce the confusion between Q3 and Q4, since valence information is mostly captured by the lyrics (Malheiro, 2017);
- Explore new features that were not able to be explored in the scope of this work, for example pertaining to automatic time signature detection and features related to expressivity.

This page is intentionally left blank.

References

(2023). Music source separation benchmarks. `https://paperswithcode.com/task/music-source-separation#benchmarks`. Accessed on: September 3, 2023.

Aljanaki, A., Yang, Y.-H., and Soleymani, M. (2017). Developing a benchmark for emotional analysis of music. *PLoS ONE*, 12.

Allamanche, E. (2001). Content-based identification of audio material using mpeg-7 low level description.

Benward, B. and Saker, M. N. (2008). *Music: In Theory and Practice, Vol. I*. McGraw-Hill, 8 edition.

Bertin-Mahieux, T., Ellis, D., Whitman, B., and Lamere, P. (2011). The million song dataset. pages 591–596.

Bhagwat, P., Shelke, V., Murugkar, A., Dakwala, K., and Dharmadhikari, D. S. C. (2023). A survey on automatic music transcription. *IRE Journals*, 6:268. IRE 1704431.

Bittner, R., Salamon, J., Tierney, M., Mauch, M., Cannam, C., and Bello, J. (2014). Medleydb: A multitrack dataset for annotation-intensive mir research.

Bogert, B. P. (1963). The quefrency analysis of time series for echoes : cepstrum, pseudo-autocovariance, cross-cepstrum and saphe cracking.

Brandt, A., Nelson, B., and McClure, R. (2011). Musical form. In Sound Reasoning (21.2).

Brownlee, J. (2019). *Probability for Machine Learning: Discover How To Harness Uncertainty With Python*. Machine Learning Mastery.

Cabrera, D. (1999). The size of sound: Auditory volume reassessed. *Mikropolyphonie*, 5.

Cabrera, D., Ferguson, S., Rizwi, F., and Schubert, E. (2008). Psysound3: a program for the analysis of sound recordings. *The Journal of the Acoustical Society of America*, 123:3247.

Camacho, A. and Harris, J. (2008). A sawtooth waveform inspired pitch estimator for speech and music. *The Journal of the Acoustical Society of America*, 124:1638–52.

Cañón, G., Sebastián, J., Cano, E., Herrera, P., and Gómez, E. (2021). Transfer learning from speech to music: towards language-sensitive emotion recognition models. In *2020 28th European Signal Processing Conference (EUSIPCO)*, pages 136–140.

Cheuk, K. W., Herremans, D., and Su, L. (2021). Reconvat: A semi-supervised automatic music transcription framework for low-resource real-world data.

Cheveigné, A. and Kawahara, H. (2002). Yin, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111:1917–30.

Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366.

Dixon, T. (2012). "emotion": The history of a keyword in crisis. *Emotion Review*, 4(4):338–344.

Dong, Y., Yang, X., Zhao, X., and Li, J. (2019). Bidirectional convolutional recurrent sparse network (bcrsn): An efficient model for music emotion recognition. *IEEE Transactions on Multimedia*, PP:1–1.

Dressler, K. (2016). *Automatic Transcription of the Melody from Polyphonic Music*. PhD thesis.

Dubnov, S. (2004). Generalization of spectral flatness measure for non-gaussian linear processes. *IEEE Signal Processing Letters*, 11(8):698–701.

Duda, R., Hart, P., and G.Stork, D. (2001). *Pattern Classification*, volume xx.

Défossez, A. (2022). Hybrid spectrogram and waveform source separation.

Défossez, A., Usunier, N., Bottou, L., and Bach, F. (2021). Music source separation in the waveform domain.

Ekman, P. (1992). An argument for basic emotions. *Cognition and Emotion*, 6(3-4):169–200.

El Ayadi, M., Kamel, M. S., and Karray, F. (2011). Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, 44(3):572–587.

Farnsworth, P. R. (1954). A study of the hevner adjective list. *The Journal of Aesthetics and Art Criticism*, 13(1):97–103.

