Pedro Miguel Lima Louro

MERGE Audio 2.0

MSc Thesis

1 2 9 0

UNIVERSIDADE Đ COIMBRA

1 2 9 0

UNIVERSIDADE Đ COIMBRA

Pedro Miguel Lima Louro

MERGE Audio 2.0

MSc Thesis

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

September 2022

Faculty of Sciences and Technology

Department of Informatics Engineering

MERGE Audio 2.0

Pedro Miguel Lima Louro

Dissertation in the context of the Master in Informatics Engineering, Specialization in Intelligent Systems advised by Rui Pedro Pinto de Carvalho e Paiva and Renato Eduardo Silva Panda, presented to the
Faculty of Sciences and Technology / Department of Informatics Engineering

September 2022

1 2 9 0

UNIVERSIDADE Ð
COIMBRA

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

As I reflect on this year-long journey of extensive research, countless experiments made, hours upon hours in front of a computer to ensure the best work within my capabilities, and some random screaming to inanimate chunks of metal, all the people that accompanied me through this challenging ordeal come to mind.

I would like to begin my acknowledgments by thanking my advisors, Prof. Rui Pedro Paiva and Prof. Renato Panda, for the constant availability from both, the precious guidance and advice provided to more easily navigate the immense amount of works and ideas considered for this work, as well as the very thorough and constructive criticism on all aspects of it, making the process a very enjoyable and rich experience for me.

I thank from the bottom of my heart all the care, patience, and unconditional support provided by my family, my mother Fernanda, my father Paulo and my sister Rita, which despite the physical distance between us, helped me stay afloat when everything seemed to be breaking apart and pulled me up, never letting me give up on completing this project.

I would also like to give a special mention to Hugo, who dealt first hand with all my concerns and frustrations throughout the development of this work, did everything in his power to provide me with moments of fun and relaxation, as well as helping me stay aware of my goal even when it seemed impossible.

A very huge thanks to the rest of my family, for all the caring words you shared with me on our calls, as well to all my friends, who endured all my rants on my work and gave me encouragement to continue pursuing my dreams.

Thank you to the University of Coimbra and especially all teachers and colleagues in it that contributed to my academic journey. I would also like to express my gratitude to the Centre for Informatics and Systems of the University of Coimbra (CISUC) and, particularly, to the Cognitive and Media Systems (CMS) research group for the research scholarship I was granted. Also, I would like to acknowledge the MERGE project (Music Emotion Recognition - Next Generation , PTDC/CCI-COM/3171/2021), funded by Fundação para a Ciência e Tecnologia (FCT), which offered the contex for this work.

The developed work would not have been possible without the support I have received from all the people in this journey, as I fully believe that it would not have been possible without them.

Thank you all again for the support, for all the memories I will cherish for years to come, for believing in me even when I could not, and more importantly than all, thank you for always being there for me and making me a better person along the way.

This page is intentionally left blank.

Abstract

The library of digital music available to consumers has pushed for the growth of the Music Emotion Recognition research field, due to the need of organizing these large collections and provide personalized recommendations to listeners. The infancy of this field is dominated by Classical Machine Learning approaches using carefully constructed features to identify the perceived emotions of music pieces. Recently, there has been an increase in Deep Learning approaches in the field due to the ability of extracting the underlying features in the pieces, making the feature design step of the previous approaches automatic.

This work thus contributes to the field by providing an extensive set of experiments using a variety of approaches conducted on two datasets: the 4 Quadrant Audio Emotion Dataset (4QAED) dataset, previously developed by our team, and its extension, whose results are compared against for studying the impact in performance.

We obtained results above the state-of-the-art. Namely, a 80.24% F1 Score using an hybrid model, comprised of a Convolutional Neural Network, pre-trained on augmentated samples obtained using classical audio augmentation techniques, and a Dense Neural Network, pre-trained on extracted handcrafted features.

The developed worked also gave some insight in some promising directions, include further exploring Data Augmentation approaches and leveraging the information from multiple spectral representation to deal with the low amount of samples available in current state of the art datasets.

Keywords

Music Information Retrieval; Music Emotion Recognition; Machine Learning; Deep Learning; Data Augmentation.

This page is intentionally left blank.

Resumo

A biblioteca de música disponível digitalmente aos consumidores levou ao crescimento do campo científico de Reconhecimento de Emoção em Música, devido à necessidade de organizar estas enormes coleções e prestar recomendações personalizadas para os ouvintes. A infância deste campo é dominada por metodologias de Aprendizagem Computacional Clássica, utilizando elementos cuidadosamente desenhados para identificar as emoções percecionadas em peças musicais. Recentemente, registou-se um aumento de metodologias de Aprendizagem Profunda no campo devido à sua capacidade de extrair elementos relevantes nestas peças, tornando o passo de desenhar *features* automático.

Este trabalho contribui para o campo oferecendo um conjunto alargado de experiências utilizando diversas metodologias avaliadas em dois conjuntos de dados: o conjunto de dados 4 Quadrant Audio Emotion Dataset (4QAED), previamente desenvolvido pela nossa equipa, e a sua extensão, cujos resultados são comparados de forma a estudar o impacto no desenvolvimento.

Obtivemos resultados acima do estado da arte. Nomeadamente, um F1 Score de 80.24% utilizando um modelo híbrido, constituído por uma Rede Neuronal Convolucional, pré-treinado em amostras sintetizadas a partir de técnicas de sintetização de áudio clássicas, uma Rede Neuronal Densa, pré-treinada com *features* extraídas desenhados à mão.

O trabalho desenvolvido deu algum entendimento sobre as direções promissoras a seguir, incluindo continuar a explorar metodologias que utilizam sintetização de dados e utilizar a informação de múltiplas representações espectrais para lidar com o número reduzido de amostras disponíveis nos conjuntos de dados no estado da arte.

Palavras-Chave

Recuperação de Informação em Música; Reconhecimento de Emoção em Música; Aprendizagem Computacional; Aprendizagem Profunda; Sintetização de Dados.

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.

A-V Arousal-Valence.

ABC Artificial Bee Colony Algorithm.

AUC Area Under the ROC Curve.

BCRFM Bidirectional Convolutional Recurrent Feature Maps.

BCRSN Bidirectional Convolutional Recurrent Sparse Network.

BiGRU Bidirectional Gated Recurrent Unit.

BP Backpropagation.

BPTT Backpropagation Trough Time.

CCC Concordance Correlation Coefficient.

CNN Convolutional Neural Network.

CRNN Convolutional Recurrent Neural Network.

DCGAN Deep Convolutional Generative Adverserial Network.

DEAM MediaEval Database for Emotional Analysis of Music.

DL Deep Learning.

DNN Deep Neural Network.

EA Evolutionary Algorithms.

EEG Electroencephalogram.

FCN Fully Convolutional Network.

FCNN Fully Connected Neural Network.

GA Genetic Algorithm.

GAN Generative Adversarial Networks.

GEMS Geneva Emotional Musical Scale.

GMM Gaussian Mixture models.

GP Gaussian Process.

GRU Gated Recurrent Unit.

ICASSP International Conference on Acoustics, Speech, and Signal Processing.

IR Iterative Reconstruction.

KNN K-Nearest Neighbour.

LSTM Long-Short Term Memory.

MAE Mean Absolute Error.

MEM MediaEval Emotion in Music.

MER Music Emotion Recognition.

MEVD Music Emotion Variation Detection.

MFCC Mel-frequency cepstrum coefficients.

MIR Music Information Retrieval.

MIREX Music Information Retrieval Exchange.

ML Machine Learning.

MSD Million Song Dataset.

MTAT MagnaTagATune.

NN Neural Network.

PCC Pearson Correlation Coefficient.

ReLU Rectified Linear Unit.

RF Random Forest.

RMSE Root Mean Squared Error.

RNN Recurrent Neural Network.

SCAE Sparse Convolutional Autoencoder.

SER Scene Emotion Recognition.

SER Speech Emotion Recognition.

SII-ASF Sequential-Information-Included Affect-Salient Features.

SMOTE Synthetic Minority Oversampling Technique.

SN Siamese Network.

STFT Short-time Fourier transform.

SVM Support Vector Machine.

SVR Support Vector Regression.

TR-IR Time-distributed Layer with Iterative Reconstruction.

WCMED-CCMED Western Classical Music Excerpts and Chinese Classical Music Excerpts.

WHBR Weighted Hybrid Binary Representation.

Chapter 1

Introduction

Music has always been present throughout human history. From ancient rituals, to commemorations after a battle, from recreative events to manifestations, it has been a crucial tool for humans to express themselves. It can deepen the bonds between us, conveying the emotions we are feeling to our loved ones, or take us to a journey of introspection, helping us cope with hardships. Due to this, many believe that music is the language of emotions, the perfect vehicle to express even the most complex feelings. However, the amount of music available to one person for most of our history has been limited to prior knowledge of artists and, to be able to experience it, music needed to be purchased to be enjoyed alone or left for social events, severely limiting the exposure to a myriad of new ideas.

The digital era completely changed the scene, making large libraries of music available to various platforms at the fingertips of users through streaming platforms, making it possible to easily find new artists and genres as well as sharing the right music at any time, anywhere. However, the sheer amount of available content evoked the necessity for ways of organizing these large digital libraries and providing more user-friendly ways of searching and personalizing the consumption of such content.

There have been countless ways of organizing music, be it from the most common descriptors for music, including genre, artist, era or even geographical location, to custom made playlists, created by the platform's users or curated by music specialists, catering to every possible necessity. The problem with these methods is the underlying necessity for human input, adding the necessary descriptors or handpicking songs.

The Music Information Retrieval field has for a long time focused on automatically identifying prominent music features, applying these to the likes of source separation, and automating the categorisation process of music, making it possible to identify and generate sets of songs following a discernable pattern. From the possible descriptors that can be used to this end, emotion has gained great interest due to the intrinsic connection with music. A major drawback of objectively classifying emotion is the very subjective nature of it. A song that evokes a sad feeling in an individual, may evoke a calming or happy feeling in another.

Many psychology studies over the years have tried to categorize the array of human emotion in a general sense, many of them being focused on musical emotion. These have all been disputed by various reasons, and to this day there is not a consensus on a model that encompasses the entire human emotion spectrum, demonstrating the abstract and complex nature of emotion.

Despite this, many efforts have been made to further delve into the intricacies of what makes a music piece convey a particular emotion, due to the immense personal, social, economic, commercial and scientific impact of this area. As such, recently the field of Music Emotion Recognition (MER) emerged as a subfield of MIR with the goal of researching and providing solutions for categorizing musical pieces based on their emotional content.

1.1 Problem, Motivation and Scope

The MER field contributes to the landscape of automatic music categorization by providing solutions such as music recommendations systems, automatic playlist generation, as well as providing solutions for a myriad of industries, including advertisement, cinema, video games, as well as health application, such as music therapy for emotional disorders [1].

To automatically recognize emotion in a musical piece, there is a need to extract relevant features from the audio signal of the song and proceed to train an algorithm for conducting the classification process of the signal into a certain emotion. The feature extraction processes in most of MER literature are based on various metrics or estimations, directly applied to the signal, for example, estimating the fundamental frequency of the sound to extract the pitch of the melody or estimating the tempo which marks the rhythm of a song. These features are normally applied to Classical Machine Learning (ML)[1] algorithms, such as Support Vector Machines or Random Forest, either as regressors, predicting continuous arousal-valence (A-V) values[2] in the emotional plane (see Section 2.1.2), or classifiers, either predicting one or multiple-labels for a piece, depending on the emotional model in question.

There are many issues regarding the whole field of MER, such as the annotation process for building a dataset, which requires enormous effort and labor from human subjects, most of the times not experts in the field, leading to long validation processes for the collected data. Not only this, due to the copyrights of the songs present in the datasets, most of the times it is not possible to fully make a dataset available, resorting to referencing the utilized songs, leaving other researchers with the task of retrieving them if they wish to further experiment with the dataset. Another issue is the capability of the employed methodologies, as the combination of a lack of new emotionally relevant features that cover crucial musical dimensions for classification, as well as the absence of large and quality datasets available to the general public, led to the current results reaching a glass ceiling for Classical Machine Learning methodologies. Various other approaches have emerged to solve this last issue, such as personalized recommendation systems [2], utilizing physiological and brain signals [3] [4], or combining both audio and lyrics [5] to attain more accurate results.

As a way to combat the lack of features, the MER field has recently seen an increased interest on Deep Learning approaches to automate the feature extraction process. Not only this, the overall better performance over Classical Machine Learning approaches in other fields made a push for exploring these approaches. This would not only cut out the time and resource consuming task of designing new features, but the automatically extracted features may also be more accurate, since these are directly extracted from the data, with the potential of more accurately identifying the key components for classification.

[1]This term is used throughout this work when referring to methodologies that do not use any type of deep neural network, i.e., neural networks with one or more hidden layers, most of the time using simple classifiers.

[2]"Arousal (or intensity) is the level of autonomic activation that an event creates, and ranges from calm (or low) to excited (or high). Valence, on the other hand, is the level of pleasantness that an event generates and is defined along a continuum from negative to positive.", https://goodmancoaching.nl/definition-of-valence-arousal-and-dominance/

As such, this dissertation aims to contribute to the MER field by exploiting the potential of deep learning approaches, as discussed in the next section. The dissertation was conducted in the scope of the MERGE project[3] (Music Emotion Recognition - Next Generation, PTDC/CCI-COM/3171/2021), funded by Fundação para a Ciência e Tecnologia (FCT), which offered the context for this work.

1.2 Objectives and Approaches

As abovementioned, this work aims to exploit DL approaches to MER.

Within MER, two problems have deserved particular attention: Static MER and Music Emotion Variation Detection (MEVD).

Static MER focuses on automatically tagging the overall emotional content of a song, either to single or multiple labels, based on the provided annotations. For the scope of this work, the focus is to explore static single-label audio classification for a reduced number of core emotion classes, namely, 4 quadrants in a 2D emotion A-V plane. In addition, but to a lesser extent, emotion regression is also tackled with the prediction of continuous A-V values. MEVD also focus on automatically tagging the emotional content of a song, however, the intent is to predict the variation of the emotion along the duration of the song.

This work aims to contribute to the MER field by:

- updating the existing 4 Quadrant Audio Emotion Dataset (4QAED) [6], previously developed by our team, as well as validating the resulting extension to the dataset;
- replicating and validating previous work by Pedro Sá [7] on static MER, a former MSc thesis student of the MERGE team;
- improving these models with the reviewed State of the Art approaches. This requires a comprehensive analysis of the most promising approaches for Static MER, namely, different DL architectures, Transfer Learning, Data Augmentation, etc.

Research, development and evaluation of current approaches mostly focused on Deep Learning to address the lack of relevant features. Moreover, strategies to cope with the reduced number of samples available in the dataset were studied.

The present work is a continuation of the work conducted by Pedro Sá for his Master's thesis on the same subject. For this reason, there is a need to first replicate and corroborate the findings of this work before proceeding to build upon it. As stated in the Replicated Work section (Section 4.1), it is not possible to accurately replicate the exact environment used to obtain the reported results, since there are no details on the library versions used for Python, the main language used in past and present work, and there was no seed set for the pseudo-random number generator, which is used in many aspects of the experiments, from preparing the cross-validation folds for training and testing, to shuffling the data for training a model, and even the optimization steps for the model, since this is based on stochastic methods of gradient optimization. With an impossibility of truly replicating the results, fluctuations between reported and obtained results are inevitable. As a way to mitigate this problem in the future, virtual environments are created and maintained to

[3]https://www.cisuc.uc.pt/en/projects/MERGE

accurately reproduce the behavior of the tools used, as well as setting the same seed for any pseudo-random number generation.

Plans for exploring MEVD architectures were dropped to low priority due to the sheer amount of work from the static approaches, time and resource constraints, and lack of new data. This last point was the most decisive of the three, since it was already previously concluded that it is necessary to considerably increase the publicly available dynamic dataset, which comprises only 29 songs with dynamic annotations, for any significant increase in the dynamic approaches' performance.

The initial planned tasks for the first and second semesters were ranked in a priority scale of *High*, *Medium* or *Low* indicating their importance for accomplishing the main objectives of this work.

Table 1.1: Objectives of the presented work.

Objective	Priority
Replication of previous work	High
Review and implementation of SoTA architectures - Static MER	High
Development and evaluation of SMOTE for Data Augmentation (Deep SMOTE)	High
Review and evaluation of pre-trained models for Transfer Learning (Artists, Same Domain)	High
Static Database validation (New Extended Dataset)	Medium
Development of Ensemble approaches (Hybrid with Augmentations, CNN with Segment-Level Raw Branch)	Medium

As to thoroughly explain the extent of this work, fulfilling the project's objectives should answer the following research questions:

- Are the results presented in previous conducted work replicable?
- Do recent state of the art approaches perform better than previously tested approaches on the same data?
- Are the new extended dataset candidates viable?

The rest of this section will elaborate on these research questions and approaches to solve them, as well as changes made to the initial objectives and the reasons behind them.

Following the replication of past work, we focused on the aforementioned new state of the art approaches experimented on the available data. Besides of a variety of CNN and RNN based architectures tested with the current publicly available dataset from the team, which include the ShortChunk and Sample architectures, as well as the CRNN architecture, other approaches, some as continuation of previous work and others as promising new methodologies in regards to the dataset in question, are explored. Data Augmentation, with new classical audio augmentation techniques and SMOTE, and Transfer Learning, with different and same domain data, are both expanded upon, while Deep Embeddings, with pre-trained networks and a data-driven approach, and Multiple Representations, with full and segment-level samples, are evaluated for their viability to more accurately solve MER.

Ensemble methods are also further explored, further improving on an existing architecture and exploring a two-branch spectral and end-to-end architecture.

A key requirement of DL is good and sizeable data. To this end, another objective of this work was to extend the current Static MER dataset (henceforth termed Legacy-MERGE), creating a larger collection (henceforth termed New-MERGE). The new dataset was then evaluated by applying the developed and evaluated approaches on their candidates.

1.3 Results, Contributions and Limitations

In this section, the most relevant results are presented, as well as the contributions of this work and the limitations encountered during the course of this work.

The main results obtained were:

- Significant improvements to the baseline model utilizing classical audio augmentations not previously explored;
- Overall improvements to F1 Score results using the New-MERGE dataset candidates across many experiments;
- An **74.23% F1 Score** with the Sample-level Multiple Representation methodology when using the New-MERGE Balanced-Genre dataset candidate, a result on par with the state of the art Support Vector Machine (SVM) approach (76.40% F1 Score, as reported in [8]);
- **80.22%** and **80.24% F1-Scores** were obtained using the Hybrid CNN with Augmentations + DNN using the New-MERGE Balanced and Balanced-Genre datasets respectively, a considerable improvement over the state of the art SVM approach.

As for the contributions made with this work:

- The validation of the New-MERGE dataset, which performed on par with Legacy-MERGE without any optimization, indicating its viability;
- A very extensive set of experiments conducted on both the Legacy- and New-MERGE datasets;
- Preparation for a comparison article using the collected data for this thesis.

To end this section, the limitations encountered during the development of this work:

- The small dataset sizes greatly impacted the performance of the DL approaches experimented with;
- Missing emotion tags for some of the songs on the New-MERGE dataset considerably reduced the available datasets for methodologies using mapped A-V values;
- An oversight also reduced the available samples for the New-MERGE dataset due to missing extracted handcrafted features for some of the songs.

1.4 Organization, Planning and Resources

This section presents the experimental environment where this work was conducted, as well as an overview of the planned tasks, allocated time and a reflection on the actual project's development.

1.4.1 Experimental Environment

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

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

Despite the resources available, the GPUs of the server were seldom free to conduct experiments, rendering the server a lot less useful. For this reason, it was decided to use Google Collaborator in tandem with the server. With a small fee, the following resources were available:

- Intel Xeon Silver 4214 CPU @ 2.20 GHz x 48
- NVIDIA P100 PCIE 16GB/T4 16GB
- Maximum of 32GB System RAM
- Maximum 24 hours runtime

There were initially a lot of difficulties taking advantage of these resources, mainly due to the fact that the time a runtime is alive varies with no discernible reason. An experiment could run the full 24 hours or 30 minutes, with no difference between them. Google does not disclose the reasons for terminating a runtime, justifying it with user inactivity and/or high server load.

Later, 3 NVIDIA RTX A5000 24GB GPUs were added to the server, as well as an increase in RAM, totalling 256GB at the time of writing. The increased resources helped to increase the throughput of experiments conducted, with some hiccups along the way due to the continuous heavy use by various users.

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. For implementing the approaches, *keras*, *tensorflow* and *pytorch* were utilized to build and train DL models, while *scikit-learn* was utilized for some Classical ML approaches and calculating the relevant metrics for analyzing the approaches performance. Other relevant libraries are noted in the section where the approaches utilize them.

1.4.2 Organization

Gantt charts are presented as a contrast between the planned timeline and real effort needed for completing each task, ending with a brief discussion on the reasons for the changes.

First Semester

It was decided that the first two months of the first semester would be focused on the literary review for the creation of a comprehensive State of the Art, as well as acquire information necessary for accomplishing the objectives of the project. The rest of the first semester would be focused on getting to know the available MER datasets, as well as replicating the previous work conducted with the available datasets, in order to get familiar with the tools to be used during the development of the new DL models. The experiments would also be applied to a corrected version of the same datasets, to access possible improvements in the performance of the models.

During the period focused on the literature reviewed, there was also some familiarization with the server where the experiments would be conducted, as well as the setup necessary for performing the experiments. Due to technical problems, familiarization was more limited than expected.

The review of the State of the Art took longer then initially expected due to lack of planning and knowledge on my end. The experiments also took longer then expected, due to issues regarding the necessary packages to replicate the abovementioned work and unavailability of the server. This led to a severe reduction of the conducted experiments, the remaining being adjusted in the planning to be conducted in the beginning of the second semester.

The focus of the whole semester became to develop a good foundation on the various approaches available to solve MER, as well as exploring new ML techniques for coping with the reduced data in the explored datasets. Replication of previous work was started, ensuring a foundation of the main tools for conducting all future experiments in the next semester.

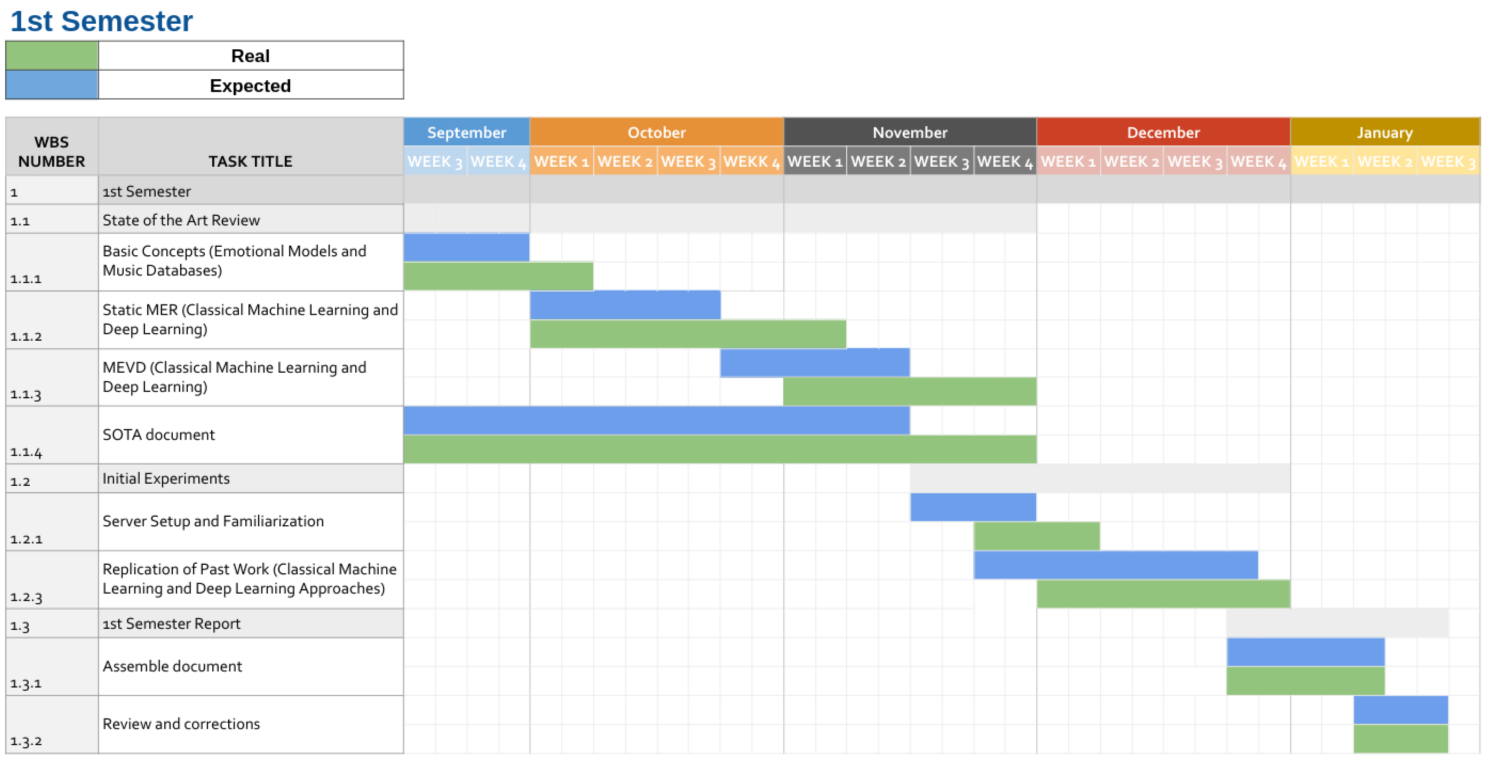
The Gantt chart of the estimated effort for each task at this point can be seen in Fig. 1.1.

Second Semester

For this semester, most of the tasks were already established with respective estimations of the time they would take to be completed. These estimations, as can be seen by the final Gantt chart, were severely underestimated due to a variety of factors. The experimentation on new methodologies for Static MER had to be pushed back due to resource difficulties, pushing replication work well over the initial three weeks estimated for conclusion to around seven weeks. All these events greatly increased the time needed. Despite this, these difficulties led to a more thorough review of the literature, leading to a lot more possible methodologies to be followed later.

With this push, MEVD experimentation was dropped to a low priority and evaluating the extended dataset viability was raised to medium priority, since it made more sense with the predicted remaining time and would also pertain a more substantial contribution to the field. At this point, the tasks at hand for the approaches to be explored were set, dividing the time for development and evaluation of Static MER approaches between each. Unfortunately, due to underestimating of the problems that would arise from implementing the approaches and more resource difficulties, each experiment took longer than expected, and this led to the delivery of the present work.

The tasks' timeline was again adjusted, dropping the Few-Shot Learning approaches, due to being the least promising of the lot and MEVD experimentation, due to the already discussed reasons. Finally, the timeline for evaluating the dataset extension was set and the resulting version of this planning can be seen in Fig. 1.2.



WBS NUMBER	TASK TITLE
1	1st Semester
1.1	State of the Art Review
1.1.1	Basic Concepts (Emotional Models and Music Databases)
1.1.2	Static MER (Classical Machine Learning and Deep Learning)
1.1.3	MEVD (Classical Machine Learning and Deep Learning)
1.1.4	SOTA document
1.2	Initial Experiments
1.2.1	Server Setup and Familiarization
1.2.3	Replication of Past Work (Classical Machine Learning and Deep Learning Approaches)
1.3	1st Semester Report
1.3.1	Assemble document
1.3.2	Review and corrections

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

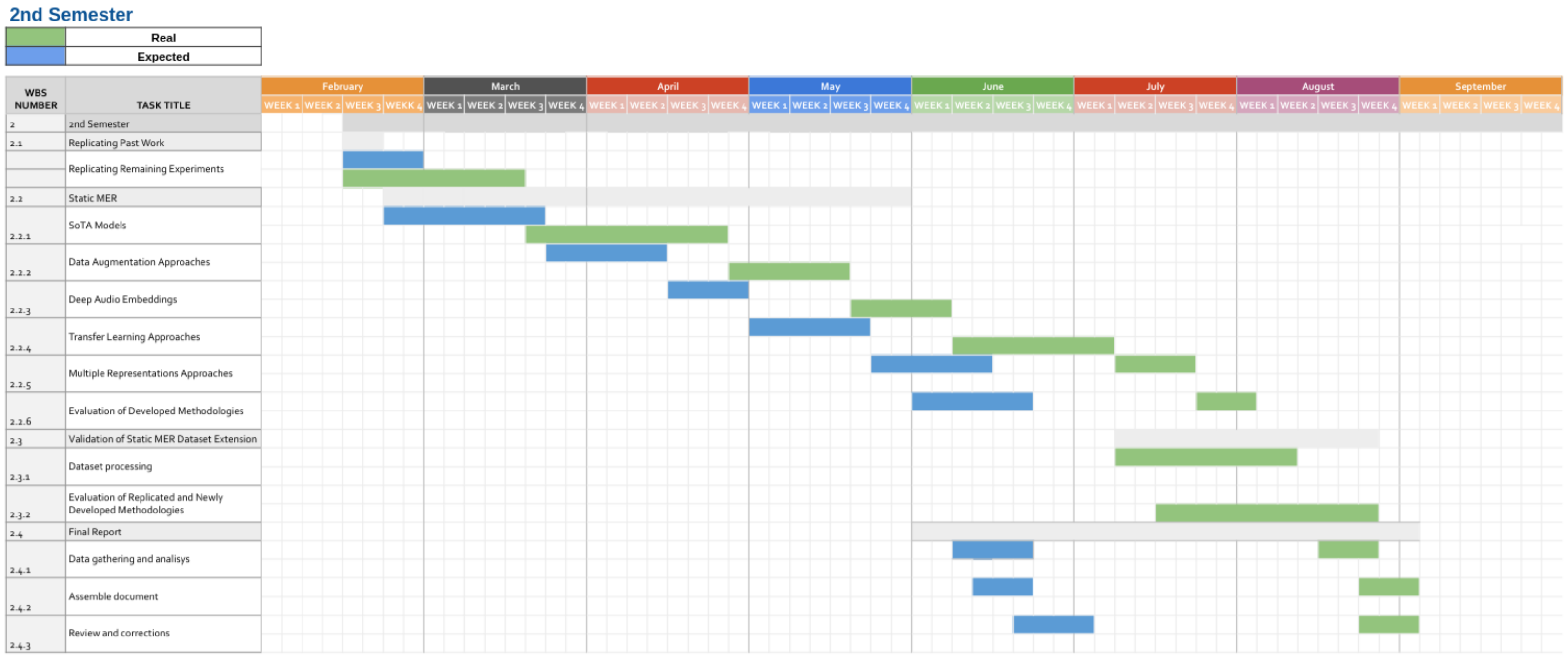


Figure 1.2: Estimated and real effort for the second semester.

1.5 Outline of the Thesis

The document is organized as followed.

Chapter 2 presents a review on the state of the art of the MER field and relevant concepts. In this chapter, the employed emotion taxonomies in the literature (Section 2.1) is followed by a review of existing public databases focusing on music (Section 2.2), as well as a background on key machine learning concepts to understand the reviewed approaches (Section 2.3), following with the reviewed approaches for Static MER (Section 2.6) and MEVD (Section 2.7).

Chapter 3 presents both studied datasets, Legacy-MERGE and New-MERGE, followed by a brief explanation of the relevant metrics used to evaluate the performance of the models on the aforementioned datasets.

Chapter 4, the core chapter of this thesis describes the methods and experiments performed. Unlike the typical organization on two chapters (one for methods and another one for experiments), we preferred to describe each method immediately followed by the performed experiments due to: i) the larger number of performed experiments; ii) and the logical sequence in each they were performed.

Finally, the conclusions from this thesis and a brief discussion on the possible directions for future work are presented in Chapter 5.

Chapter 2

Background Concepts and State of the Art Review

This chapter has the objective of providing some exposure to core concepts of MER models, while discussing the different approaches to them, and a review and critical discussion of the research accomplished in the area.

2.1 Emotion: Basic Concepts and Representation Models

In this section we define emotion, as well as the different types of emotions: perceived (the focus of this thesis), expressed and induced.

2.1.1 Emotion Concepts

Although inherent to humans, emotion as a concept is very hard to accurately describe. A definition proposed by Kleinginna et al. [9] states that "Emotion is a complex set of interactions among subjective and objective factors, mediated by neural/hormonal systems, which can (a) give rise to affective experiences such as feelings of arousal, pleasure/displeasure; (b) generate cognitive processes such as emotionally relevant perceptual effects, appraisals, labeling processes; (c) activate widespread physiological adjustments to the arousing conditions; and (d) lead to behavior that is often, but not always, expressive, goal-directed, and adaptive." Emotion can be briefly summarized as rapid changes in the disposition of an individual, most of the times occurring in small intervals of time.

Not only is emotion highly subjective, it can also be expressed, perceived or induced. These are the emotion the composer and/or performer aim to transmit in a song, the emotion identified in a musical piece and the emotion evoked by the song in the listener, respectively [10]. Some important points in these different types of emotions are the possible independence of perceived emotion from the other types, the same may be stated for induced emotion, and that perceived and induced emotion are not easily distinguishable, due to the paradox of negative emotions, for example, sadness and sorrow being considered as enjoyable [11].

Perceived emotion is the focus of this work, nonetheless, there are works focused on induced emotion.

2.1.2 Emotional Models

A large problem in MER is the annotation methods used to build databases due to the high subjectivity inherent to emotions already discussed above. To extract relevant and accurate information from the annotations of humans, and in the process reduce ambiguity, emotional models are used. These can be categorical, where emotions are aggregated in clusters and music is classified with tags corresponding to one of those clusters, or dimensional, where emotions are in a discrete or continuous space, mostly two-dimensional, and music is classified with a value for each dimension.

There is a large debate about which type of model is better [12], both being used in the literature.

Next, the most prominent models in recent literature are discussed.

Categorical Models

These models are based on the basic emotions, a concept introduced by Ekman [13], which states that emotions are distinct from one another and can be defined as categories. These basic emotions are happiness, sadness, anger, fear, disgust and surprise. They seem to be linked to the key aspects important to our evolution, although this has been disputed, as some authors have found different sets of emotions.

Hevner's Adjective Circle

Based on these basic emotions, Hevner[14] developed an adjective circle which clusters similar emotions into groups. There are 8 different groups of adjectives, having a range of 6 to 11 emotions in each, totaling 67 adjectives. Neighboring groups have higher similarity compared to more distant ones.

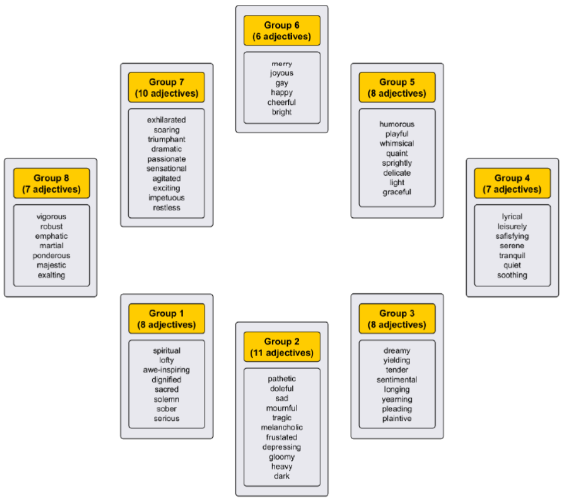


Figure 2.1: Hevner's Adjective Circle [14].

Despite encompassing a wide array of the emotion spectrum, this approach as some problems, namely the clusters being unbalanced, having varying number of adjectives, and being developed for classical music in mind.

This model would later be revised to 10 groups by Farnsworth [15], and 9 groups by Schubert [16].

Geneva Emotional Musical Scale Model (GEMS)

This approach [17] aimed to study induced emotions in listeners and develop a categorical model for music emotion classification. Four different studies were conducted in order to build a comprehensible list of relevant affect terms.

The first two studies focused on collecting the known adjectives from the literature and validating them. The first study, which was conducted with 92 psychology students, narrowed down the list from 515 to 146 terms. The second study asked 262 undergraduate psychology students for their music genre of choice. Three questions were made to the students that chose a preference in order to attest how often they felt and perceived a given emotion when listening to that genre, as well as the frequency it was experienced in extra musical everyday life. The final result produced a list of 66 relevant terms. The third study focused on extending the aim of the second and also examine the possibility of dividing these terms into sub-units and the last study focuses on finding whether the term list is more accurate at describing and lessing the subjectivity of the reported emotions.

The final GEMS comprises 45 terms, divided into 9 categories, proven to be relevant and encompassing. Two shorter scales comprising of 25 and 9 terms were also created.

However, there are some aspects that put in question its representativeness of the various genres. Five genres were chosen as the subject of the study meant to encompass the remaining genres, which is very difficult to achieve due to the huge amount of musical genres. The samples chosen are also very limited, not only in number, 16 in total, but also in genre, focusing exclusively in classical music.

Dimensional Models

Dimensional models regard emotional experiences as “a continuum of highly interrelated and often ambiguous states” [18]. Here, a multidimensional space is employed, mapping different emotional states to points in that space.

Russel’s Circumplex Model

Assuming the presence of multiple stimuli as necessary for the existence of emotion, Russell[19][18] proposed a two-dimensional emotional model, comprised of valence, which quantifies how pleasurable the emotion is, and arousal, quantifying its namesake. The second dimension can also be referred as activity.

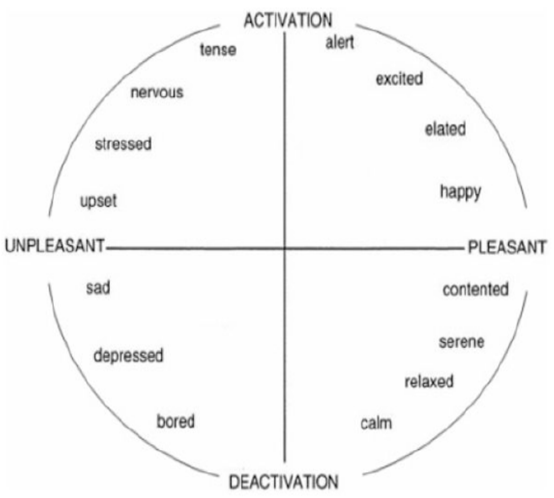


Figure 2.2: Russel’s Circumplex Model [18].

According to the two aforementioned axes (arousal and valence), this model is composed

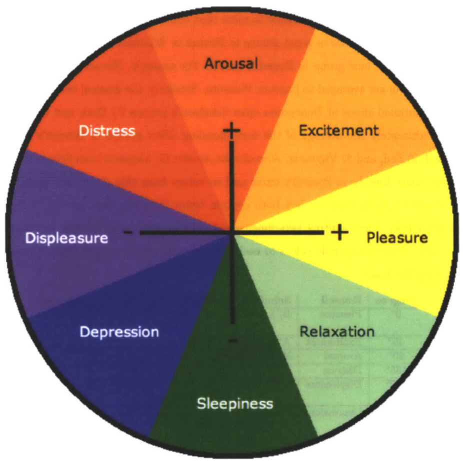


Figure 2.3: Meyer's Discrete Dimensional model [20].

by four different quadrants, mainly defined in the literature as: Q1 - exuberance, happy and energetic emotions; Q2 - anxiety, frantic and energetic emotions; Q3 - depression, melancholic and sad emotions; Q4 - contentment, calm and positive emotions.

Despite some identified shortcomings, for example, placing emotions with distinct meanings next to each other, this has been the most widely used model in MER literature [12]. Alternative models using two main dimensions to classify emotion have been proposed in the literature, one of these being the Thayer's model of emotion, as cited in [20]. The concept of arousal is now divided into energetic arousal and tense arousal and these are defined by rotating the Russel's circumplex model 45 degrees.

Meyer's Discrete Dimensional Model

Another approach used by interpolating Russell is presented by Meyer [20]. There a circumplex model divided into 8 equal parts, each one characterized by it's arousal and valence signal, is presented.

The reasoning behind making the two-dimensional plane discrete is due to it being more comprehensible to annotators and greatly reducing the computational cost.

The resulting sections are: arousal, excitement, pleasure, relaxation, sleepiness, depression, displeasure and distress.

This is an example of discrete dimensional models, which are widely used since they are more comprehensible to the annotators and used as the emotional taxonomy of datasets [6].

Three-dimensional models have been proposed over the years, such as the Schimmack and Grob model, Fig 2.4, which fuses the Russell and Thayer models to produce a more comprehensive emotional plane, or the Tellegen-Watson-Clark Model, Fig 2.5, which is based on a hierarchy that takes into account the phenomena of sad songs having a perceived happy feeling, as well as separating positive from negative affect, being judged separately. Despite solving some issues with the models discussed before, these models have not been studied in the context of MER, since these are more complex and would require more resources to train models that adopt these taxonomies, and knowing that two dimensional models are already complex for human annotation, introducing another dimension would add even more level of complexity, leading to lower quality annotations.