Feng, Y., Zhuang, Y., and Pan, Y. (2003). Popular music retrieval by detecting mood. pages 375–376.

Foote, J., Cooper, M., and Nam, U. (2002). Audio retrieval by rhythmic similarity.

Friberg, A., Schoonderwaldt, E., Hedblad, A., Fabiani, M., and Elowsson, A. (2014). Using listener-based perceptual features as intermediate representations in music information retrieval. *The Journal of the Acoustical Society of America*, 60:1951–1963.

Gabrielsson, A. (2001). Emotion perceived and emotion felt: Same or different? *Musicae Scientiae*, 5(1_suppl):123–147.

Gardner, J., Simon, I., Manilow, E., Hawthorne, C., and Engel, J. (2021). Mt3: Multi-task multitrack music transcription.

Goto, M. (2006a). A.chorus section detection method for musical audio signals and its application to a music listening station. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14:1783 – 1794.

Goto, M. (2006b). A chorus section detection method for musical audio signals and its application to a music listening station. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1783–1794.

Goto, M., Hashiguchi, H., Nishimura, T., and Oka, R. (2002). Rwc music database: Popular, classical, and jazz music databases.

Grekow, J. (2021). Music emotion recognition using recurrent neural networks and pretrained models. *Journal of Intelligent Information Systems*, 57:531 – 546.

Grey, J. M. (1975). An exploration of musical timbre. Master's thesis, Stanford University, Stanford, California.

Grosche, P. and Müller, M. (2009). A mid-level representation for capturing dominant tempo and pulse information in music recordings. *Proceedings of th 10th International Society for Music Information Retrieval Conference, 189-194 (2009)*.

Gómez, E. (2006). Tonal description of polyphonic audio for music content processing. *INFORMS Journal on Computing*, 18:294–304.

Harrington, J. and Cassidy, S. (1999). *Techniques in speech acoustics*. Text, Speech and Language Technology. Kluwer Academic.

Harte, C., Sandler, M., and Gasser, M. (2006). Detecting harmonic change in musical audio.

Hawthorne, C., Simon, I., Swavely, R., Manilow, E., and Engel, J. (2021). Sequence-to-sequence piano transcription with transformers.

He, H. and Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284.

He, Q., Sun, X., Yu, Y., and Li, W. (2022). Deepchorus: A hybrid model of multiscale convolution and self-attention for chorus detection.

Hennequin, R., Khlif, A., Voituret, F., and Moussallam, M. (2020). Spleeter: a fast and efficient music source separation tool with pre-trained models. *Journal of Open Source Software*, 5(50):2154. Deezer Research.

Hevner, K. (1936). Experimental studies of the elements of expression in music. *The American Journal of Psychology*, 48(2):246–268.

Hu, Y., Chen, Y., Yang, W., He, L., and Huang, H. (2022). Hierarchic temporal convolutional network with cross-domain encoder for music source separation. *IEEE Signal Processing Letters*, 29:1517–1521.

James, W. (1884). What is an emotion? *Mind*, 9(34):188–205.

Jiang, D.-N., Lu, L., Zhang, H.-J., Tao, J.-H., and Cai, L.-H. (2002a). Music type classification by spectral contrast feature. In *Proceedings. IEEE International Conference on Multimedia and Expo*, volume 1, pages 113–116 vol.1.

Jiang, D.-N., Lu, L., Zhang, H.-J., Tao, J.-H., and Cai, L.-H. (2002b). Music type classification by spectral contrast feature. In *Proceedings. IEEE International Conference on Multimedia and Expo*, volume 1, pages 113–116 vol.1.

Kim, M., Choi, W., Chung, J., Lee, D., and Jung, S. (2021). Kuielab-mdx-net: A two-stream neural network for music demixing.

Kim, M., Lee, J. H., and Jung, S. (2023). Sound demixing challenge 2023 music demixing track technical report: Tfc-tdf-unet v3.