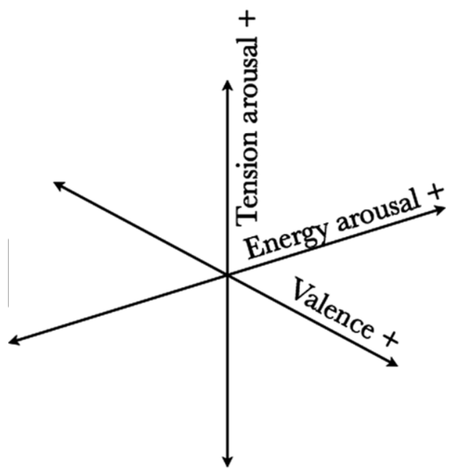


Figure 2.4: Schimmack and Grob model, from Eerola et al. [12].

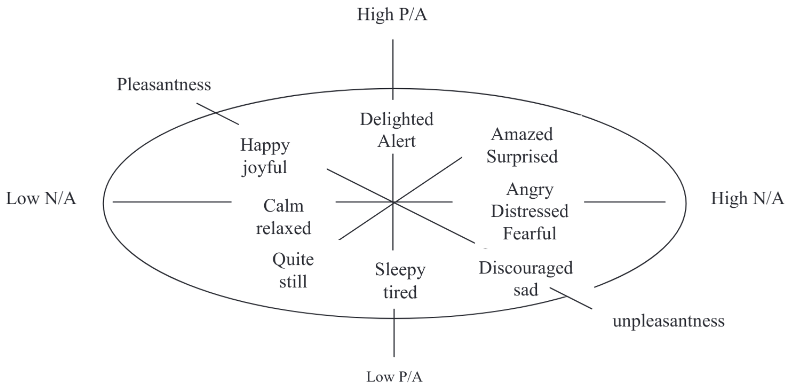


Figure 2.5: Tellegen-Watson-Clark model, from Trohidis et al. [21].

2.2 Music Emotion Databases

A necessary element in any approach for solving MER are data samples annotated with the corresponding emotion. These data samples are normally samples of full songs which represent the entire song content should the approach focus on solving Static MER, or various samples of a song that represent the emotion at a certain interval for solving MEVD. These samples also need to be humanly annotated, also called ground-truth data, which is normally a very labor-intensive chore, conditioning the size and the quality of the annotations, due to the number of annotators and their music expertise respectively. Finally, features also need to be extracted from these samples, which can vary from a few features to hundreds [22]. Various authors have assembled the aforementioned elements into databases, some private and others released to the general public, enabling other researchers to use these in their one studies. The databases may not always provide all the elements as they are. The samples may not be present due to copyright reasons, instead the songs and the interval used for the sample are annotated, or the features are not present, needing to be extracted with audio feature extraction tools.

Another important point is the quality of the database itself. There are various factors that impact the quality, including: the number of annotations per sample, the number of samples in the database, how balanced are these according to genre or distribution across the emotions in the used emotion taxonomy, how the data is collected and the agreement rate of annotators. Due to the subjective nature of the emotion evoked by music, it is important to understand how the samples are annotated and how the agreement is measured, to mitigate this issue as much as possible. Databases that do not take annotators agreement in account, such as very large datasets mostly composed of user submitted tags, have considerably less quality due to the inconsistencies of the annotated emotions and the tags themselves may not be related to emotions. Despite this, a very large quantity of samples can provide several significant features for the classification process.

These databases also range from completely varied, such as in genre or artists, to a very focused database, such as including only classical music, as can be found throughout MER literature, or focus on popular music from select languages, such as a English and Chinese popular music dataset.

Since MER strives to accurately classify the emotions present in music to facilitate the search and management in music listening in general, this project will mainly focus on datasets that provide variety on the genre, time-frame of release and artists, to more accurately represent the variety found in music.

This section will provide an analysis of publicly available datasets, first analyzing each database's characteristics, followed by a table presenting the key points of each one.

CAL500exp

The CAL500exp dataset [23], as its namesake implies, is an expansion of the CAL500 dataset composed of segments of the original music clips that are within a range of 3 to 16 seconds each. This segmentation was performed by using an algorithm that splits the music piece according to the acoustic homogeneity in a certain interval. These segments were then linked to the corresponding tags, which were also trimmed down from the original CAL500. The process was carried out using 11 music experts annotations, obtaining a total of 67 out of the original tags, where freely user created tags could be added. This makes it more consistent than the original dataset, since all tags were expert picked.

MTAT

The MagnaTagATune, or MTAT, is a result of the annotations obtained from the TagATune game for evaluating music tagging algorithms presented by Law et al. [24]. The tags are aggregated by matching two human players, or a bot in case matching is not possible, and asking them to tag the tune they are hearing as they desire, followed by asking each player to assess whether the other player's song is the same or different using only the submitted tags, which should ensure that only the most relevant tags for a song are kept.

The dataset itself contains 25,877 audio clips, each 30 seconds in length, and the corresponding multi-label annotations, including common tags such as genre and decades as well as others including mood, instruments and some related techniques (ex: plucking), however, from the 168 available tags, only the top 50 are normally used [25]. The data is divided into 16 folders and the normal train/validation/test split is 1-12/13/14-16, but other splits exist.

The annotation process, although not as controlled and no agreement rate is provided, provides better quality compared with other datasets, such as MSD, however, it does not provide a great diversity of artists, which may limit the generality of models trained on it.

Million Song Dataset

The Million Song Dataset [26] aims to provide a large-scale dataset for the MIR field, adressing the problem of the smaller public datasets available not being very suitable to demonstrate the scalability of proposed approaches.

The dataset is an aggregation of multiple smaller datasets compilled from different sources, such as The Echo Nest, MusicBrainz, MusixMatch and more. The annotations are tags freely inputed by the large communities of listeners in the relevant services, which may not accurately represent the song track, since common listeners may rashly input the first impression of a track in the system without properly assessing it.

Bi-Modal

The Bi-Modal database, proposed by Malheiro et al. [27], is composed of various audio samples with their corresponding lyrics, with the intent of achieving better performance by combining the two, something out of the scope of this project. The audio samples can be used independently, meaning that audio based approaches for solving MER are possible with this database. The dataset was constructed by first collecting 200 music samples and their corresponding lyrics spanning different music genres and eras. The annotation phase was carried out with 39 annotators with diverse backgrounds, where each classified either the audio or the lyrics based on their emotional content, assigning numerical values on a discrete Russel's A-V model for both valence and arousal, ranging from -4 to 4. After evaluating the standard deviation of the annotations between annotators, the agreement between and whether the audio and lyrics annotations for the same song match, the database was reduced to 133 samples. This database has a reduced number of samples, which is a recurring negative point of databases in MER. Due to its small size, the discrepancy of the samples representing each quadrant is more noticeable and therefore, more unbalanced than at first.

DEAM

DEAM, presented by Aljanaki et al. [28], a database which in reality is a combination of the three databases developed for the 'Emotion in Music at the MediaEval Evaluation Campaign' between 2013 and 2015 for model performance evaluation. The database is comprised of 1802 samples, each 45 seconds in length, with both static and dynamic annotations, according to the continuous two-dimensional Russel's A-V model, meaning that the database can be applied to both Static MER and MEVD. For the dynamic annotations, each sample has an emotional label every 0.5 seconds. The samples are extracted from the original songs at random starting points, which raises some concerns, such as the sample not representing the emotional content of the song or being completely silent. The mean agreement rate for this dataset is around 0.47 [7], meaning that annotators did not agree in most of the songs of this dataset. The annotations were a joint effort from the authors and crowdsourcing through Amazon Mechanical Turk, for which a method was proposed to filter out poor quality annotations. Due to the low agreement rate, a selection of samples should be conducted to keep the most consistently annotated samples.

PMEmo

PMEmo, proposed by Zhang et al. [29], is a database composed of 794 annotated songs along with the electrodermal activity of the annotators. The process of collecting electrodermal activity signals from annotators will not be discussed since it is out of the scope of this project. The construction process of this database started by fetching 1000 English pop songs from various United States and United Kingdom music charts. After removing duplicates, which includes songs that were already present in the collection but added other artists, it was reduced to 794 songs and samples of various sizes representing the chorus of each song were extracted. 457 subjects, mainly Chinese students not majoring in music, conducted the annotation process, which began by training the subjects to understand the emotional model used, discrete Russel's A-V model, and get familiar with the interface. After relaxing with the help of a song, each subject annotated both valence and arousal for 20 excerpts of which 1 was a duplicate. This is stated to increase the quality of the annotations, as proposed in [30], since the annotations are only accepted if the bias between the values of both clips are low enough (within 0.25 in the A-V space). By removing these annotations, each music clip has annotations by at least 10 different subjects. Finally, the agreement between annotators is evaluated, which produces very good results, indicating a very high agreement rate. This database has a decent size in comparison to others in the literature, but focuses heavily on a single genre and a very specific era, only two years. This may explain the high results obtained in the experiments with this database.

MTG-Jamendo

The MTG-Jamendo dataset, introduced by Bagdanov et al. [31], includes 55,701 full length songs, each having at least a duration of 30 seconds, extracted from the Jamendo[1] music platform, composed of copyright free music provided directly by artists. Each track has 692 associated tags provided by the music's author including genre, instruments, moods and more, which were submitted to a preprocessing step. There are also five different splits provided by the authors, from which the first one is recommended to be used for music auto tagging, as well as providing the 50 most frequent tags.

Due to being very recent, it has not seen much use and therefore, its generality is not yet confirmed. Using copyright free music is also a positive point for this dataset, since song clips can be provided without repercussions, however, not including popular music may hurt the performance of the model when used in a more mainstream context.

4QAED

4 Quadrant Audio Emotion Dataset, or in short 4QAED, is a dataset composed of 900 music clips, each being a sample of 30 seconds from a music piece, which annotations have been mapped to the quadrants of Russel's model. The clips were gathered from the AllMusic API as well as the tags corresponding to the music's mood, which were then mapped to Warriner's list of adjectives [32], which provides A-V values, in order to map these into the above mentioned model. After a very thorough process to discard poor quality clips and clips which the annotations did not match the All Music tags, the resulting dataset is equally balanced across every quadrant, having 225 samples for each emotional quadrant considered. One possible issue of this dataset is the use of the All Music tags, since it is not explicit how the process for obtaining such tags is conducted.

WCMED-CCMED

Transfer Learning is a relatively new approach in MER, which, in simple terms, consists in training a model with a set of data, extracting the learned weights and using them in a model for a different problem hoping to achieve similar performance to the original model. Fan et al. [33] proposed a database consisting of both Western and Chinese classical music samples, providing some experimental results for Transfer Learning. The database consists of 400 samples of Western classical music and 400 samples of Chinese classical music, collected from a royalty free dataset and a music streaming platform, respectively, each sample being between 8 to 20 seconds long. The annotations were collected through a crowdsourcing effort, where a possible annotator needed to read a tutorial and provide accurate answers to a quiz before it could continue to the actual annotation process. The annotation process is done through pairwise comparison using the Russell's circumplex model, meaning that each annotator decided which sample had

[1]https://jamendo.com

higher valence and/or arousal, which is stated to simplify the annotation process as well as improving the agreement between annotators. After the samples are ranked relative to each other, these are mapped into a -1 to 1 interval for both valence and arousal. In this process, the authors assume that the distance between two successive rankings are the same, which may lead to mislabeled data. The agreement rating was shown to be decent, showing a 77.3% agreement rate for arousal and 76.1% for valence. For the conducted experiments on this database, two approaches were taken, one were the features where extracted using a modified version of the VGGish model, a pre-trained model trained using a large dataset of more than 2 million humanly annotated 10-second clips, which were then fed to a SVR model, and another using a Soundscape Emotion Recognition model based on a LSTM-RNN architecture trained using the Emo-Soundscapes [2] dataset. The obtained results, measured through squared error and mean squared error, for the first approach shows slightly better results for the Chinese portion of the database over the Western one in terms of arousal, having a significant difference in valence according to the squared error. The same is observed in the second approach. Further analyzing the samples for training, a binary classifier was built using the dataset from the second experiment to understand if the samples fall into Chinese or Western classical music. The findings show 85.97% of them fall into Chinese classical music, which may explain the results observed in both experiments.

There are some key points to a good quality database that can be inferred from the reviews conducted. Firstly, the samples themselves should be a good representation of the emotional content of the songs they are extracted from, as well as being around 30 seconds in length, as it has been proven to be a good length to preserve the emotional content of the song . The emotional model used may directly impact the quality of the annotations, results showing that dimensional models tend to perform better, although it is necessary to provide an annotation method, such as pairwise comparison, that does not rely on annotator's prior knowledge of the models, as well as analyzing the agreement between this to preserve the quality of the annotations.

[2]https://metacreation.net/emo-soundscapes/

Table 2.1: Database review.

Databases Review						
Name	Approach	Clip Duration	Features/Data	Size	Emotion Taxonomy	Notes
CAL500exp	Dynamic	3 to 16 seconds song clips	Sound Signal	500 songs divided into various clips	67 tags	Based on the 67 tags of the CAL500 dataset.
MTAT	Static	30 seconds song clips	Sound Signal, Metadata and Features (extracted from the Echo Nest API)	25,877 audio clips	Tags obtained through the TagATag game (168 tags)	Does not provide a good artist coverage and annotator's agreement is unknown.
Million Song Dataset	Static	30 seconds song clips	Sound Signal (majority of samples), Extracted Features (such as tempo and energy), Metadata(Artists, Song Title, Genre, ...)	1000000 audio clips	Tags freely submitted by users	Annotations vary in quality, combination of a variety of different databases.
Bi-Modal	Static	15 seconds clip	Sound Signal, Lyrics	360 audio clips	Discrete Russel's A-V model (4 quadrants)	It has very small size even compared to other databases reviewed, making experimentation results not very significant.
DEAM	Static/Dynamic	45 seconds audio clips	Sound Signal, Metadata (Artist, Song Title, ...)	1802 song clips	Continuous Russel's A-V model	Samples may not contain the emotional content of the song it was taken from, sample duration may not be ideal.

Table 2.2: Continuation of the database review.

Databases Review						
Name	Approach	Clip Duration	Features/Data	Size	Emotion Taxonomy	Notes
PMEmo	Static	Variable size	Sound Signal, Lyrics, Features	768 labelled samples	Continuous Russel's A-V model	Very focused database, although focusing on popular music may lead to decent results, focusing on only two years heavily limits its generability.
4QAED	Static	30 seconds song clip	Sound Signal, Metadata (Genre, Emotion Tags and A-V values)	900 audio clips	Discrete Russel's A-V model (4 quadrants)	Randomly selected intervals of song for samples.
MTG-Jamendo	Static	Full length songs, minimum 30 seconds	Sound Signal, Metadata	55,701 full songs	Tags submitted by songs' authors (692 tags)	Only encompasses copyright free songs and tags may represent more expressive than perceived emotions.
WCMED-CCMED	Static	8 to 20 seconds song samples	Features (loudness, rhythm, timbre, ...), Metadata (Song names, Sample start and end points, ...), Rankings and Ratings	800 samples (400 Western and 400 Chinese classical music pieces)	Continuous Russel's A-V model	A database composed of only one genre is very limiting when trying to find a general solution for MER.

2.3 Machine Learning and MER

The purpose of this section is to introduce Machine Learning concepts, more specifically related to Deep Learning, that will be further discussed when presenting the reviewed literature or as subject of future experimentation.

2.3.1 Fundamentals of Deep Neural Networks

A deep neural network (DNN), or dense neural network, or simply a fully connected neural network (FCNN), in contrast to a simple neural network which has a single layer of neurons between the input and output layers, are composed of a varying number of neuron layers between the above mentioned layers. An example of a FCNN is depicted in Fig. 2.8. A neuron, a mathematical analogy of a biological neuron of a brain, is a very simple structure composed of a set number of inputs and respective weights, which can be trained. A representation can be seen in 2.6. The output from a single neuron is a weighted sum or product, depending on the use case, passed through an activation function. In the literature, Rectified Linear Unit (ReLU) is the most used activation function in the input and hidden layers, whereas in the output layer, Softmax activation is used.

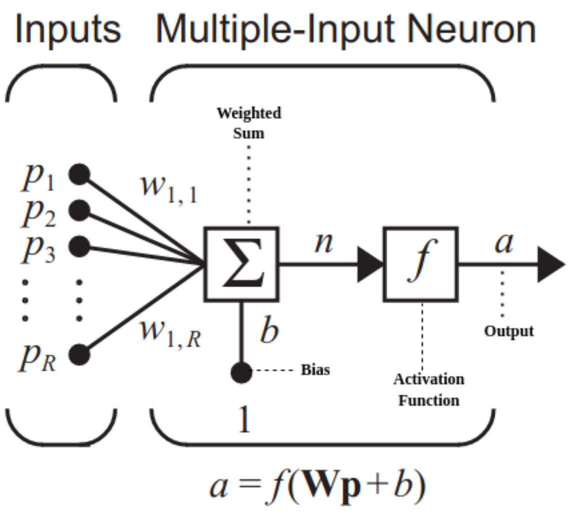
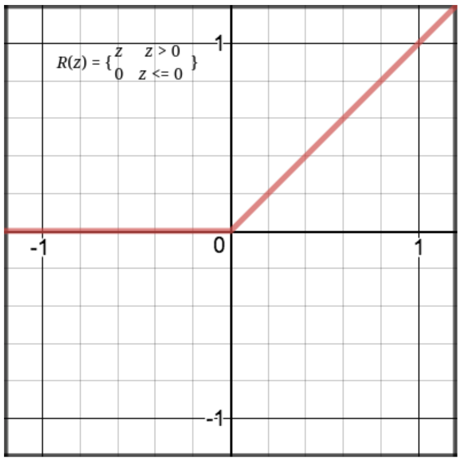


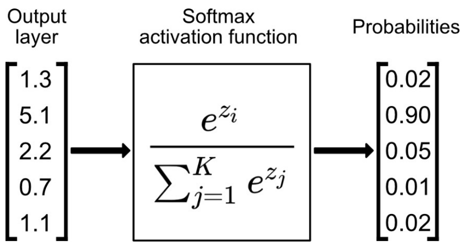
Figure 2.6: A simple multiple-input neuron adapted from [34]. The output of this neuron is calculated by applying the given activation function to the sum of the weighted sum of the inputs and the set bias.

A DNN has a variety of layers, the number corresponding to the depth of the DNN, namely, an input layer, which width, or size, is fully dependent on the problem at hand, a varying number of hidden layers or none at all, and an output layer, which width is also dependent on the problem at hand. Unfortunately, it is not easy to know the optimal depth and width of each layer, but, generally, this should scale accordingly with the size of the dataset and the number of classes, or outputs, of the problem.

All neural networks go through a training phase, where it attempts to adjust the weights to reduce the overall loss. This loss is normally calculated through a function that is the difference between the obtained output and the correct output. This is done with the backpropagation method, where the aforementioned loss function is taken into account to update the weights in reverse, beginning with the output layer, then the hidden layers and finally the input layer. The network is trained for some amount of epochs, these being the number of passes of the entire dataset for training, depending on the batch size, it can differ from one to more passes.



(a) ReLU activation function, used in the input and hidden layers.


Output layer
Softmax activation function
Probabilities
1.3
5.1
2.2
0.7
1.1
$\frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$
0.02
0.90
0.05
0.01
0.02

(b) High-level overview of Softmax activation function, used in the output layer.

Figure 2.7: Most widely used activation functions in MER literature. Left[3]. Right[4].

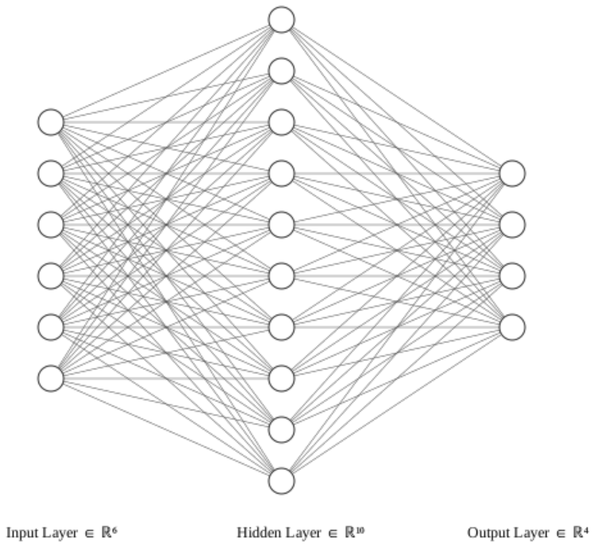

Input Layer $\in \mathbb{R}^6$
Hidden Layer $\in \mathbb{R}^{10}$
Output Layer $\in \mathbb{R}^4$

Figure 2.8: Simple FCNN architecture[5].

Should a single fully connected layer be used in any discussed methodology, it will be referred to as a dense layer to differentiate from the FCNN architecture.

One of the most prevalent problems in neural networks is overfitting. This happens due to the main goal of the neural network in the training phase, minimizing the loss function. It can normally be spotted by the neural network model achieving scores close to perfect in the training phase, but does not perform well when using the testing dataset. A dropout layer, which randomly discards a certain number of samples, can be used to mitigate this problem.

[4]https://towardsdatascience.com/softmax-activation-function-explained-a7e1bc3ad60
[4]https://ml-cheatsheet.readthedocs.io/en/latest/activation_functions.html#relu
[5]http://alexlenail.me/NN-SVG/index.html

2.3.2 Common Neural Network Models found in MER literature

Convolutional Neural Network

Convolutional Neural Networks (CNN) are often used for Computer Vision tasks such as image recognition and classification, but uses outside this field have been found, such as in MER, were they are applied to Mel-spectrograms for music tagging [35] as an example.

These networks are usually divided into convolutional layer, pooling layer and fully-connected layer, although there can be more pooling and fully-connected layers after the first convolutional layer.

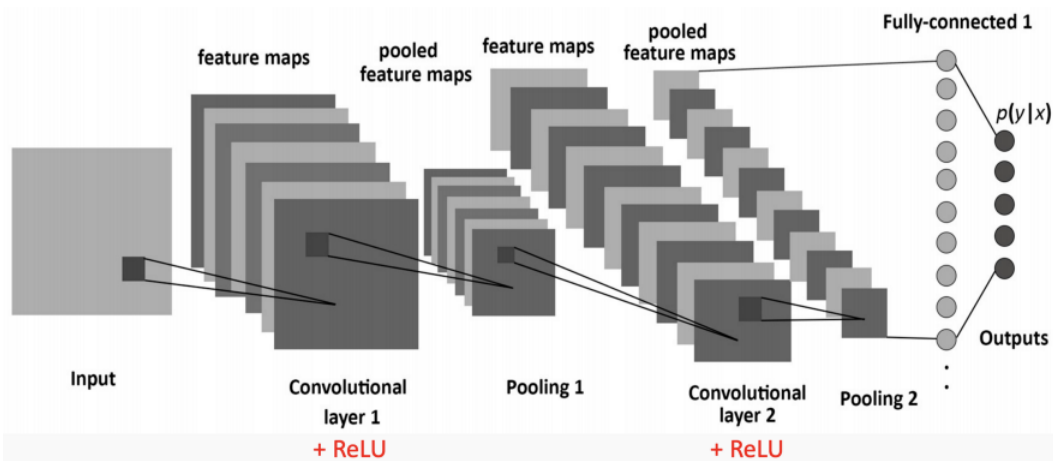


Figure 2.9: A simple CNN architecture consisting of two convolutional and pooling layers, computing the output using a fully connected layer[6].

The input feed to the convolution layer is normally an image, or more precisely, the values of each pixel in its respective channel should there be more than one, as for example, an RGB image has three channels, one for each color. A kernel, also known as filter or even feature detector, is a matrix of weights, whose size can vary depending on the problem at hand. It is applied to a particular area of the input, known as the receptive field, performing a dot product between the input values and itself. The process is repeated throughout the image, sliding a set amount of pixels horizontally and vertically, known as stride. The resulting output is called a feature map or a convolved feature, which is feed to the pooling layer.

The pooling layer objective is reducing the dimension of the feature map, achieved by sweeping the map, similar to the filter, but applying either a max operation, using the maximum value of the receptive field, or an average operation, meaning it calculates the average of all values in the receptive field, and producing an output array to be feed to the fully-connected layer.

Recurrent Neural Networks

A simple Recurrent Neural Network, unlike the discussed models, uses "memory" for more accurate predictions, feeding previously computed values to future iterations of the network. These are most suited to time-series data, which are a collection of observations made by repeated measurements over time, such as annotating the predominant emotion of a song clip in regular intervals of time.

Another characteristic that distinguishes RNN from other, is weight sharing within each

[6]https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac

layer, meaning that a given layer has the same weight for each unit, which are still adjusted through gradient descent and the backpropagation algorithm, as the already discussed architectures. The backpropagation algorithms are slightly different in these types of networks since it accounts for each iteration, or time step, of the network. Instead of adjusting the weights with the current computed loss, backpropagation through time, or BPTT, sums the errors of each time step to get the current loss before adjustments. This is done to account for the weight sharing already discussed.

These have shown to be very efficient with natural language processing tasks, such as speech recognition, and are heavily used in conjunction with CNNs in the literature for solving MEVD [36]

Despite the discussed above, these networks tend to run into some problems regarding their weights, known as exploding weights, which occur when the gradient is too large, meaning that the weights of the model have grown too much, and vanishing weights, occurring when the gradient has become too small, leading to the weights reaching near 0. Either way, the model becomes unable to learn, massively impacting the overall performance.

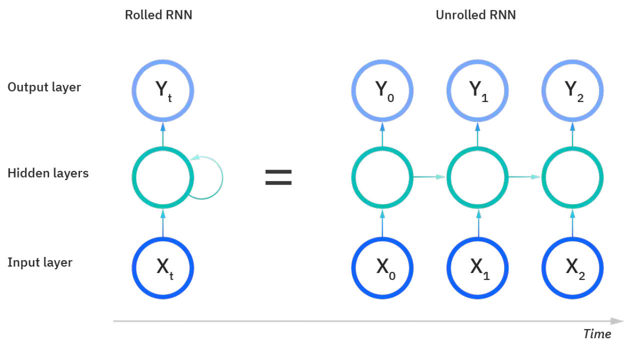


Figure 2.10: A simple RNN architecture [7].

Long short-term memory networks, (LSTM), were proposed to combat the former problem. This architecture adds three gates to each unit[8]: an input gate, an output gate and a forget gate. This allows the unit to decide whether to keep a certain computed value in "memory" if it is found relevant. Some variations include Bidirectional LSTMs, which use not only past information, but also future information to update the current input and output.

These have been widely adopted due to performing better than regular RNNs, having been used in recent literature also to solve MEVD [37].

2.3.3 Training with Few Samples

Few-Shot Learning

Few-Shot learning can be defined as a Meta-Learning problem using a very limited amount of samples for each class. The purpose of Meta-Learning algorithms is to learn how to learn to classify the given input data, in contrast to the usual Machine Learning approaches, where the model learns how to classify the data using the training data and evaluate the results using test data. This is accomplished by defining a set of tasks to be used for training the algorithm, using that experience when trying to solve the target task. The

[7]https://www.ibm.com/cloud/learn/recurrent-neural-networks
[8]An LSTM unit can also be referred to as a LSTM neuron

model will receive a set of tasks, also referred to as episodes, each episode comprised of N classes, each one with k images, as well as a query set of images Q. For each episode, the model will learn how similar or different each set of images are from each other, evaluating the performance of the learned parameters on the Q set, maximizing the accuracy on this last one as much as possible. Finally, after the model is trained, a test task is presented in order to evaluate the performance of the trained model. The discussed process is depicted in figure 2.11.

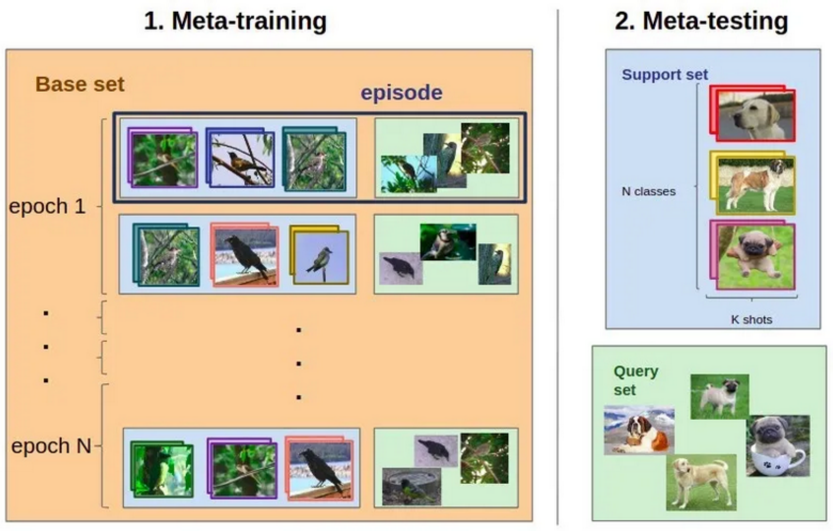


Figure 2.11: Process of training and evaluating a Few-Shot learning algorithm.

Due to being a very recent technique, it as not seen much research in the MER field, mostly using other physiological signals such as EEG signals that are not in the scope of this work [38]. The achieved results lead us to believe that this should be further explored. In the literature, one of the most common architectures based on Few-Shot learning found are Siamese Networks, or SNs, which are made up of two or more identical subnetworks, mirroring the weight and parameter adjustments made in a subnetwork to the others. The ultimate goal of this architecture is to learn a similarity function based on the distance of the given samples in a high dimensional space, known as an embedding, to determine how similar or dissimilar two samples are from each other.

2.4 Data Augmentation

In this section, classical audio augmentation are presented, followed by a GAN which generates new samples from a learned latent space.

2.4.1 Classical Audio Augmentation

Classical audio augmentation apply a certain transformation directly to the raw signal of the sample. Below some of the most commonly used audio augmentations, which are also used in previous developed Data Augmentation methodologies, are presented:

- Time shifting: randomly shifts the sample in time, making some of the beginning or ending of the sample silent, as seen in Fig. 2.12;

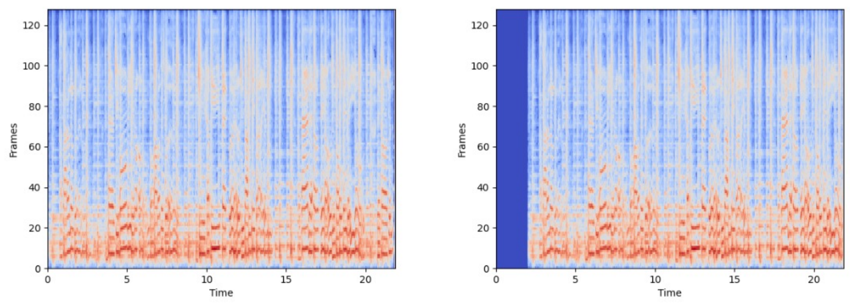


Figure 2.12: Music clip before and after applying Time Shifting.

- Pitch shifting: randomly lowers or raises the pitch for the whole sample in some amount of tone, be they complete or half tones, as seen in Fig. 2.13;

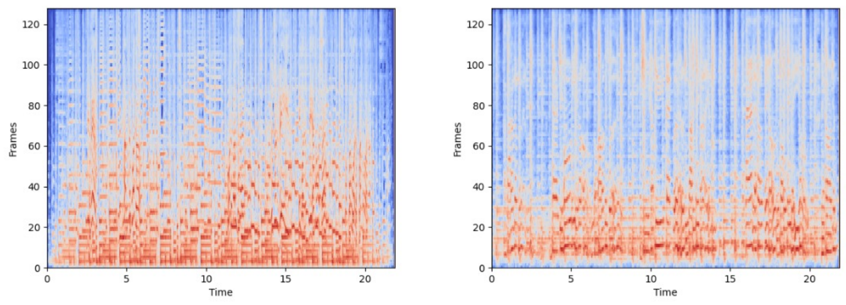


Figure 2.13: Music clip before and after applying Pitch Shifting.

- Time stretching: randomly increases or decreases the tempo of the song by some factor, as seen in Fig. 2.14;

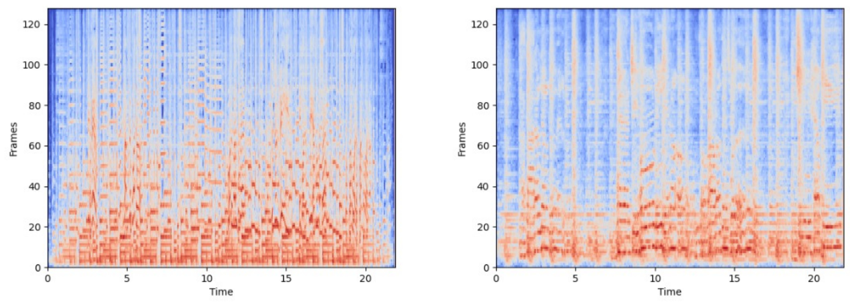


Figure 2.14: Music clip before and after applying Time Stretching.

- Power shifting: randomly increases or decreases the intensity of the whole sample by some amount of dB, as seen in Fig. 2.15.

There also other audio augmentation techniques experimented with in this work. These were:

- Time-Frequency Masking: Presented by Park et al. [39], Time-Frequency Masking, Fig 2.16 is a technique that, as its name suggests, masks some part of the sample,

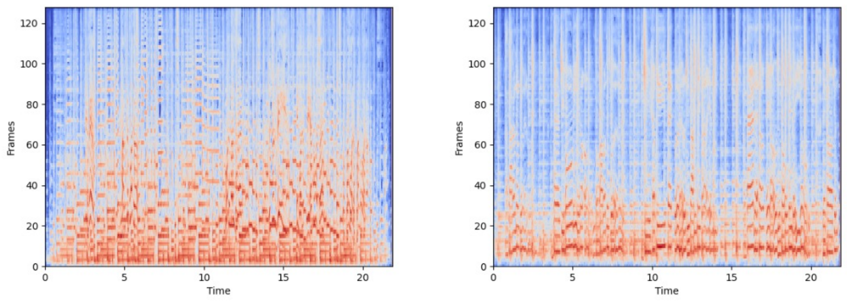


Figure 2.15: Music clip before and after applying Power Shifting.

in the time and frequency axis, making a random segment of time and a random frequency band silent, in order to generate new samples. This technique is widely used in Speech Emotion Recognition due to making the models more robust, having been applied to the MER task with the same intent [40].

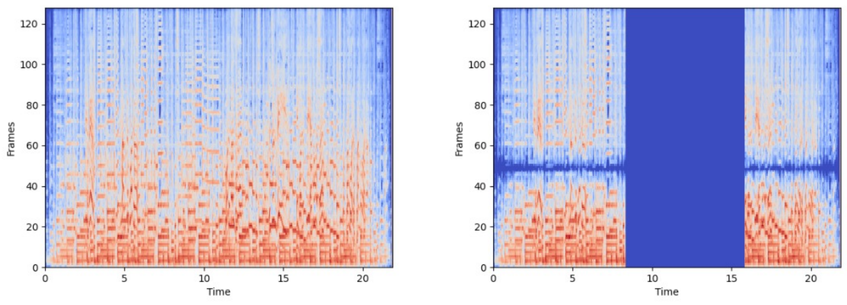


Figure 2.16: Music clip before and after applying Time-Frequency Mask.

- Tanh Distortion As suggested in the audiomentation repository, this technique, seen in Fig 2.17, was experimented with. The whole sample is distorted following the tanh function, making mostly instruments appear, to be distorted, such as the guitar.

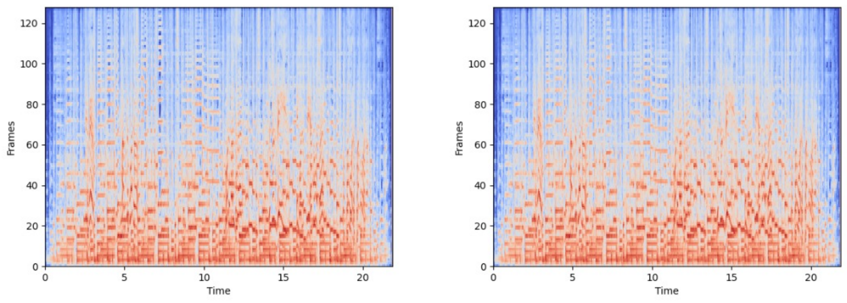


Figure 2.17: Music clip before and after applying Tanh Distortion.

- Seven-band Parametric Equalization Also suggested in the audiomentation to make the model more robust, this technique, seen in Fig. 2.18 adjusts the amplitude of certain frequencies of the samples. From seven predefined frequency intervals, a set value is picked for each and the transformation is applied. The technique is actually a composition of a low and high shelf filters (increases or decreases amplitude randomly), and five picking filters for the mid frequencies (increases or decreases amplitude randomly in the neighborhood of the center frequency picked).

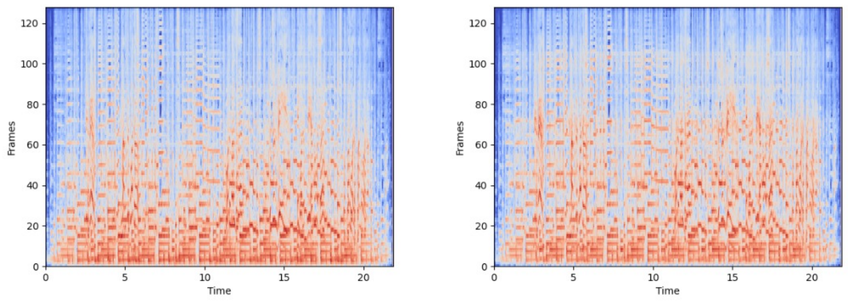


Figure 2.18: Music clip before and after applying Seven-banded Parametric Equalization.

- Random Gain This classical audio augmentation, seen in Fig. 2.19 randomly increases or decreases the amplitude of the whole sample, meaning that the sample gets louder or quieter respectively. No change is perceptible possible due to .

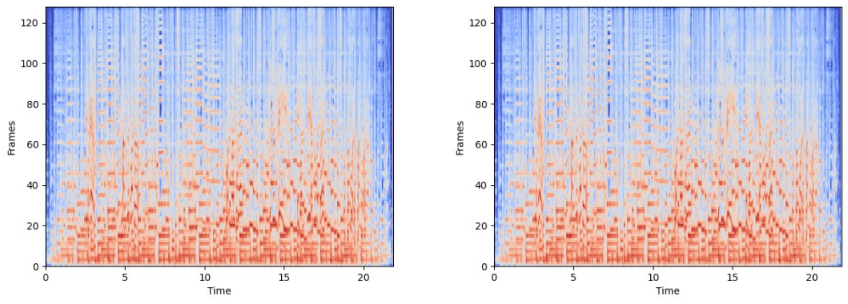


Figure 2.19: Music clip before and after applying Random Gain.

- Background Noise Another classical audio augmentation used for making a model more robust. This one randomly adds background noise from a preselected set of audio clips. The audiomentations library points to possibilities for this preselected set, from which the ESC-50 [41] was chosen. This dataset provides 2000 audio clips captured in the real world and are separated into five different types of environmental sounds: animals, soundscapes, speech, interior and exterior sounds. For each augmentation, a random clip from the abovementioned 2000 is chosen to be merged with the original clip, generating a new sample.

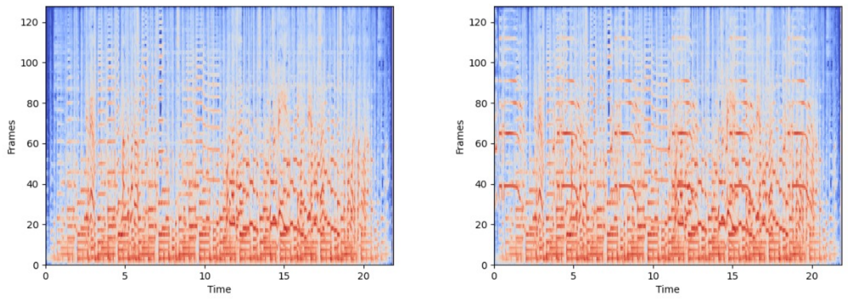


Figure 2.20: Music clip before and after applying Background Noise.

There are still a lot more audio augmentations that could be experimented with, and should be experimented with in future work for possible further improvements.

2.4.2 Generative Adversarial Network

One of the most prevalent problems in MER is the size of the available public datasets, which are normally reduced due to the labor and time intensive effort that is required to produce quality annotations for each sample. Recently, Generative Adversarial Networks, or GANs, introduced in [42] by Goodfellow et al., were proposed as an approach to generate new and unique samples using the already annotated samples [7].

The architecture of a GAN is composed by two Neural Networks: a generator, which, as its name suggests, generates samples using a sampled vector from a latent space distribution[9], and a discriminator, which has the objective of differentiating the real samples from the generated samples coming from the generator. The adversarial process is conducted by training the generator to maximize the fidelity of the generated samples in order to make them as indistinguishable as possible from the real ones in regard to the classification of the discriminator, while the discriminator is trained as already discussed models, adjusting the weight parameters in order to better distinguish the samples fed to it.