Klapuri, A. and Davy, M. (2006). *Signal Processing Methods for Music Transcription*.

Lagrange, M., Martins, L., and Tzanetakis, G. (2008). A computationally efficient scheme for dominant harmonic source separation.

Laitz, S. G. (2007). *The Complete Musician*. Oxford University Press, USA, 2 edition.

Lartillot, O. (2018). Mirtoolbox 1.7.1 user's manual.

Laurier, C. (2011). Automatic classification of musical mood by content-based analysis.

Louro, P. (2022). Merge audio 2.0: Music emotion recognition next generation - audio classification with deep learning. Master's thesis, University of Coimbra.

Lu, L., Liu, D., and Zhang, H.-J. (2006). Automatic mood detection and tracking of music audio signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):5–18.

Luo, Y. and Yu, J. (2022). Music source separation with band-split rnn.

M, G. R., Rao, K. S., and Das, P. P. (2022). Melody extraction from polyphonic music by deep learning approaches: A review.

Malheiro, R. (2017). Emotion-based analysis and classification of music lyrics.

Malheiro, R., Panda, R., Gomes, P., and Paiva, R. P. (2016). Bi-modal music emotion recognition: Novel lyrical features and dataset.

Malheiro, R., Panda, R., Gomes, P., and Paiva, R. P. (2018). Emotionally-relevant features for classification and regression of music lyrics. *IEEE Transactions on Affective Computing*, 9:240–254.

Malloch, S. N. (1997). *Timbre and technology: an analytical partnership*. PhD thesis, University of Edinburgh.

Malík, M., Adavanne, S., Drossos, K., Virtanen, T., Ticha, D., and Jarina, R. (2017). Stacked convolutional and recurrent neural networks for music emotion recognition.

Manilow, E., Seetharaman, P., and Pardo, B. (2020). Simultaneous separation and transcription of mixtures with multiple polyphonic and percussive instruments.

Markov, K. and Matsui, T. (2015). Speech and music emotion recognition using gaussian processes.

Mauch, M., Cannam, C., Davies, M., Dixon, S., Harte, C., Kolozali, S., Tidhar, D., and Sandler, M. (2009). Omras2 metadata project. In *Proceedings of the 10th International Society for Music Information Retrieval Conference*. ISMIR.

Mauch, M. and Levy, M. (2011). Structural change on multiple time scales as a correlate of musical complexity. pages 489–494.

Meyers, O. (2007). A mood-based music classification and exploration system.

Nieto, O., McCallum, M., Davies, M., Robertson, A., Stark, A., and Egozy, E. (2019). The Harmonix Set: Beats, Downbeats, and Functional Segment Annotations of Western Popular Music. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, pages 565–572, Delft, The Netherlands. ISMIR.

Orjesek, R., Jarina, R., and Chmulik, M. (2022). End-to-end music emotion variation detection using iteratively reconstructed deep features. *Multimedia Tools Appl.*, 81(4):5017–5031.

Owen, H. (2000). *Music theory resource book*. Oxford University Press.

Paiva, R. P., Mendes, T., and Cardoso, A. (2006). Melody detection in polyphonic musical signals: Exploiting perceptual rules, note salience and melodic smoothness. *Computer Music Journal*, 30:80–98.

Panda, R. (2019). *Emotion-based Analysis and Classification of Audio Music*. PhD thesis.

Panda, R., Malheiro, R., and Paiva, R. P. (2018). Novel audio features for music emotion recognition. *IEEE Transactions on Affective Computing*, 11:614 – 626.

Panda, R., Malheiro, R., and Paiva, R. P. (2020a). Audio features for music emotion recognition: a survey. *IEEE Transactions on Affective Computing*, PP:1–1.

Panda, R., Malheiro, R., and Paiva, R. P. (2020b). Novel audio features for music emotion recognition. *IEEE Transactions on Affective Computing*, 11(4):614–626.

Panda, R. and Paiva, R. P. (2011). Using support vector machines for automatic mood tracking in audio music. volume 1.