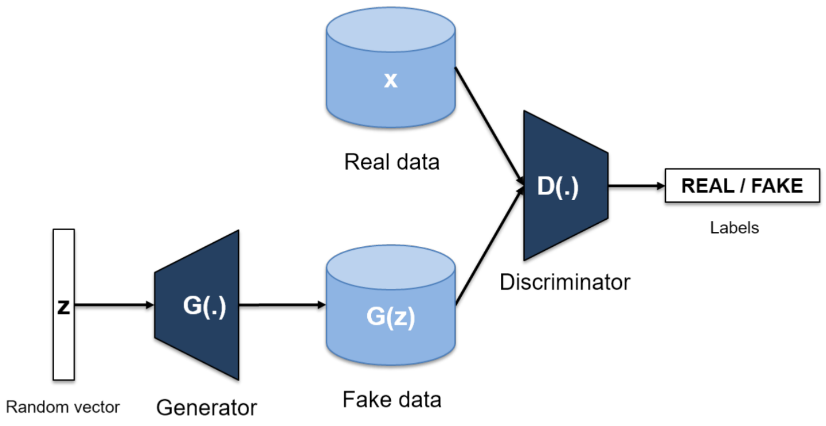


Figure 2.21: Overview of GAN architecture adapted from [43].

Although promising, these networks bring more complexity to the training phase, since it is no longer as simple as minimizing loss as discussed previously. As stated in [43], the training process is more akin to finding a solution to a minimax algorithm: minimize the discriminator error and maximize the quality of the generator samples. Problems that arise include the above mentioned vanishing gradients, mostly due to one of the networks performing significantly better than the other, and mode collapse, which, in simple terms, means that the samples outputted by the generator have stagnated, implying that the generator as clearly outperformed the discriminator.

A different approach to GANs was proposed by Radford et al. [44]. Known as Deep Convolutional Generative Adversarial Networks, or DCGANs, these have shown to be more stable on the training phase in comparison with simple GANs. All layers in both networks are convolutional and are deprived from pooling and fully-connected layers, resorting to

[9]A latent space is defined by the dimensionality of the compressed input data feed to a DL model, where the data can be represented as a unique point, meaning that similar data will be relatively close to each other in this space. A latent space distribution takes advantage of this property, generating a distribution of similar data relative to this space.

fractional-strided convolutions to fit the output of a layer to the input size of the next convolutional layer, in contrast with the discussed architecture.

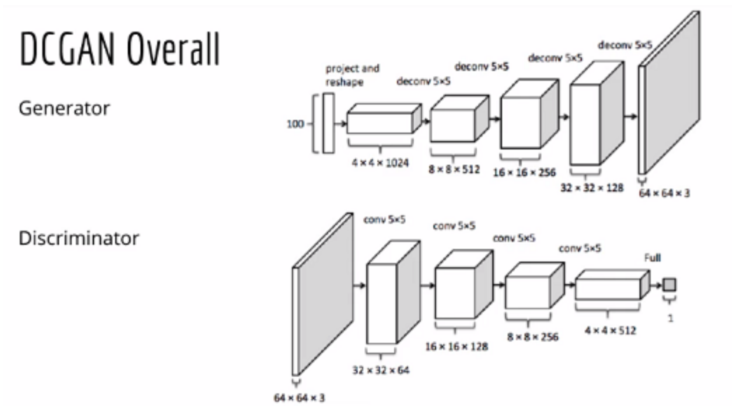


Figure 2.22: DCGAN architecture representation.[10]

2.5 Evaluation Metrics

There are a variety of metrics that are utilized to assess the choice of hyperparameters for training a model, the model's performance after being trained and whether the results prove the formulated hypotheses. In this section, these points are briefly discussed, focusing on the metrics found more frequently in the literature and used for the experimentation portion of

The choice of hyperparameters for a DL model, which include number of epochs, batch size, optimizer and respective learning rate, is normally done by evaluating the evolution of the Loss, which represents how bad the prediction is for a certain sample against the real target, meaning that lower Loss is better, and Accuracy, which represents how accurate the predictions are on the overall data, meaning that higher Accuracy is better. For regression purposes, Accuracy is replaced by more relevant metrics, such as the Root Mean Squared error, which represents how distinct are the regressor's predictions against the actual targets of the samples in question, lower being better.

As for a DL model's performance, this is mainly assessed through the F1 Macro[11] Score, from here on referred to as F1 Score achieved on the test data, this being the harmonic mean of the Precision and Recall on the model's predictions on the test data. Formulas for calculating Precision, Recall and F1 Score can be seen bellow:

$$Precision = TP/(TP + FP)$$
$$Recall = TP/(TP + FN)$$
$$F1Score = 2 * ((Precision * Recall)/(Precision + Recall))$$

Precision is used to tell how many of the predictive positive examples are true positives, while Recall is used to tell how many of the positive examples were correctly predicted as positive. Taking as an example a binary problem to differentiate between dogs and cats, as

[10]https://debuggercafe.com/implementing-deep-convolutional-gan-with-pytorch/

[11]Contrary to F1 Macro, F1 Micro takes into account the class distribution, using a weighted vector when making the computation, ensuring that the score does not erroneously show good performance due to the over represented classes.

Table 2.3: Generic Confusion Matrix with F1 Score example.

	C1	**C2**	**C3**	**...**	**Cn**	**F1 Score Per Class**
C1	x			...		🟩
C2		x		...		🟥
C3			x	...		🟦
...	...	...	...	...	...	**...**
Cn				...	x	🟨
					F1 Score Total	

the positive and negative classes respectively. The Precision for the dog class would be the ratio between the correctly predicted dogs and all examples predicted as dogs, including the ones which are in reality cats, while the Recall would be the ratio between the correctly predicted dogs and all true dog samples.

Performance on each individual class can be further analyzed using the Confusion Matrix of the predicted and actual targets, as seen in Table 2.3, along with the F1 Score for each class and the overall F1 Score.

Having trained and evaluated the model, the formulated hypotheses are validated or refuted using inferential analysis on the F1 Scores calculated for every fold on the chosen baseline and the methodology which performance is expected to be better. This is done through a significance test with a certain confidence interval, most of the time 5%. Should the p-value, the result of the significance test, be below the confidence interval, the results are deemed statistically significant and the hypothesis is validated.

2.6 Static MER

This section present a review of proposed approaches for solving Static Music Emotion Recognition, presenting the progress and current state of the art approaches for this problem.

2.6.1 Classical Machine Learning

For recognizing patterns, computers need to learn models using previously labeled data from experts to a number of possible outcomes dependent on the problem in order to predict unlabeled data. For recognizing static emotion, the same applies. Firstly, various songs are compiled and a sample is extracted for each song, representing the whole emotional content of the song. These samples, after being thoroughly analyzed, are annotated by human subjects, with support on an emotional model. The audio features are then extracted from the samples, carefully selected, and are used to discover patterns between samples with the same annotations. These features are used to train the ML model, which after finishing training, receives data for testing the outcome of the predictions. This means that the models are mostly dependent on the features extracted and selected, directly influencing its performance.

A high-level view on solving MER can be seen in Fig. 2.23.

Having the generation of playlists based on the listener's mood as the focus, Meyers [20] proposes an approach combining audio and lyrics of a musical piece to classify the under-

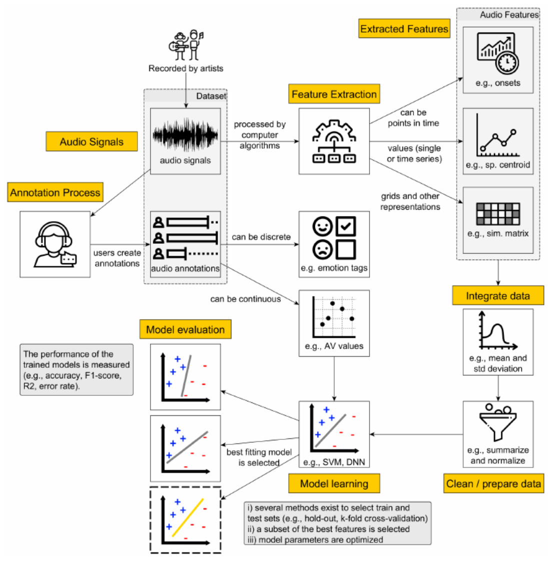

Recorded by artists
Extracted Features
Audio Features
e.g., onsets
Dataset
Feature Extraction
Audio Signals
audio signals
processed by computer algorithms
can be points in time
values (single or time series)
e.g., sp. centroid
Annotation Process
grids and other representations
e.g., sim. matrix
users create annotations
audio annotations
can be discrete
e.g. emotion tags
can be continuous
Integrate data
e.g., AV values
Model evaluation
e.g., mean and std deviation
The performance of the trained models is measured (e.g., accuracy, F1-score, R2, error rate).
best fitting model is selected
e.g., SVM, DNN
e.g., summarize and normalize
Model learning
Clean / prepare data
i) several methods exist to select train and test sets (e.g., hold-out, k-fold cross-validation)
ii) a subset of the best features is selected
iii) model parameters are optimized

Figure 2.23: A high-level view on solving MER as seen in [7].

lying emotion of the piece in question. For this purpose, it was also proposed a discrete dimensional model, obtained by mapping Hevner's Adjective List into Russel's Circumplex Model, already discussed in this section. The approach consists in classifying the song into a class using decision trees and then using KNN, trained with 360 songs, to assign it to one of the eight sectors on the used model. Various features were extracted, such as mode, harmony, tempo, rhythm and loudness, in addition to the lyrics information also extracted, led to good results when compared against All Music Guide experts tags, according to the author, with no data for comparison available.

Panda et al. [45] proposes an approach combining standard and melodic audio features in an attempt to outperform the recent models at that time, which had plateaued [46]. The compiled dataset for evaluation was obtained by mapping tags from AllMusic[12] into five clusters used by the MIREX Mood Classification Task, resulting in 903 music clips with a length of 30 seconds each, reasonably balanced across all clusters. After feature selection, 9 melodic and 2 standard audio features were selected from the best ranking ones, reducing from 351 to 11 features. As for the classifiers, Naïve Bayes, KNN and SVM models were evaluated, the best value being 0.64 F1-Score obtained by the SVM model.

Later, the focus shifted to developing novel features. In [8], the lack of relevant features for musical characteristics proved as relevant for emotion identification and 29 newly developed features were proposed to improve emotion classification. For testing the significance of these features, a new dataset was created, the 4QAED dataset, already discussed in Section 2.2, and training the model proposed in [45] with various features, achieving the best results, a 76.4% F1-Score, using a combination of the novel features with 71 standard, which represents a 9% increase compared to using 70 baseline features.

A different approach from the reviewed so far was proposed by Markov et al. [47] in which a Gaussian Process, or GP, was used. This consists in setting a probability distribution over possible regression functions that comprehend a set of values, which in turn gives a certain level of confidence to a solution for the problem at hand [48]. For the music emotion classification evaluation step of this approach, the MediaEval 2014 database was used, consisting of 1744 clips, each 45 seconds, of distinct musical pieces, balanced across

[12]All Music API link: http://developer.rovicorp.com/io-docs

8 genres, where 500 were picked for training and 500 for testing. The results for static emotion classification shows the same or better performance over SVM regression-based estimation. As noted, the feature set used may be limiting the overall performance of the models evaluated, since the features may be too simple or in reduced numbers for effective evaluation.

Bargaje [49] proposes an approach using an SVM as a classifier, but different from the mentioned approaches, a feature selection process is conducted before training the model using a genetic algorithm, or GA for short, a paradigm of evolutionary algorithms, or EA. The motivation behind this approach is reducing the time needed for computation by selecting the more relevant features. The emotion taxonomy used is the discrete two-dimensional model based on Russell's A-V, but adding a third dimension, loudness, increasing the recognizable emotions from 4 to 8. As stated by the author, due to the unavailability of a public dataset, a dataset was built, composed of samples of 200+ randomly selected English and Hindi songs with a length around 30 seconds. Besides mentioning using jAudio[13] to normalize the samples and the usage of a software to ease the annotation task, there is no mention of the distribution of samples across quadrants, musical background of the annotators or agreement between their annotations. The features were then extracted using jMIR[14], which exact number is not stated, this being above 600.

As mentioned above, before training the model, a feature selection step is performed using a GA. In simple terms, the algorithm is based on the Darwinian theory of evolution. An initial population is set, composed of a given number of individuals, which are subsets of the feature set retrieved with the mentioned tools. The number of iterations is known as the number of generations, which value depends on the task at hand. In each iteration, two operations have a set probability of occurring: crossover, exchange of information between individuals; and mutation, a change that occurs at the individual level. Finally, a fitness function needs to be set in order to define how fit the solution provided by an individual is. Other parameters not mentioned in this approach, which may be influenced by the fitness of the individual, are the selection parameters used for parent selection, which indicates which individuals may perform crossover, and survivor selection, which indicates how many individuals transition to the next generation.

For this approach, various numbers of generations were evaluated, 20/50/100, as was the case for the probability of crossovering, more specifically a one-point crossover, 60% or 80%, leaving the probability of mutation, more specifically a flip mutation, at 3.3%. These values are set this way to focus on global search, or exploration, over local search, or exploitation, which is a very good approach considering the mentioned over 2^{600} possible combinations of features. The fitness function is set as the accuracy of an individual to correctly identify the samples of the training dataset, which is done by training the SVM with the feature subset of the individual.

The results of this approach shows that the best feature set was found using a 60% probability crossover over 50 generations, resulting in a 84.88% accuracy with a 656 feature set. Using the optimal feature set found for training the model, the proposed method performed significantly better than the others, reaching 84.57% accuracy. Despite these results, there are some concerns, mainly with the dataset used, as mentioned before, but also the fact that the comparisons are made with approaches that used different datasets. Not only this, the focus on the training dataset may limit the performance of the model, since the features are selected in function of the training data and may not apply well to other samples not present in the dataset.

[13]http://jaudio.sourceforge.net/
[14]http://jmir.sourceforge.net/overview.html

The implementation of Classic Machine Learning approaches for solving Static MER requires a robust knowledge of all necessary audio features and previous computation of these before being fed into a model. Not only this, the relevancy of features need to be further calculated for preventing lower performance due to the use of unnecessary features. Feature selection and implementing a EA for the same effect should be explored as solutions.

2.6.2 Deep Learning

Due to the limitations made apparent with successive proposals of approaches using classic machine learning algorithms, researchers shifted their focus to deep learning methods. The necessity to previously calculate relevant features to increase the performance of the networks may have been a contributing factor to this shift, as well as exploring approaches the better replicate the human process of emotion perception in music.

One of the earliest known Deep Learning approaches was proposed by Feng et al.[50], using a simple 3 layer feedforward neural network, composed of an input, hidden and output layer. The only features used are related to tempo and articulation, based on Juslin's theory[51], which are mapped to one of four emotional categories: happiness, sadness, anger and fear. Despite presenting good results for an early approach, these are misleading, since the dataset compiled for evaluation was very unbalanced.

Choi et al. [52] contributes to the field by developing an approach using Fully Convolutional Networks, or FCNs, which differs from a simple CNN due to not implementing any fully-connected layers, implementing only convolutional and pooling layers. The task itself is treated as a multi-label classification problem, meaning that a clip can be classified as part of multiple classes, somewhat addressing the subjectivity problem of the field, but very demanding computationally, which leads to a need for better optimization in regards to the model used. Various architectures were proposed with increasing number of layers, having as baseline the FCN-4 architecture depicted in Fig. 2.24, which consists in an input layer, 4 convolutional layers, each one preceded by a max-pooling layer. Each convolutional layer has a Rectified Linear Unit, or ReLU, as an activation function, and a two dimensional 3x3 convolutional kernel, except the output layer, where a Sigmoid activation function is mapped to a 50 dimensional tag vector and a simple 1x1 kernel is used.

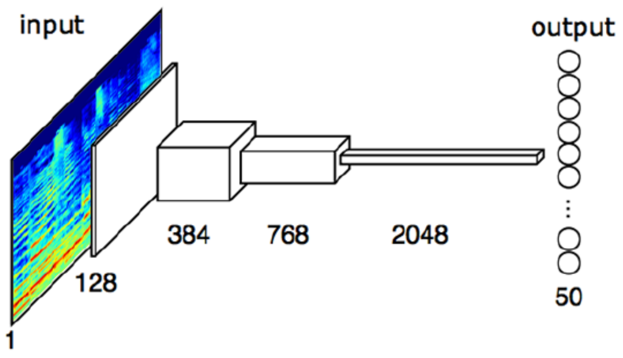
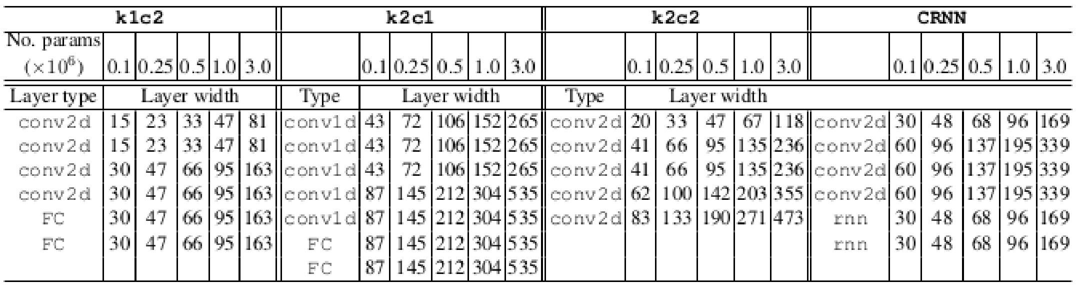


Figure 2.24: Diagram of the proposed architecture FCN-4 architecture, comprised of 4 layers, each number indicating the number of features maps implemented in each layer, outputting the prediction on a 50 dimensional tag vector.

The evaluation was done on the MagnaTagATune dataset, were it achieved an AUC score,



k1c2						k2c1						k2c2						CRNN					
No. params ($\times 10^6$)	0.1	0.25	0.5	1.0	3.0		0.1	0.25	0.5	1.0	3.0		0.1	0.25	0.5	1.0	3.0		0.1	0.25	0.5	1.0	3.0
Layer type	Layer width					Type	Layer width					Type	Layer width										
`conv2d`	15	23	33	47	81	`conv1d`	43	72	106	152	265	`conv2d`	20	33	47	67	118	`conv2d`	30	48	68	96	169
`conv2d`	15	23	33	47	81	`conv1d`	43	72	106	152	265	`conv2d`	41	66	95	135	236	`conv2d`	60	96	137	195	339
`conv2d`	30	47	66	95	163	`conv1d`	43	72	106	152	265	`conv2d`	41	66	95	135	236	`conv2d`	60	96	137	195	339
`conv2d`	30	47	66	95	163	`conv1d`	87	145	212	304	535	`conv2d`	62	100	142	203	355	`conv2d`	60	96	137	195	339
`FC`	30	47	66	95	163	`conv1d`	87	145	212	304	535	`conv2d`	83	133	190	271	473	`rnn`	30	48	68	96	169
`FC`	30	47	66	95	163	`FC`	87	145	212	304	535							`rnn`	30	48	68	96	169
						`FC`	87	145	212	304	535												

Figure 2.25: A table describing the implemented layers and layers width applied to each parameter size experimented with, as presented in [53].

a scoring measure better fitted for the data provided on this dataset, higher then most of the approaches evaluated with it, a 0.894 AUC score. Input features evaluated include STFT, MFCC and Mel-spectrogram representations, the last one having the best performance compared with the others using the same architecture. Due to the above mentioned approaches all sharing a very small interval of AUC scores, the proposed architecture was also evaluated using the Million Song Dataset [26], were the FCN-6 architecture using Mel-spectrogram representations of inputs reached 0.851 AUC score. Despite achieving very good results, the experiment on the Million Song Dataset is not compared with other approaches and the dataset itself has questionable quality, as discussed previously.

In the same year, Choi et al. [53] proposes an architecture combining CNN and RNN layers, a Convolutional Recurrent Neural Network, or CRNN, for extracting the relevant features from the Mel-spectrogram given as input and extracting the temporal features from the processed input, respectively. Mel-spectrograms have been more and more adopted in the MER literature due to the achieved results, these being a representation of the sound spectrum as captured by the human ear. The model is compared to the model discussed in [52], as well as other baselines, such as an architecture based of 4 one-dimensional convolutional layers, and another that uses 5 convolutional layers, the first being two-dimensional in order to compress the input data and reducing computation complexity, outputting to 2 fully-connected layers. The CRNN model itself consists of 4 CNN layers, with 2 RNN layers using GRU units on top of the first layers. A summary of the discussed networks can be found in Fig. 2.25.

As in [52], the output of all networks are given by a 50 dimensional tag feature taking as input the output of the layer. The MSD dataset with the last.fm tags as targets was used, which is a negative point in evaluating this approach due to the quality of the dataset. The models were also scaled between 0.1 million and 3 million parameters in order to assess the gain in performance of introducing more adjustable variables. Although an increase of parameters should lead to better results, it is also important to note that a fewer number of feature maps used leads to removing redundancy in the input data. The proposed model outperforms all the baseline models assessed, achieving a 0.86% AUC score when adopting the 3 million parameter approach.