Panda, R., Rocha, B., and Paiva, R. P. (2015). Music emotion recognition with standard and melodic audio features. *Applied Artificial Intelligence*, 29:313–334.

Pannese, A., Rappaz, M.-A., and Grandjean, D. (2016). Metaphor and music emotion: Ancient views and future directions. *Consciousness and Cognition*, 44:61–71.

Peeters, G., Giordano, B. L., Susini, P., Misdariis, N., and McAdams, S. (2011). The timbre toolbox: Extracting audio descriptors from musical signals. *The Journal of the Acoustical Society of America*, 130(5):2902–2916.

Plomp, R. and Levelt, W. J. M. (1965). Tonal consonance and critical bandwidth. *The Journal of the Acoustical Society of America*, 38(4):548–560.

Posner, J., Russell, J., and Peterson, B. (2005). The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development and psychopathology*, 17:715–34.

Rachman, F. H., Sarno, R., and Fatichah, C. (2020). Music emotion detection using weighted of audio and lyric features. In *2020 6th Information Technology International Seminar (ITIS)*, pages 229–233.

Rafii, Z., Liutkus, A., Stöter, F.-R., Mimilakis, S. I., and Bittner, R. (2017). The MUSDB18 corpus for music separation.

Refaeilzadeh, P., Tang, L., and Liu, H. (2009). *Cross-Validation*, volume 25, pages 532–538. Springer US, Boston, MA, 2nd edition.

Robnik-Sikonja, M. and Kononenko, I. (1997). An adaptation of relief for attribute estimation in regression. In *International Conference on Machine Learning*.

Rouard, S., Massa, F., and Défossez, A. (2023). Hybrid transformers for music source separation. In *ICASSP 23*.

Russell, J. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*, 39:1161–1178.

Sachs, C. (1914). Die hornbostel-sachs'sche klassifikation der musikinstrumente. *Naturwissenschaften*, 2:1056–1059.

Salamon, J. and Gómez, E. (2012). Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Transactions on Audio, Speech Language Processing - TASLP*, 20:1759–1770.

Schubert, E. (2003). Update of the hevner adjective checklist. *Perceptual and Motor Skills*, 96(3_suppl):1117–1122. PMID: 12929763.

Schubert, E. (2004). Modeling perceived emotion with continuous musical features. *Music Perception*, 21:561–585.

Sgouros, T., Bousis, A., and Mitianoudis, N. (2022). An efficient short-time discrete cosine transform and attentive multiresunet framework for music source separation. *IEEE Access*, 10:119448–119459.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423.

Smith, J., Burgoyne, J., Fujinaga, I., De Roure, D., and Downie, J. (2011). Design and creation of a large-scale database of structural annotations. pages 555–560.

Sá, P. (2021). *MERGE Audio: Music Emotion Recognition next Generation - Audio Classification with Deep Learning*. PhD thesis, University of Coimbra.

Tzanetakis, G. (2002). *Manipulation, Analysis and Retrieval Systems for Audio Signals*. PhD thesis, USA. AAI3041872.

Tzanetakis, G. and Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10:293 – 302.

Variety (2022). Spotify reaches 456 million total monthly users in q3, up 20% year over year and topping expectations. `https://variety.com/2022/digital/news/spotify-q3-results-456-million-total-users-1235414241`. [Online; accessed 05-January-2023].

Vincent, E., Gribonval, R., and Fevotte, C. (2006). Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1462–1469.

Vyas, G., Dutta, M. K., Atassi, H., and Burget, R. (2014). Detection of chorus from an audio clip using dynamic time warping algorithm. In *2014 Recent Advances in Engineering and Computational Sciences (RAECS)*, pages 1–6.

Wang, J.-C., Hung, Y.-N., and Smith, J. B. L. (2022). To catch a chorus, verse, intro, or anything else: Analyzing a song with structural functions.

Wang, J.-C., Smith, J. B. L., Chen, J., Song, X., and Wang, Y. (2021). Supervised chorus detection for popular music using convolutional neural network and multi-task learning.