As previously stated in Section 2.2, Transfer Learning is a recent approach to MER, where the weights of good performing models are transferred to models aimed at solving MER in an attempt to achieve similar results. The purpose of pursuing such techniques is due to various studies showing that listeners with different mother tongues rate the same song samples with different emotional states, as pointed out in [7]. Cañón et al. [54] explores this approach in order to build a model for emotion recognition, transferring features learned from models trained using English and Mandarin speech to models for MER. The same authors explored this issue in [55], proposing that there is a cultural layer to the emotions felt by humans.

The approach itself uses the discrete Russell's A-V emotional model and the 4QAED database is used proposed by Panda et al. [6]. The feature extraction process for the model trained using speech was a sparse convolutional autoencoder, or SCAE, composed of two 2-D convolutional layers with a 3x3 kernel, which output is feed to a max-pooling layer as well as a dropout layer, to mitigate overfitting as mentioned previously in Section 2.3. The decoder portion uses the same structure as the encoder, but changing the max-pooling layers to up-sampling layers. For extracting features, the decoder portion is replaced with a DNN layer, composed of 3 fully connected layers, totaling 512 neurons. Two variations of the models were experimented with: *SCAE Feat. Ext.*, which freezes the weights transferred from the speech model, tuning only the DNN portion of the network, and *SCAE Full*, where all the network is trained, but with a lower learning rate. The performance of these models for feature extraction is not very impressive, reaching a 48% F1-Score with the experimented configurations. These also do not perform very well when predicting the quadrant of a given sample, although it is stated by the authors that this is common in MER literature, since it is hard to distinguish between quadrants when considering the valence axis (i.e. differentiating between the third and fourth quadrant), and is validated with a better performance when predicting arousal and valence independently. It is also important to note that the models where not validated through 10-fold validation, as it is a common approach in other reviewed papers, and the database itself is not balanced, there being a very large difference between Mandarin and English samples, the first having considerably more.

A recent paper authored by Grekow [56] that explores an approach where a pre-trained model is used to pre-process the data before being fed to a RNN. To achieve this, the development was divided into two parts.

Firstly, the GTZAN data collection from [57] was used, where 324 six-second fragments of the samples in the dataset were all annotated by all five musical experts that participated in the study, setting an arousal and valence value between -10 to 10 to map each sample to Russel's model [19].The correlation between arousal and valence on the mapped segments were considered uncorrelated, obtaining a Pearson correlation coefficient of -0.03, and that the agreement levels were high, above 90% using Cronbach's α.

The features were extracted using audio analysis and audio-based music information retrieval tools, more specifically Marsyas[15], which resulted in a 124 feature vector, and Essentia[16], this resulting in a 529 feature vector. Since a RNN architecture is used, as seen in Fig 2.26, the samples were further segmented in order to produce data sequences, which the RNN will find relationships between the segments of a given sequence.

The full architecture, as seen in the above figure, is composed by an input sequence, a feature vector, fed to the input layer, a LSTM layer, where a tanh activation function was used, a dense layer and an output layer. The configuration and training was done using the WekaDeeplearning4j package from Lang et al. [58]. There is no mention of a dropout layer to help prevent overfitting, being instead handled by an early stopping strategy. Further architectures were developed, which vary in number of LSTM layers and layer width. To evaluate this first architecture, Linear Regression and the SMOreg algorithm, developed by the same author, were used as baselines. The evaluation was conducted for each tool used. The feature set retrieved with Marsyas achieved the best performance using the RNN4

[15]https://github.com/marsyas/marsyas
[16]https://essentia.upf.edu/streaming_extractor_music.html

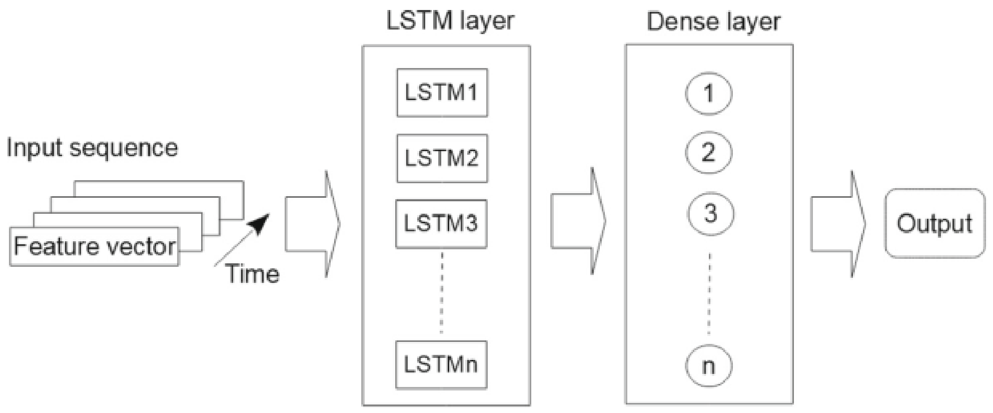


Figure 2.26: Depiction of the proposed RNN architecture [56].

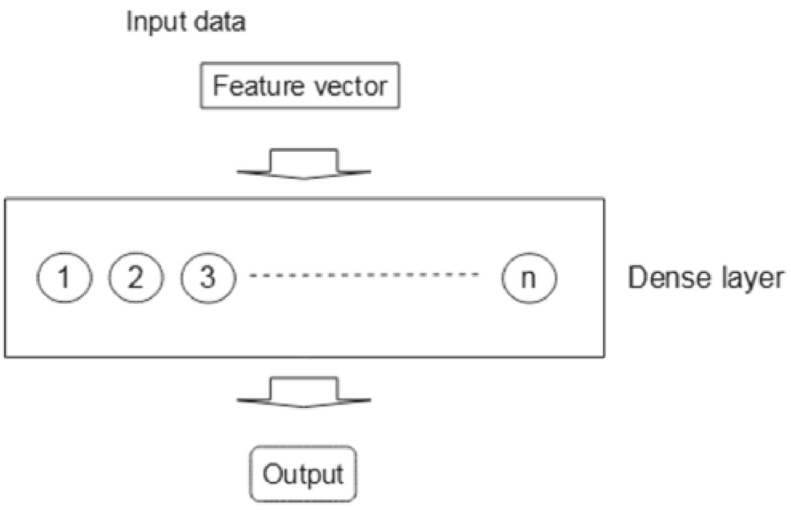


Figure 2.27: Depiction of the pre-trained model for feature extraction [56].

architecture, consisting of 2 LSTM layers instead of one, each with 248 units, achieving squared error and MAE of 0.67 and 0.12 for arousal and 0.17 and 0.12 for valence. As for the Essentia feature set, the best performance using all features was also achieved by the RNN4 architecture, despite adding more complex architectures to better handle the feature set size, having a squared error and MAE of 0.69 and 0.11 for arousal and a 0.40 and 0.13 for valence, a considerable improvement on valence compared to the Marsyas feature set. This may be achieved to the larger quantity of features of Essentia, meaning that it may encompass more features relevant for valence, which is very difficult to accurately calculate as seen in the reviewed literature.

For the last part, focusing on the Essentia features, another architecture was explored. To further select the most relevant features from the Essentia feature set, a simple neural network was implemented consisting of a single dense layer, being trained with the feature set. The layout of the architecture was essentially the same as the discussed before, but the input sequence is first fed into the pre-trained model, which is depicted in Fig. 2.27, which outputs a new set of features of the input sequence, which is then feed as the input of the first LSTM layer, following the same process as the first approach. The evaluation for this approach was conducted using all the previous models evaluated in the first approach, achieving the best results for the RNN 4 architecture, with a squared error and MAE of

0.73 and 0.11 for arousal and 0.46 and 0.12 for valence, which means a 6% improvement in squared error for the arousal regressor and 15% for the valence regressor compared to the best results of the first approach. Although the approach achieved better results on using the pre-trained network in addition to the RNN architecture, there is not a comparison between using a feature selection algorithm instead of the pre-trained network to best pick the features in order to justify this more computationally expensive approach.

Also using EAs, Yang [59] proposes a neural network with a modified BP algorithm, using the artificial bee colony algorithm, ABC, to more easily explore the search space of the possible weight values and reduce the time needed for convergence. To evaluate this approach, the DEAM database was chosen, in this paper referred as MediaEval Emotion in Music, MEM, meaning that the emotional taxonomy adopted is Russel's A-V, which was already discussed in a previous section. The structure of the model itself is not described in much detail, being always referenced as a BP neural network, having a lot more focus on how the ABC algorithm is used to optimize the network weights. A diagram explaining overall functioning of the algorithm is presented in Fig. 2.28.

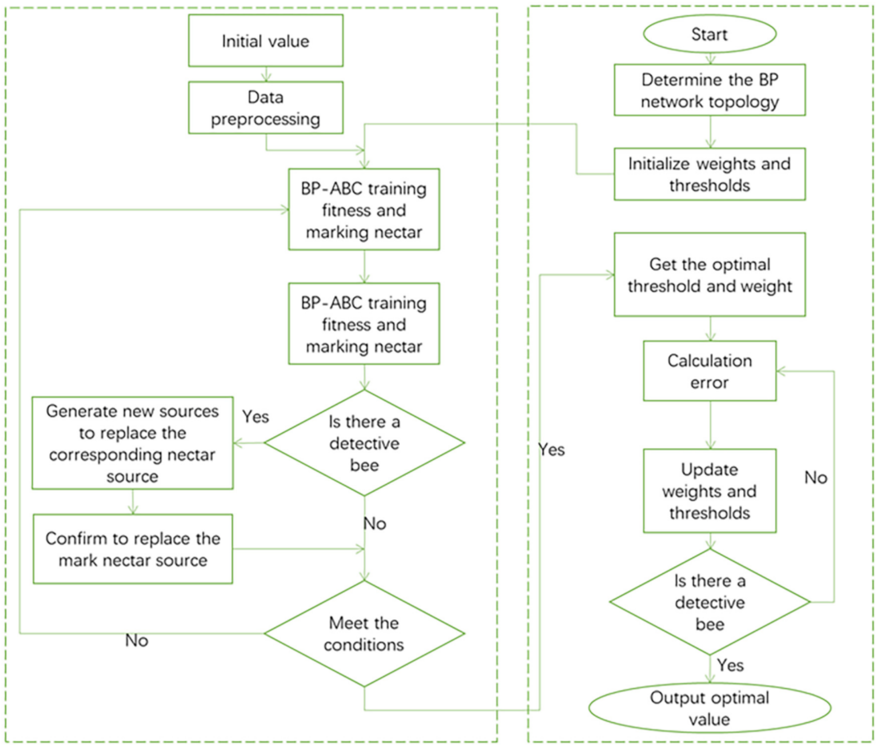

Initial value
Data preprocessing
BP-ABC training fitness and marking nectar
BP-ABC training fitness and marking nectar
Is there a detective bee
Yes
Generate new sources to replace the corresponding nectar source
Confirm to replace the mark nectar source
No
Meet the conditions
No
Yes
Start
Determine the BP network topology
Initialize weights and thresholds
Get the optimal threshold and weight
Calculation error
Update weights and thresholds
Is there a detective bee
No
Yes
Output optimal value

Figure 2.28: A high-level representation of the proposed modified BP algorithm by Yang, as presented in [59].

The algorithm can be more easily explained by understanding how the overall model learns the optimal weights. Firstly, the NN parameters, such as the number of hidden layers and input, output and each hidden layer size, and ABC parameters, value of the nectar source, size of bee colony, maximum number of iterations and the parameters discussed for GA, need to be set. Each nectar source, a set of values on the search space, are then assigned to one employed bee, which will search the entire search space for possible new nectar sources. The fitness of the nectar sources are evaluated in order to assess which are the best sources, which neighborhood will be searched for better sources. The found optimal solution is then compared with the already found optimal solution in memory, replacing it if the fitness value is higher. Should the preset limit for a nectar search be reached and the optimal solution does not change, the search will stop and the related employed bee will become a scout bee to search other nectar sources. Should the maximum number of iterations be reached, the current optimal solution is passed as the optimal weights and threshold for the NN. The names used for each of the bees in the colony were the same found in the paper proposing this variant of ABC [60].

The evaluation of this approach used SVM, KNN, GMM and a simple neural network with BP model as baselines. The features used for these models were a combination of features present in the dataset, the best one resulted in 83.83% accuracy using the baseline NN. using this combination, the evaluation was conducted. The results show that the proposed approach performs significantly better than the simple NN, which itself performs better than the other baseline approaches, but unfortunately, the exact architecture is not disclosed, meaning these results can not be verified. The author concludes that a deep learning model could improve the results and should be further explored.

After a thorough review of the literature for Static MER, there are some takeaways of note. Due to the deep learning models capabilities, some approaches with state of the art performance utilize Mel-spectrograms as input, having a set of CNN layers for the effect, which results surpass even standard features with conjunction with Mel-spectrograms, meaning that a CNN as the capabilities of extracting the most relevant features from the visual representation of the audio. One CNN layer tend to implement a 3x3 layer, using a ReLU activation function and dropout layers, to prevent overfitting, as well as batch normalization and a pooling layer, varying between 3 to 5 CNN layers in total. Some approaches also employ an RNN in order to extract local features of the data related to the temporal nature of the emotional content of music. These approaches should be explored as well as experimenting with Data Augmentation to deal with the prevailing problem of small numbers of samples on the available datasets. Transfer Learning has also been more recently used with interesting results and should also be explored. For evaluation, a 10-fold cross validation procedure should be utilized in all approaches due to the small datasets available as discussed, and applying an F1-Score for analyzing performance as many of the reviewed approaches employ.

Table 2.4: Review of MER Classical ML approaches in the literature.

Static MER Review							
Author	Approach	Emotion Taxonomy	Database	Features/Input	Models	Results	Observations
Meyers [20]	Classic ML	Fusion of Hevner's with Russel's models (see Section 2.1.2)	372 songs samples	5 standard audio features (Mode, Harmony, Tempo, Rhythm, Loudness)	Decision Tree for classifying, KNN to cluster into quandrants	Evaluation performed against tags from different sizes	There are no statistical evaluation metrics for analyzing the performance of the chosen models.
Panda et al. [45]	Classic ML	MIREX task's five clusters	Based on MIREX taxonomy	11, 9 melodic and 2 standard audio features	Naïve Bayes, KNN and **SVM**	0.64 F1-Score	The taxonomy used, although supported by the MER community, isn't psychologically supported.
Markov et al. [47]	Classic ML	Russel's A-V, continuous approach	MediaEval 2014	Sub-set of features from database	SVM (different kernels) and **GP** (different models)	0.6986 R^2 for arousal and 0.3594 R^2 for valence	Features may be limiting the performance of the models, which may explain the very minor performance difference.
Bargaje [49]	Classical ML	Three-dimension Russel's model (Arousal, Valence and Loudness)	200+ randomly selected English and Hindi songs	600+ features extracted using audio information retrieval frameworks	GA for feature selection and SVM for clasiffication	84.57% accuracy with test data	The approach has some problems, such as a unbalanced dataset as well as not providing statistically relevant results for analyzing the performance of the approach adopted.
Panda et al. [8]	Classical ML	Discrete Russel's A-V (4 quadrants)	4QAED	29 novel and 71 standard features	SVM	0.76 F1-Score	The process of building the database is explained well, providing some good benchmark results as well.

Table 2.5: Review of MER Deep Learning approaches in the literature.

Static MER Review							
Author	Approach	Emotion Taxonomy	Database	Features/Input	Models	Results	Observations
Feng et al. [50]	Deep Learning	4 labels (happiness, sadness, anger and fear)	223 samples retrieved from available sources	3, 1 regarding tempo and 2 regarding articulation	Three layer feedforward network	67% precision and 66% recall	Early DL approach to MER, dataset isn't balanced at all, leading to biased results, and very few audio features are explored.
Choi et al. [52]	Deep Learning	50 emotional tags	MagnaTag ATune and MSD	Mel-spectrogram, STFT and MFCC	FCN and variations of this architecture	0.894 AUC for MagnaTagATune and 0.851 AUC for MSD	The use of the Million Song Dataset hinders the credibility of the provided results. The problem is treated as multi-class classification problem, not adopting widely used emotion taxonomies.
Choi et al. [53]	Deep Learning	50 emotional tags	Million Song Dataset	Mel-Spectrogram	Stacked model comprised of a CNN and a RNN portion, outputting to a FCNN	0.86% AUC score	As already mentioned, the Million Song Dataset has some questionable quality, using it hinders the credibility of the provided results.
Cañón et al. [54]	Deep Learning	Russel's A-V model	Speech Recognition and 4QAED	Spectrogram	CNN	0.48 F1-Score (as a classifier)	Results show emotion in music is co-related with language of speech.

Table 2.6: Continuation of the review on MER Deep Learning approaches in the literature.

Static MER Review							
Author	Approach	Emotion Taxonomy	Database	Features/Input	Models	Results	Observations
Grekow [56]	Deep Learning	Russels'A-V	324 six-second fragments from the corresponding samples of GTZAN[57]	124 from Marsyas and **529 from Essentia**	Different RNN architectures, without and **with a pre-trained model for feature selection**	0.73 and 0.46 mean squared error for arousal and valence, 0.11 and 0.12 MAE	Despite the pre-trained model approach achieving better results, a comparison with classic feature selection is not provided.
Yang [59]	Deep Learning	Russel's A-V	DEAM	Combination of features present in database (short-term energy, frequency spectrum, ...)	BP network with weight adjustment through the ABC evolutionary algorithm	0.89 MAE for valence and 0.92 MAE for arousal	Replicating the approach is not possible due to the architecture not being disclosed, pursuing a DL approach could yield better results.

2.7 MEVD

This section is a collection of reviewed literature concerning approaches for solving Music Emotion Variation Detection that provide interesting insights for the implementation of a robust architecture and methodology. The reviewed approaches are first presented and these are summarized in Table 2.7 in the end of this section.

2.7.1 Classical Machine Learning

According to the reviewed literature, the first approach for solving MEVD was proposed by Schubert [61] using linear regression models to predict the variation of a song's emotional content, for both arousal and valence, using 5 standard audio features: melodic contour, tempo, loudness, texture and spectral centroid. The dataset used for training and evaluating this model was built using the annotations from 67 participants, targeting Russell's A-V model, for four Romantic music pieces in one second intervals. The linear regression models were trained separately, due to traditional linear multiple regression models not being suited to time-series data. The evaluation found that loudness and tempo variation was correlated with variations in arousal, none of the reviewed features had much impact on valence, probably due to the omission of mode and articulation, features that were proven to influence valence [62], as stated by the author.

An approach based on SVM models to detect the variation of the emotional content of a song through time using the 4 quadrant categorical approach to the Russel's A-V model was presented by Panda et al. [63], part of a mood tracking platform. The features were extracted using the Marsyas framework and MIR toolbox, in part due to their ability to extract features for smaller clips. Using a 194 song dataset, with arousal and valence values for each sample of 25 seconds, the model was trained. For testing the model, two volunteers listened and annotated the changes between quadrants for the full duration of 57 songs in increments of 25 seconds. The annotations were analyzed in order to determine the agreement rating, which led to only 29 songs being used with the criteria of more than 80% agreement rate. The performance of the model attained an average of 44.08% accuracy. The size of the dataset, as is very much the case across the literature, greatly hinders the performance of the model. The authors also state that using a ranked set of features could result in a performance boost.

2.7.2 Deep Learning

Malik et al. [36] proposes a model for solving MEVD based on a CRNN architecture for predicting the A-V values in a 2-dimensional plane. The architecture, as presented in Fig. 2.29, can be explained as two smaller architectures. First, a CNN composed of 8 layers, implementing a 3x3 kernel, ReLU activation layers and a dropout layer. The output of the CNN is fed separately to two RNNs. This RNN architecture is comprised of a FCNN network with 8 layers outputting to 8 bidirectional GRUs, finally outputting the arousal or valence value, depending on the branch, after a maxout layer [64]. The dataset used for evaluating the performance of this architecture is part of the already discussed DEAM database in Section 2.2. This fragment of the dataset consists of 45 second long 431 audio samples from the Free music archive, using the first 15 seconds for the annotators to get familiar with the annotation process. This produced a set of 60 annotations for each 30 second sample, or an annotation for the emotional content of the song every 500 ms, based on A-V values on an [-1, 1] range.