Wang, S.-Y., Wang, J.-C., Yang, Y.-H., and Wang, H.-M. (2014). Towards time-varying music auto-tagging based on cal500 expansion. In *2014 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6.

Warriner, A., Kuperman, V., and Brysbaert, M. (2013). Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior research methods*, 45.

Wu, B., Zhong, E., Horner, A., and Yang, Q. (2014). Music emotion recognition by multi-label multi-layer multi-instance multi-view learning. *MM 2014 - Proceedings of the 2014 ACM Conference on Multimedia*, pages 117–126.

Yang, L., Rajab, K., and Chew, E. (2016). Ava: An interactive system for visual and quantitative analyses of vibrato and portamento performance styles.

Yang, y.-h. and Chen, H. (2012). Machine recognition of music emotion: A review. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3.

Yeh, C.-H., Lin, Y.-D., Lee, M.-S., and Tseng, W.-Y. (2010). Popular music analysis: Chorus and emotion detection. pages 907–910.

Zheng, F., Song, Z., Li, L., Yu, W., Zheng, F., and Wu, W. (2000). The distance measure for line spectrum pairs applied to speech recognition.

This page is intentionally left blank.

Appendices

Appendix A

MEVD, MEVD Dataset expansion and chorus detection approaches

One of the previous focuses of this work was in Music Emotion Variation Detection (MEVD). However, due to slow progress on the annotations for the MEVD songs, it was not possible to conclude this work in the scope of this work. However, the following section provides a bit of insight on the existing approaches for MEVD as well as approaches that might be useful for the creation of features for the scope of MEVD. Furthermore, information regarding the status of the expansion of our current MEVD dataset is provided.

A.1 MEVD

This section provides an overview of approaches to solve the problem of emotion classification with MEVD.

A.1.1 Classical approaches

As stated before, compared to Static MER, MEVD has seen a considerable less amount of research put into it, mainly due to the sheer difficulty in creating high quality datasets.

One approach was conducted by (Schubert, 2004), in which using linear regression models, predictions for arousal and valence values were made for a song using the Russell model. In total 5 features were used: melodic contour, tempo, loudness, spectral centroid and texture. The dataset consisted of romantic songs annotated by 67 annotators, for a total of four songs annotated every 1000 ms. The results obtained showed that loudness and tempo variations had correlations with changes in arousal, however none of the studied features showed any correlation with valence. However, the dataset used in this dataset is very small and all the songs are of the same genre (romantic).

Another approach was by (Panda and Paiva, 2011), in which supervised learning was used in an attempt to propose a solution for automatic mood tracking in audio. The approach also used the Russell model. The training dataset comprised of 189 songs, with 25 seconds in length, annotated with arousal and valence values by at least 10 volunteers. For testing, volunteers were asked to annotate changes between quadrants in 57 full songs, with every song being annotated by two volunteers. Only the songs that had 80% matching rate between both annotators were selected for testing, shortening the testing set from 57 to 29 songs. The results obtained hovered around the 53% to 55% in terms of accuracy, mostly due to the smaller size of the dataset, which is also a problem.

In (Markov and Matsui, 2015), an approach using Gaussian Process (GP) regression was proposed. The feature vectors were calculated in a window of 1000 ms, with a total of 45 vectors per music clip. The results obtained were a value of Kendal rank correlation coefficient of 0.51 for arousal and 0.32 for valence. This approach was done for the MediaEval 2013 benchmark evaluation campaign, and also used the Russell model of emotion.

A.1.2 Deep Learning approaches

One deep learning approach to tackle MEVD was proposed in (Malík et al., 2017) where both convolutional neural networks (CNN) and recurrent neural networks (RNN) were used for emotion recognition using the Russell model of emotion. The dataset used for training was a sub-set of the DEAM dataset, previously mentioned in Section 3.1. The best performing system on this dataset used 65 features, with these being the features fed to the DL model. Moreover, in an attempt to see if the model would be able to achieve good classification results using just raw data, another model was created but it was only trained using raw features.

A total of 431 audio samples with a duration of 45 seconds was used for training the model. However, the first 15 seconds of each sample were used for the annotators to get accustomed with the annotation process, so only the final 30 seconds were used for training. This process resulted in 60 annotations for each of the 30 second samples, annotated every 500 ms with arousal and valence values in the range of [-1,1]. The evaluation was conducted using 58 complete songs from the MedleyDB dataset (Bittner et al., 2014) and from the music website Jamendo [1]. The system with the baseline features ended up achieving the best results (having a Root Mean Squared Error (RMSE) of 0.202 for arousal and 0.268 for valence).

This approach outperformed the previously considered best systems, however it has the problem of using long segments, which means that there is the possibility of having more than one emotion present inside the same segment.

(Dong et al., 2019) proposed a new Bidirectional Convolutional Recurrent Sparse Network (BCRSN) for emotion recognition in music using Russell's model. This model adaptively learns the Sequential-information-included Affect-salient Features (SII-ASF) from the two-dimensional time-frequency representation (e.g. spectrogram) of music audio signals. This BCRSN combines feature extraction, affect-

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

salient feature selection and emotion prediction in order to achieve continuous emotion prediction on audio files. To train this model a part of the DEAM dataset was used, totalling 431 full songs. The evaluation set consists of 58 songs from the same database. To test the capacity of the model to generalize, 240 pop songs from the MTurk dataset [2] were also used for evaluation. These songs are annotated in 15-second segments by 7 to 23 annotators.

The results obtained were very good, having an average RMSE of 0.101 for arousal and 0.123 for valence in the DEAM dataset, and similar results in the MTurk dataset. However, this approach is a very complex one, and even with the implementation of new methods to reduce the training time of the model, it still takes a long time to train.

(Orjesek et al., 2022) proposed a solution based on deep neural networks that mines emotion-related features from the raw audio waveform. The datasets used for training and evaluation were the same used in (Dong et al., 2019), and the emotional taxonomy was also the Russell model of emotion. The results were compared against other models from the latest edition of MediaEval's "Emotion In Music" benchmark, as well as the BCRSN model (Dong et al., 2019) using both baseline features and spectogram as input possibilities. The model overall performed better compared to other approaches, achieving a Pearson Correlation Coefficient (PCC) of 0.66 for arousal and 0.637 for valence.

However, the RMSE scores are not always better when compared to other approaches, under-performing when compared to the BCRSN approach trained with spectogram as input in arousal and the LSTM-RNN architecture in valence.

These approaches further example the need of a bigger and higher quality dataset, with this objective being one of the main goals of this thesis.

[2] Dataset built using Amazon Mechanical Turk

Table A.1: Review of MEVD approaches

Paper	Approach	Emotion Taxonomy	Datasets	Features and Input	Models	Results	Notes/Observations
(Schubert, 2004)	Classical ML	Russell's A/V Model	4 romantic songs, annotated every 1000ms	5 features	Linear regression models	Detected changes in arousal but not in valence	The dataset is very small and all the songs are of the same genre.
(Panda and Paiva, 2011)	Classical ML	Russell's A/V Model	57 full songs annotated in 25 second intervals	Standard audio features	SVM	0.53 to 0.55 accuracy	The dataset used is very small.
(Markov and Matsui, 2015)	Classical ML	Russell's A/V Model	MediaEval 2014, annotated in 0.4 second intervals	Standard audio features	Gaussian Process regression	Kendal rank correlation coefficient of 0.51 for arousal and 0.32 for valence	
(Malík et al., 2017)	Deep Learning	Russell's A/V Model	DEAM 431 sample sub-set for training and 58 complete songs for evaluation	Standard features or spectrogram	CNN and RNN	RMSE of 0.202 for arousal and 0.268 for valence	The size of the samples can make it so that one sample has more than one dominant emotion.