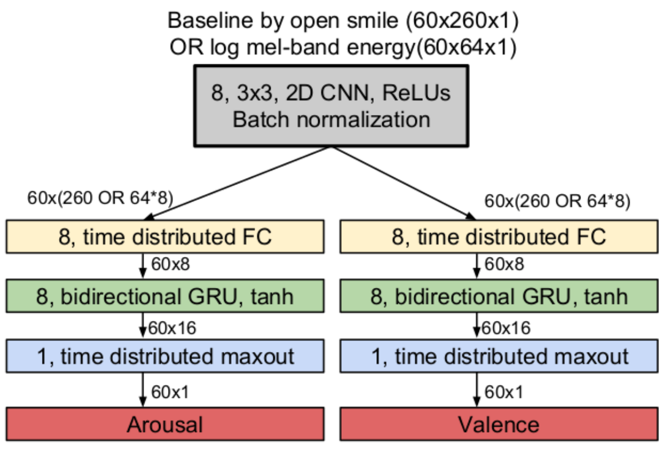

Baseline by open smile (60x260x1)
OR log mel-band energy(60x64x1)
8, 3x3, 2D CNN, ReLUs
Batch normalization
60x(260 OR 64*8)
60x(260 OR 64*8)
8, time distributed FC
8, time distributed FC
60x8
60x8
8, bidirectional GRU, tanh
8, bidirectional GRU, tanh
60x16
60x16
1, time distributed maxout
1, time distributed maxout
60x1
60x1
Arousal
Valence

Figure 2.29: CRNN approach presented by Malik et al. as found in [36].

The evaluation set consists of 58 full songs from the MedleyDb[17] dataset and music website Jamendo[18]. There were two approaches used for the input of the CRNN architecture: standard features retrieved from the openSMILE[19] toolbox present in the used database and another approach using only Mel-band features extracted with the python library librosa[20]. This last approach was stated as more applicable to the architecture, since the standard features can be inferred by the model itself. The best scored achieved by the proposed architecture reached a RMSE of 0.231 and 0.279 for arousal and valence, respectively, reaching a similar performance to the architecture presented by Li et al. [65]. It is important to note that this result was obtained with a slightly different configuration, where there is only one branch trained to output arousal and valence, not presenting the capabilities of separately training a branch for outputting a certain value.

Dong et al. [66] proposes a Bidirectional Convolutional Recurrent Sparse Network, or BCRSN, architecture which learns the sequential-information-included affect-salient features (SII-ASF) from spectrograms extracted from an audio signal. The architecture consists of a CNN as an input layer, 4 hidden layers, 2 forward and 2 backward RNN layers, the outputs of these layers being connected to a dense layer that predicted the emotion. The architecture can be explained in three key parts: bidirectional convolutional recurrent feature maps (BCRFM) learning, weighted hybrid binary representation (WHBR), and the objective function, concordance correlation coefficient (CCC);

- BCRFM learning consists in updating the weights of both the first forward and backwards RNN layers using the results of the convolutional operations between the input layer and these. The first hidden layers' outputs are then fed to the second hidden layer as input after being subsampled. These operations enable the model to learn the features intrinsic to the time-series data in the first hidden layers and further add to these more advanced features learned from the second hidden layers, as well as storing the overall sequential information in the last frame of the first hidden layers.
- In simple terms, weighted hybrid binary representation translates the numerical ground-truth data to a hybrid binary vector, meaning the output of the previously

[17]https://medleydb.weebly.com/
[18]https://www.jamendo.com/?language=pt
[19]https://www.audeering.com/research/opensmile/
[20]https://librosa.org/doc/latest/index.html

discussed part is the predicted value for every digit of the translated hybrid binary vector. Each neuron from the output layer of the previous part has their weight adjusted to control the convergence of the loss function. It is stated that by facing the problem as a weighted combination of multiple binary classification problem instead of a regression based prediction the A-V values for a sample, the computational complexity is reduced significantly, in turn reducing the time to produce a prediction;

- The last part is the objective function of the approach, of which CCC is a part of, proved to be a good performance measure for MEVD approaches, as it takes into account both he correlation coefficient and mean square error of the predicted and ground-truth data. The CCC between each segment predicted and target hybrid binary vectors is calculated, the average CCC of a sample as the quotient of summing the CCC of all segments multiplied by the respective loss and the number of segments related to the sample. However, the final objective function implements a least absolute shrinkage and selection operator, or lasso, in order to extract the SII-ASFs and control the sparsity of the feature maps of the BCRFM learning portion of the model.

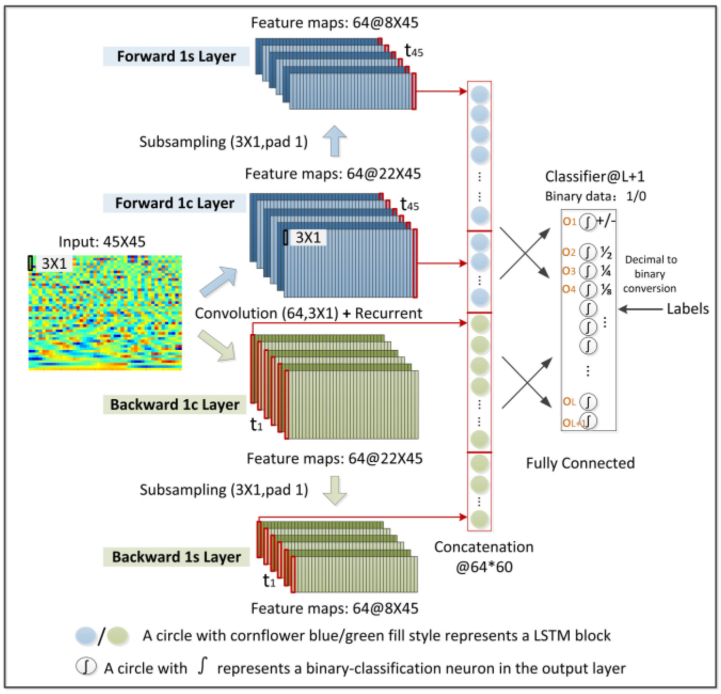


Figure 2.30: Presented BCRSN structure, as presented in [66]

The model, depicted in Fig. 2.30, was tested using a selected portion of the DEAM database as a benchmark, more specifically, a training dataset consisting of 431 music samples and an evaluation dataset consisting of 58 full-length songs, these providing an acceptable consistency of the annotations in contrast with the full database. The MTurk[21] dataset is used for testing the generalization of the model, consisting of samples of 240 American pop songs, each being annotated in 15s intervals by 7 to 23 annotators. These annotations are also analyzed in order to remove inconsistencies. The evaluation of this model show that it outperforms all compared models when using spectrograms as input, achieving an average RMSE of 0.123 for valence and 0.101 for arousal for the DEAM dataset, as well as the highest PCC and CCC scores for both valence and arousal. The same can be

[21]Built using Amazon Mechanical Turk: https://www.mturk.com/

observed in the MTurk dataset, as well when the input are the baseline features of the dataset in question, although the performance of the model may be hindered due to loss of emotional information when extracting the baseline features. Overall, this approach, although complex, is very robust, mostly due to its built-in feature extraction capabilities in the BCRFM portion of the model. Of note is the training time of the model, which despite the WHBR method introduced to mitigate the problem, is longer than two out of the three compared models. This is in part negligible due to the performance increase.

A very recent approach presented by Orjesek et al. [67] proposing two new architectures for MEVD, one based on the already discussed approach by Malik et al. [36] replacing the two-dimensionalCNN portion of the model with a one-dimensional one, as well as the GRU units with bidirectional GRU, or BiGRU, units, and another one similar to the aforementioned one, but swapping the dense layer between the CNN and BiGRU layers with a novel time-distributed layer with Iterative Reconstruction, or TR-IR, discussed ahead. The first proposed architecture is mainly used as a baseline, as stated by the authors, due to the similarity of the newly proposed model. The decision to change the CNN portion from a two-dimensional to a one-dimensional one is due to various studies stating that a one-dimensional CNN is capable of extracting relevant features from a audio signal representation, in this case a spectrogram, without prior pre-processing of the data [68]. The layer is composed by 8 filters, as well as applying batch normalization and the ReLU activation function, which is then fed to the dense layer. The resulting output is then fed to the 8 unit BiGRU layer to further learn the intrinsic temporal features from the received features, finally being fed to a dense layer with maxout for predicting the A-V values. The second proposed architecture has the same structure as the first proposed, changing the first fully connected layer with the already mentioned TR-IR layer. In simple terms, this layer is based on an autoencoder using the tanh activation function, which after a certain number of iterations, extracts the most relevant features from the input data. This is accomplished with the help of a boosting procedure of the features using a shared weight matrix, which with enough iterations, removes irrelevant features, leaving the most relevant ones as the output of the layer. The datasets used are the same as the ones used in the already discussed work by Dong et al. [66], a subset of the DEAM dataset consisting of 45 seconds long 431 music samples for the testing dataset and 58 full songs for the evaluation dataset. Results are compared against the best performing systems from the latest edition of the medieval "Emotion in Music" benchmark, as well as the BCRSN model [66] using both the baseline features and the spectrogram as input. The results show that the proposed architectures performed better overall against the compared architectures, specially in the valence dimension, where the second model achieved a 0.66 PCC in arousal and 0.637 in valence. The higher performance of the model is not as noticeable when comparing the achieved RMSE scores, underperforming in comparison with the BCRSN architecture with spectrogram as input in arousal and the LSTM-RNN architecture in valence.

As has been already stated, MEVD experiments were initially planned with a focus on implementing state of the art approaches found promising in the literature review. Various factors, such as resource constraints,inability to complete the extension of the team's MEVD dataset, and the very reduced size of the current dataset led to the decision of postponing these experiments until the update was concluded. The MEVD literature review is still present for future reference when developing the new methodologies.

Table 2.7: Review of MEVD approaches in the literature.

MEVD Review							
Author	Approach	Emotion Taxonomy	Database	Features/ Input	Models	Results	Observations
Schubert et al. [16]	Classical ML	Categorical Russell's A-V	4 romantic songs, annotated in 1 second intervals	5 features (melodic contour, tempo, loudness, texture and spectral centroid)	Linear regression for arousal and valence separately	0.33 accuracy in detecting changes	The dataset is very limited and contained in one genre.
Panda et al. [63]	Classical ML	Categorical Russell's A-V	57 songs, annotated in 25 second intervals	Standard audio features, such as timbre and rhythm	SVM	0.44 accuracy	The used dataset presents a very reduced size.
Markov et al. [47]	Classical ML	Categorical Rusell's A-V	MediaEval 2014, annotated in 0.4 second intervals	Standard audio features	GP regression	0.69 and 0.44 mean square error for arousal and valence	The complexity of applying this kinds of models may be infeasible in large scale.
Malík et al. [36]	Deep Learning	Continuous Rusell's A-V	Standard audio features / Mel-Spectrogram	CRNN	Continuous Rusell's A-V	0.231 and 0.279 RMSE for arousal and valence respectively	The duration of the samples may produce conflicting emotions, since 45 seconds may not encompass only one emotion.

Table 2.8: Review of MEVD approaches in the literature.

MEVD Review							
Author	Approach	Emotion Taxonomy	Database	Features/ Input	Models	Results	Observations
Dong et al. [66]	Deep Learning	Continuous Rusell's A-V	Portion of the DEAM dataset and Amazon's Mechanical Turk	Spectrogram	BCRSN	0.101 and 0.123 RMSE scores for arousal valence on the portion of DEAM and 0.079 and 0.145 RMSE for the Amazon's Mechanical Turk annotations	The model is very complex, requiring a very large amount of time for training.
Orjesek et al. [67]	Deep Learning	Continuous Rusell's A-V	Portion of the DEAM dataset, annotated in 0.5 second intervals	Standard audio features and Mel-spectrogram	Modified CRNN proposed by [36], adopting an IR time distributed layer	0.66 and 0.637 PCC for arousal and valence	The performance of the model can be disputed due to the achieved RMSE scores in comparison to the other models.

This page is intentionally left blank.

Chapter 3

Datasets and Features

There were two main datasets used for conducting the experimentation, along with variants. The reason for using these variants is explained in the course of this section.

3.1 Legacy-MERGE Audio

This dataset, which was used in the replicated past work, was previously described in Chapter 2 as the 4QAED, from here referred to as Legacy-MERGE, including the data available and distribution. In brief, the dataset is composed by 900 30 seconds clips from various songs that encapsulate the song's perceived emotion, these being mapped to the four quadrants of the discrete Russell's VA model in equal quantities, meaning that the dataset is balanced according to class, as seen in Fig. 3.1. The creation of this dataset was a byproduct of the work conducted by Panda [10] and later released with an accompanying baseline by Panda et al. [8].

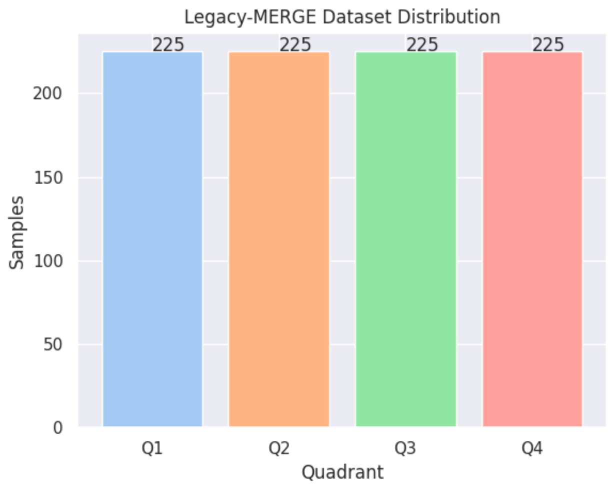
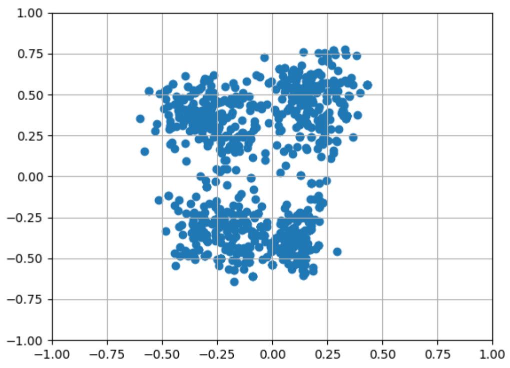
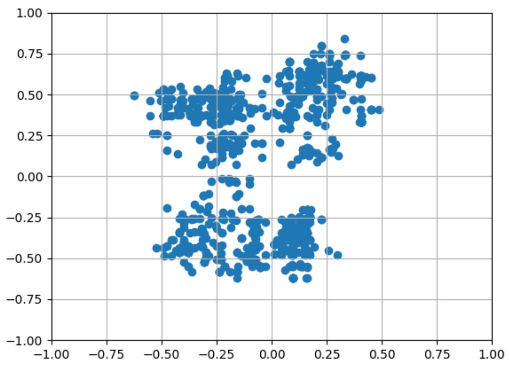


Figure 3.1: Legacy-MERGE distribution over each quadrant.

The process for creating the dataset began by extracting all available music from the All Music API, as mentioned before. From here, a very thorough selection process was conducted to eliminate all samples that presented dubious annotations from the "professional editors" credited for their creation. To make sense of the 289 existing emotion tags, not all from known emotional taxonomies, these were matched with the words present on Warriner's Affective Word list, which mapped VA values enable the annotation of each sample with a respective quadrant. A great number of samples were removed at this point since it was decided that only samples that had at least 50% of tags present in the list would be viable for the dataset, resulting in 2200 samples. After a manual inspection, samples which music clips presented low quality, due to being unclear or not encompassing the song's emotion, were removed. Remaining samples were manually annotated, removing ones that did not match with the quadrant attributed by the tags or without the annotator's agreement, resulting in 900 samples after balancing.



(a) A-V average values distribution



(b) A-V median values distribution

Figure 3.2: A-V values distributions for Legacy-MERGE used for experimentation.

The dataset contains not only the music clips and A-V values, obtained through the mean and averaging of the A-V values attributed to each tag, each distribution seen in Fig. 3.2 respectively, but also the standard and novel features presented by Panda et al. [8] for every clip, which are discussed more in depth in Section 3.3. This allows the dataset to be explored with a variety of Classical ML and DL approaches, through either classifiers or regressors.

A variation of this dataset, from here on referred to as Legacy-MERGE Corrected, was used in the development of the new approaches presented. This variation removes 7 clips, due to being duplicates or being mislabeled, and although it is no longer balanced as the main dataset, it showed better performance in preliminary experiments. The distribution of clips can be seen in Fig. 3.3.

3.2 New-MERGE Audio

The New-MERGE Audio dataset, referred to as New-MERGE from here on, is actually a collection of three new dataset candidates: New-MERGE Complete, New-MERGE Balanced and New-MERGE Balanced-Genre. All of the candidates are in some way an extension to the Legacy-MERGE dataset. The Complete candidate has all new samples that were deemed viable to be included in the extension, while the Balance candidate comprehends new samples while maintaining class balance, and the Balance-Genre candidate maintains class balance and balances each class in regards to the genres present. The new samples are from the previously mentioned selection process for creating the Legacy-MERGE dataset. Having no previous baseline, this dataset is tested for their viability as an extension, evaluating its quality. This is also an opportunity to validate previous hypotheses regarding the previous dataset, such as:

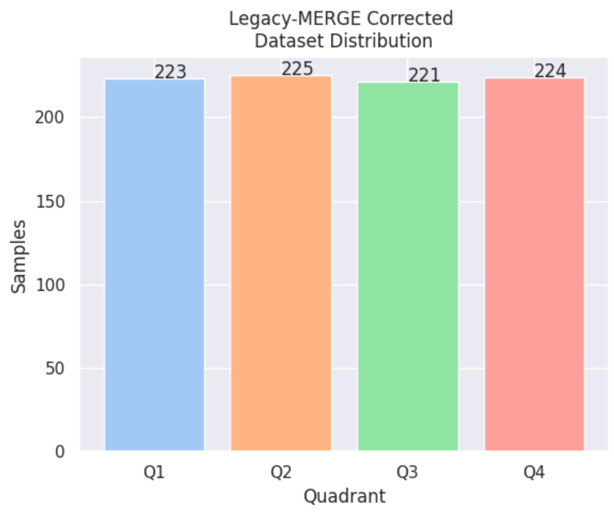


Figure 3.3: Legacy-MERGE Corrected distribution over each quadrant.

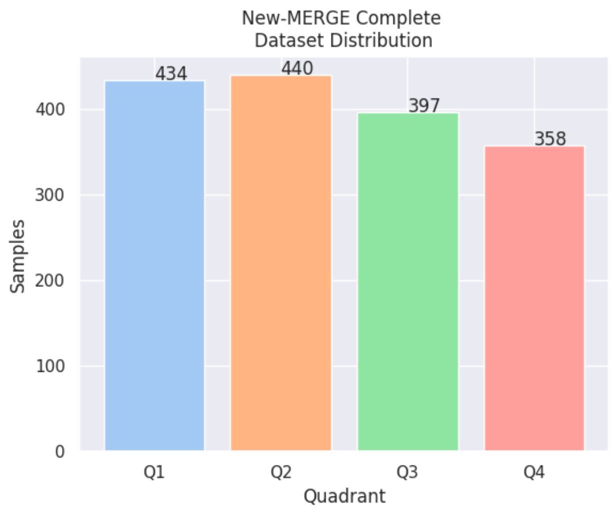


Figure 3.4: New-MERGE Complete distribution over each quadrant.

- Sample quantity increases the methodologies' performance;
- Equal genre distribution in each quadrant is as impactful to performance as equal quadrant distribution.

The obtained results may shed some light on these hypotheses, but only if it is indeed considered an improvement. Not only this, since the same hyperparameters found using Legacy-MERGE for the various methodologies, there is no guarantee that the models could perform better should a proper search be made.

Each candidate's sample distribution is depicted in Figs. 3.4, 3.5 and 3.6. The significantly unbalanced Complete candidate, having more Q2 and Q1 representatives than Q3 and Q4, may lead to some subpar performance due to a certain bias on samples with high arousal, which may lead to more difficulty in predicting the perceived valence of a song, a recurring problem in MER. The Balance candidate, although balanced, as hinted by the name, may suffer due to not taking into account the genre distribution, since it has been proven that certain genres are more dominant in certain quadrants, as explained in Section 4.1.6. The Balance-Genre candidate is the most promising candidate of the three, since it avoids the

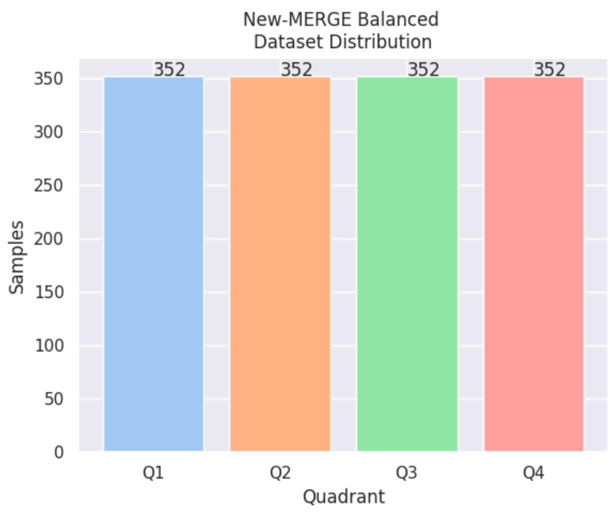


Figure 3.5: New-MERGE Balanced distribution over each quadrant.