Table A.2: Review of MEVD approaches

Paper	Approach	Emotion Taxonomy	Datasets	Features and Input	Models	Results	Notes/Observations
(Dong et al., 2019)	Deep Learning	Russell's A/V Model	DEAM 431 sample sub-set for training and 58 complete songs for evaluation	Spectogram	BCRSN	RMSE of 0.101 for arousal and 0.123 for valence	Model is very complex leading to a long time required for training.
(Orjesek et al., 2022)	Deep Learning	Russell's A/V Model	DEAM 431 sample sub-set for training and 58 complete songs for evaluation	Standard audio features and spectogram	Modified CNN and RNN	PCC of 0.66 for arousal and 0.637 for valence	The RMSE achieved underperforms when compared to a few other models.

A.2 DeepChorus and chorus detection approaches

Due to the main focus of this work being working with Static MER (which is made up of 30 second audio segments), using this tool did not prove useful in the creation of form features. However, it is important to consider this framework as a viable option for the creation of form features related to song structure for MEVD studies, where the full song is used.

DeepChorus (He et al., 2022), is a model created to identify the chorus segment in a given song. It is, at the time of writing, the state of the art model in chorus detection, surpassing the approach proposed in (Wang et al., 2021) by a few percentage points.

(Wang et al., 2021) was later applied in the whole context of song structure, instead of only identifying the chorus. In (Wang et al., 2022), a model was proposed to segment the most common parts of a song (e.g. verse, chorus, bridge, interlude) and achieved state of the art results in this specific task, although using a chorus detection model that had since been surpassed.

The DeepChorus model has two main structures: a multi-scale network that determines preliminary representation of the chorus segments, and a self attention convolution network that processes the features into probability curves that represents the presence of chorus. Finally, an adaptive threshold is applied in order to get a binary value out of the original curve (chorus or non-chorus). An illustration of the model can be seen in Figure A.1.

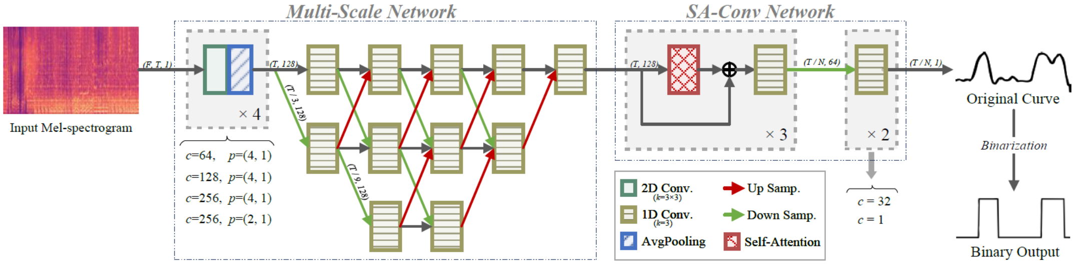


Figure A.1: Visualization of the DeepChorus Model (He et al., 2022).

This model was trained using 886 tracks from the HARMONIX dataset (Nieto et al., 2019) and from an extra 102 songs from The Beatles and Michael Jackson from the Isophonics dataset (Mauch et al., 2009). For testing, various datasets were used, in particular: 100 songs from the RWC dataset (Goto et al., 2002), 210 songs with the "Popular" tag from SALAMI (Smith et al., 2011) and 198 songs from the same dataset. There is also no overlap between both subsets of the SALAMI dataset.

This model achieved F1-scores of 0.675, 0.611 and 0.501, respectively, on the three aforementioned test datasets. These values are all higher than other approaches, with an increase of over 14% on the SALAMI "Popular" dataset and 12% on the SALAMI "Live" dataset over the at the time best approach.

Since chorus detection is not the main focus of the thesis, a thorough review of the state of the art was not conducted in this work, however throughout the years there have been many attempts at automating the task of detecting the chorus of a song, with various degrees of success. (Goto, 2006b) (Yeh et al., 2010) (Vyas et al., 2014) (Goto, 2006a) (Rachman et al., 2020)

The model has also been modified to produce images to better showcase the results of the model, as well as outputting the timestamps for the start and finish of each chorus section, instead of only reporting values such as F1-score and accuracy as is the case for the original model.

The image below shows an output for the song “Crown” by Seulgi, with the top graph corresponding to the ground truth chorus, illustrated by the green sections, and the below graph corresponding to the predicted chorus, illustrated by the red sections.

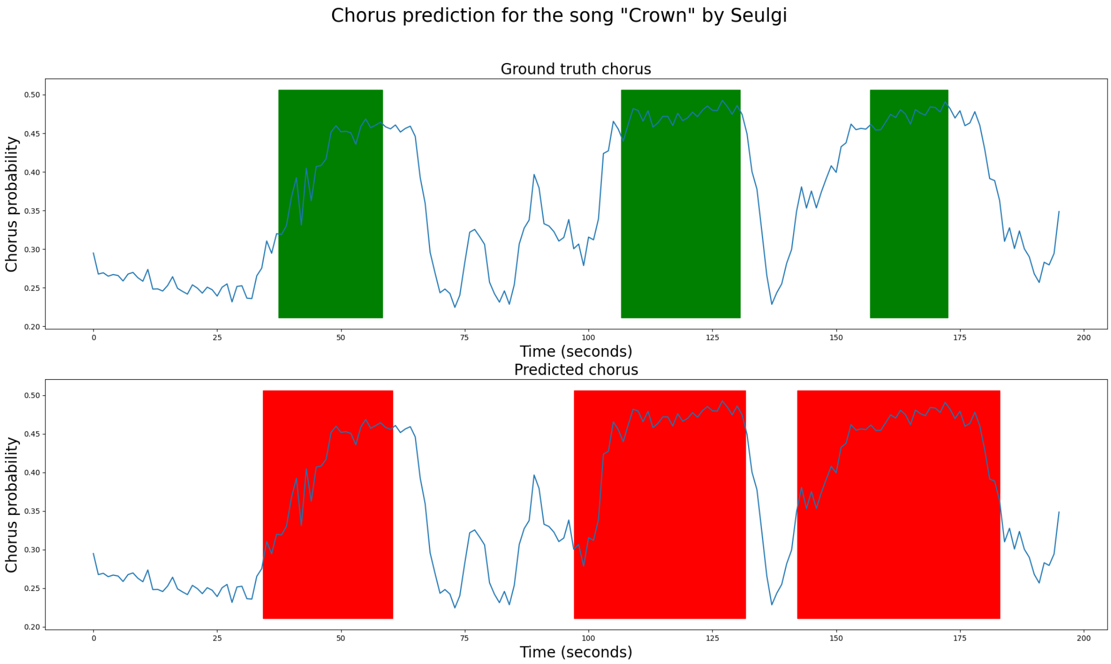


Figure A.2: Output of the changed model of DeepChorus for Crown by Seulgi.

The early results are promising, but more research needs to be done to corroborate the performance of the model. If it proves to be a reliable way of determining the chorus, features related to amount of chorus sections in one song, chorus length, and even the same types of features used previously (e.g. tempo, beats per minute) could be extracted just for the chorus section and studied to see if they would prove valuable for emotion classification. It would also provide a way of decreasing the work required to create segments for annotation for MEVD datasets.

A.3 Expansion of the MEVD dataset

To further expand the current MEVD dataset, it was chosen that 500 songs would be added onto the dataset. In order to facilitate the process, we started out by picking 125 songs of each quadrant that were already in the Static MER dataset. These songs were picked with maximizing genre distribution in mind.

Afterwards, since the Static MER only includes 30-second audio snippets, for each version the full song had to be downloaded. The search for the songs was conducted on YouTube with attention to pick songs from official channels, and to pick official audio versions (where applicable) and to download them in the highest quality possible.

Moreover, work has begun in annotating these songs in order to create the dataset. In the Static MER snippets, only one emotion is present in the snippet. However in the case of MEVD annotation, there is the possibility of a song having many different emotions.