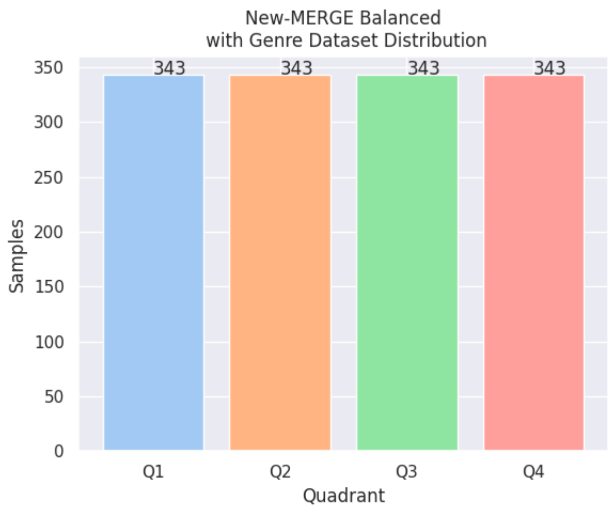


Figure 3.6: New-MERGE Balanced-Genre distribution over each quadrant.

pitfalls of the others, but it is also the least significant improvement to Legacy-MERGE, only being a 52% increase in sample quantity while the Complete candidate is a more exciting 81%, emphasizing how difficult it is to build a large and quality dataset.

3.3 Handcrafted Features

Present in both datasets are the handcrafted features previously mentioned. There are a total of 1714 features extracted from each sample which represent different musical dimensions, such as melody, harmony, rhythm, and others. Many of the original features developed fall into the expressive techniques dimension, which is dominated by high-level descriptors. The lack of high-level musical descriptors can be explained by the difficulty to accurately calculate compared to the low-level descriptors found in other dimensions [10].

The standard features encompass 1603 features extracted from existing audio information extraction toolboxes, namely and ordered by amount of features extracted, Marsyas, PsySound[1] and MIR Toolbox[2], previously evaluated in the literature, along with 558 original features extracted from the original audio signal and the same 558 features extracted from the isolated voice signal, amounting to 2719 features in total. In order to reduce the high dimensionality and after removing bugged features, the ReliefF [69] feature selection algorithm is used in order to assess the correlation between features, being set to a 0.9 threshold for removal as stated in the thesis, their predictive attributes, which are ranked from -1 to 1. The process is first conducted for the standard features, while a similar process was conducted for the original features, with the caveat of removing a novel feature if any standard feature is found to be similar enough to it.

[1]http://www.densilcabrera.com/wordpress/psysound3/

[2]https://www.jyu.fi/hytk/fi/laitokset/mutku/en/research/materials/mirtoolbox

This page is intentionally left blank.

Chapter 4

Methods and Experimentation

This chapter comprehends all experimentation conducted in the present work, including essential theoretical background to understand implementation choices and methodology followed for each experiment, as well as discussion on the obtained results, including the initial evaluation and further evaluation with the New-MERGE dataset, and conclusions taken from them. The chapter begins with the replication of previously developed methodologies, followed by the developed methodologies for this present work. The chapter concludes with a brief note on the evaluation methodology followed, as well as a summary on the conclusions from each experiment.

4.1 Replicating Past Work

Being a continuation of the work by Sá [7], referred to as foundation work from here on, experimentation started by replicating the methodologies previously evaluated, comparing against the reported results, as well as applying them to the new dataset candidates being finalized by the team. The rest of this section will consist of briefly explaining the experiments and comparing the obtained results for the dataset used in the foundation work against the already mentioned new dataset candidates, ending with an also brief discussion on them. The hyperparameters reported are the same used to replicate the experiments.

4.1.1 SVM

The initial baseline for the foundation work is the SVM classifier approach presented by Panda et al. [8] using the Top 100 features available in Legacy-MERGE, achieving a **76.4% F1 Score** using cross-validation with 10 splits and 10 repetitions. The achieved F1 Score in the foundation work was slightly lower, a 76%, and as stated, it was only replicated to serve as baseline for the developed methodologies.

As previously done in the foundation work, the implemented SVM model is based on the scikit-learn[1] Python package Support Vector Classification method, which receives as input the kernel to be used and necessary parameters for the implemented kernel.

For achieving optimal performance, the best hyperperameters for the model were searched, including the linear, polynomial, RBF and sigmoid kernels, as well as values for the cost, the degree for the polynomial kernel and gamma value for all kernels except the linear

[1]https://scikit-learn.org/stable/

kernel in an interval between 1e-6 and 100 using a bayesian optimization using Gaussian Process, as implemented in the already mention package.

For Legacy-MERGE, the optimal hyperparameters found used the RBF kernel with a cost of 28.264 and gamma 4.16e-04. The same was done for Legacy-MERGE Corrected to assess the impact on Classical ML, which found the sigmoid kernel with cost 100 and gamma 2.67e-04 as the best parameters.

Legacy-MERGE Corrected does not improve the results. This result is most likely explained by the removal of data, since it has been found in the literature that Classical ML approaches are solely dependent in the quantity and quality of the used features, meaning that reducing the number of samples hinders the training process of the model. As for Legacy-MERGE, it performs according to documented results, achieving around 1% worse performance than reported, as shown in Table 4.1.

Table 4.1: Confusion Matrix obtained for the SVM methodology for Legacy-MERGE.

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**1842**	144	59	205	**76.67%**
Q2	273	**1869**	62	46	**86.17%**
Q3	170	68	**1529**	483	**70.85%**
Q4	273	3	407	**1567**	**68.68%**
				F1 Score Total	**74.89%**

The results obtained for all New-MERGE candidates are presented in Table 4.2 and compared against the baseline results, using Legacy-MERGE, and between candidates.

Table 4.2: Results and comparisons obtained for New-MERGE dataset on SVM.

	F1 Score	Statistical Significance Test Against:		
		Baseline	Balanced	Balanced-Genre
Complete	**69.79%**	Significant (2.2447×10^{-18})	Not Significant	Not Significant
Balanced	**69.22%**	Significant (9.3508×10^{-22})	//	Not Significant
Balanced-Genre	**69.82%**	Significant (1.2314×10^{-16})	//	//

Results show a significant decrease in performance using all of the New-MERGE dataset candidates with SVM. This however may be due to some songs missing handcrafted features as mentioned before. Not only this, the used set of features may not be ideal, since feature selection was performed only for the Legacy-MERGE dataset.

4.1.2 Simple CNN

Following the same approach as Sá, we set the basic architecture used, here defined as Simple CNN, as our baseline. The architecture itself is a faithful reproduction of the one presented by Choi et al. [52]. The input of the architecture are full sample Mel-spectrograms, which will be utilized for most of the experiments conducted.

As for most of the experiments to be conducted, Mel-spectrograms are used, which are generated from .wav files containg the 30 second clips of each song on the dataset at

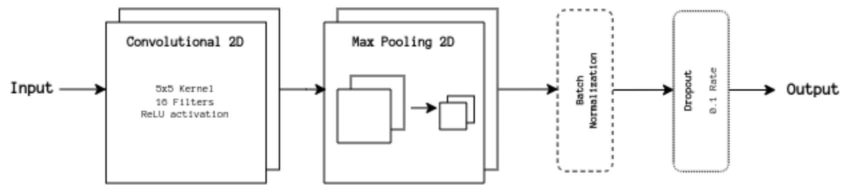


Figure 4.1: Diagram of the Conv 2D Block, the fundamental building block of the Simple CNN's feature extractor.

a 16kHz sampling rate and 128 filter bins, meaning that each second in the spectrogram corresponds to 16000 different datapoints on the raw signal and that there are 128 different frequency bins across it. These have been proven to be the most effective parameters to be used as input to the feature extractor in the foundation work.

The feature extraction portion consists of four Conv 2D blocks, each being a sequence of a Convolutional 2D, Max Pooling 2D, Batch Normalization and Dropout layers as seen in Fig. 4.1, except the last one which only has the first two layers. The output is flattened and passed to the classification portion of the network, comprising a Dropout and two Dense layers, the last one outputting one of the four quadrants in the discrete Russell's VA model. A diagram representing the described architecture is depicted in Fig. 4.2

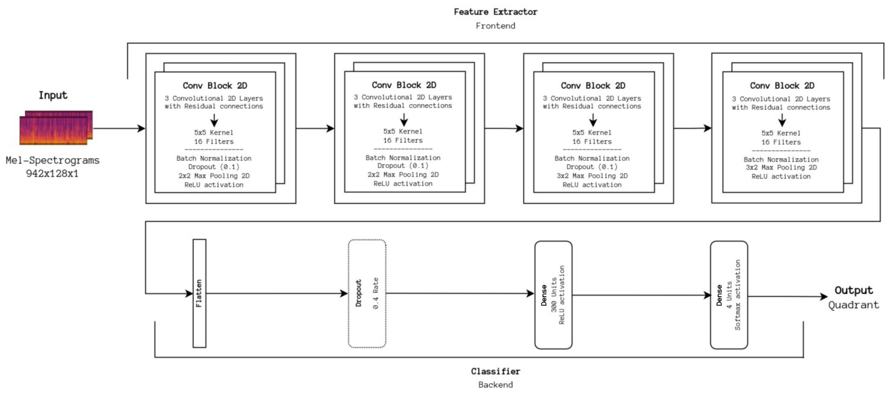


Figure 4.2: Diagram of the Simple CNN architecture. The frontend and backend of the model is clearly defined in this figure for future reference.

The replication results obtained was a **60.62% F1 Score** using Legacy-MERGE, as seen in Table 4.3, which is around a 3% decrease from the reported results. Most replicated results show a decrease, mostly are small and others fall around in the same range as this experience, and this may be due to the impossibility of replicating the results, which was already discussed previously in Section 1.2. For baseline purposes, the model was also trained using Legacy-MERGE Corrected, resulting in a **60.61% F1 Score**, which disproved our previous notion of an improvement over the original dataset. Nevertheless, the previous dataset had some mistakes and due to the similar performance, the Corrected version will still be used for evaluating newly developed approaches.

The results for the New-MERGE dataset, displayed in Table 4.4, show that an increase in number of samples does not directly translates to better performance, which is expected, since the hyperparameters were the same across all experiments, indicating that the New-MERGE dataset may still outperform with more optimal ones.

Table 4.3: Confusion Matrix obtained for Simple CNN

	Q1	Q2	Q3	Q4	F1-Score Per Quadrant
Q1	**1359**	470	217	204	**58.99%**
Q2	316	**1809**	93	32	**76.77%**
Q3	277	101	**1138**	734	**49.70%**
Q4	249	62	631	**1308**	**57.01%**
				F1 Score Total	**60.62%**

Table 4.4: Results and comparisons obtained for New-MERGE dataset on Simple CNN.

	F1 Score	Statistical Significance Test Against:		
		Baseline	Balanced	Balanced-Genre
Complete	**61.66%**	Not Significant	Not Significant	Significant (0.0245)
Balanced	**60.97%**	Not Significant	//	Not Significant
Balanced-Genre	**60.28%**	Not Significant	//	//

4.1.3 Split CNN

To deal with the less than ideal amount of data for a DL experiment, each sample was sliced in half, essentially doubling the amount of samples available for training a model. The Simple CNN architecture was taken as the foundation for evaluating this approach, tuning the input layer to accept Mel-spectrograms of half the size to accommodate the "half-samples", as seen in Fig. 4.3.

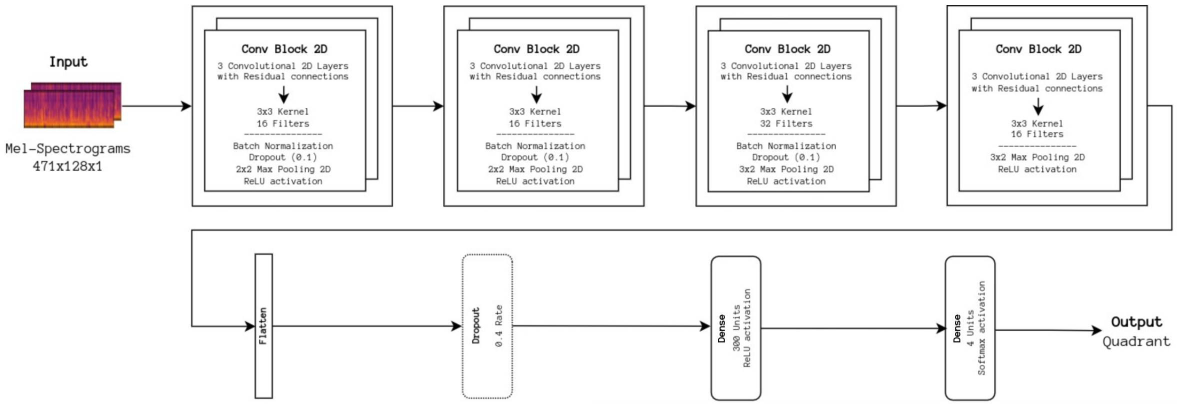


Figure 4.3: Diagram of the Split CNN architecture.

Replicating the experiment produced a **64.77% F1-Score**, using Legacy-MERGE, as seen in Table 4.5, a decrease of around 2% when compared to the reported results.

Splitting samples for New-MERGE degrades the performance of the model significantly when compared with the baseline, as can be seen in Table 4.6. This may indicate that splitting the new songs' clips introduced in New-MERGE is not preserving the emotional content found in the full clips.

Table 4.5: Confusion Matrix obtained for Split CNN

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**3027**	885	263	325	**64.41%**
Q2	517	**3793**	114	76	**80.03%**
Q3	640	197	**2372**	1291	**55.85%**
Q4	647	122	1126	**2605**	**58.75%**
				F1 Score Total	**64.77%**

Table 4.6: Results and comparisons obtained for New-MERGE dataset on Split CNN.

	F1 Score	Statistical Significance Test Against:		
		Baseline	Balanced	Balanced-Genre
Complete	**62.95%**	Significant (0.0031)	Not Significant	Not Significant
Balanced	**62.79%**	Significant ($7.2376\text{x}10^{-04}$)	//	Not Significant
Balanced-Genre	**62.30%**	Significant ($1.1261\text{x}10^{-04}$)	//	//

4.1.4 Simple CNN with Audio Augmentations

In the MER and related fields literature, some classical audio augmentation approaches were found to be helpful when training models. For MER, the pool of augmentation techniques explored is limited and mostly falls on the ones experimented with by Sá. The techniques explored used were explained back in Section 4.4.

Individually, each augmentation technique made a positive impact on the overall performance, except for time stretching. This can be explained by the difficulty in preserving the sentiment of an augmented sample, which is explored later in Section 4.4.1. In brief, applying time stretching negatively to a sample on a quadrant of high arousal, which has normally high energy and fast tempo (first and second quadrant), may be perceived as having lower arousal and as a result, being predicted as belonging to the third or fourth quadrant. The reverse also applies, higher tempo in a sample that originally belonged to the third or fourth quadrant, may place it on the second or third.

Applying all four augmentation techniques was also evaluated, producing the best results. Only the all-encompassing experiment is replicated for this reason and as a way to save resources for original experiments. Results are a lot lower than reported, achieving around **59.92% F1-Score** with Legacy-MERGE, as seen in Table 4.7. Despite the expected differences due to the randomness aspect inherent to the augmentation process, there is a need to investigate each augmentation technique contribution performance individually, since the use of Time Stretch is reported to significantly degrade the performance, which may have contributed to the obtained results.

As can be seen in Table 4.8 for New-MERGE, the increase in number of samples with augmentations is substantial enough to produce statistical significant results with a 0.0112 p-value, which shows that the dataset indeed is an improvement on Legacy-MERGE. As previously discussed, further investigating the individual performance of augmentation techniques may also lead to an increase in performance.

Table 4.7: Confusion Matrix obtained for Simple CNN with Audio Augmentations

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**1364**	327	296	263	**59.00%**
Q2	359	**1741**	70	80	**78.43%**
Q3	346	68	**1068**	768	**48.23%**
Q4	250	33	706	**1261**	**54.03%**
				F1 Score Total	**59.92%**

Table 4.8: Results and comparisons obtained for New-MERGE dataset on Simple CNN with Audio Augmentations.

	F1 Score	Statistical Significance Test Against:		
		Baseline	Balanced	Balanced-Genre
Complete	**61.55%**	Significant (0.0112)	Not Significant	Not Significant
Balanced	**61.13%**	Significant (0.0665)	//	Not Significant
Balanced-Genre	**60.41%**	Significant (0.4551)	//	//

4.1.5 Simple CNN with GAN generated samples

A GAN, as explained in Section 2.3, has the ultimate goal of generating artificial samples as close to the real samples used for training the network. The process starts by training an autoencoder, discussed ahead in 4.4.2, on the real samples in order to get the most optimal latent space, from where the embeddings, samples projected in the latent space, should accurately be reconstructed to their original representation. This can be difficult if not enough data is available, as is the case, which leads to a large latent space dimension, since there is not enough data to reduce it and maintain the accuracy of the reconstruction.

After having the autoencoder trained, all samples are embedded into the latent space to calculate the mean and covariance matrix for the samples in each class, in our case, in each quadrant, for sampling the corresponding distributions. To train the GAN, the discriminator is first trained on real samples and then on fake samples from the generator. Only one is trained at a time, meaning that after training the discriminator, the generator is trained by evaluating the accuracy of the discriminator on distinguishing real and fake

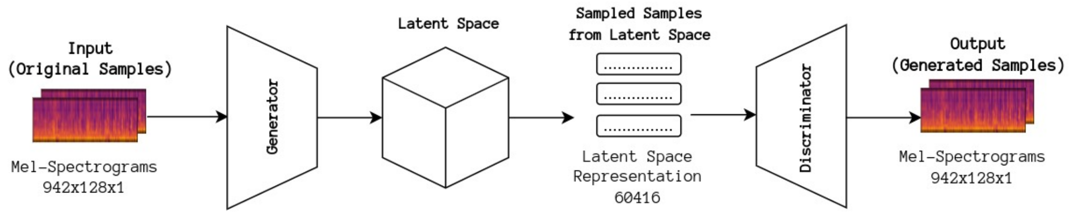


Figure 4.4: Process of generating new samples using a GAN. After being trained, the original Mel-spectrograms of the samples are passed through the generator, which embedds the samples into the learned latent space. From here, the desired samples are generated and passed through the discriminator, where these are reconstructed as accurately as possible to the dimensions of the used Mel-spectrograms.

samples. The GAN is ready to be used after a number of epochs or, ideally, when the discriminator is unable to differentiate real and fake samples. The process can be seen in Fig. 4.4.

Replicating this experiment proved to be a difficult task, since the covariance matrix for each quadrant is dependent on the size of the latent space, the smallest possible being 60416, meaning that we have four 60416x60416 matrix to work with. Processing this matrix along with some intermediate calculations needed for each quadrant made generating samples with a GAN the most time consuming methodology to be replicated.

The results obtained for this approach reached a **52.36% F1 Score** with Legacy-MERGE as seen in Table 4.9, which is a very huge decrease in performance from the reported **60.44% F1 Score**. Although a decrease in performance was expected compared with the baseline model, the uncertainty of certain details for replicating the GAN's training process seem to have made a very considerable negative impact on the results. Nonetheless, we expect that by using more samples some of this negative impact can be mitigated.

Table 4.9: Confusion Matrix obtained for Simple CNN with GAN

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**1168**	447	366	239	**49.39%**
Q2	534	**1526**	120	70	**67.84%**
Q3	394	117	**1000**	739	**43.89%**
Q4	337	88	759	**1066**	**48.31%**
				F1 Score Total	**52.36%**

To prove if more samples are indeed enough for improving the latent space representations, we look at the results for New-MERGE, shown in Table 4.10.

Table 4.10: Results and comparisons obtained for New-MERGE dataset on Simple CNN with GAN

	F1 Score	Statistical Significance Test Against:		
		Baseline	Balanced	Balanced-Genre
Complete	**54.75%**	Significant (0.0052)	Not Significant	Not Significant
Balanced	**55.89%**	Significant ($2.7965\text{x}10^{-05}$)	//	Not Significant
Balanced-Genre	**55.81%**	Not Significant	//	//

Here, the results show a statistically significant improvement in performance, which may be due to the increase in accuracy of the GAN model, as it was expected. Since more samples are available, the mappings of the latent space become more accurate, generating more relevant artificial samples.

4.1.6 CNN Model Pre-Trained with Genre

As stated in [7], there is some correlation between music genre and emotion, since some genres have a tendency to fall in particular quadrants, such as Pop and Rock being predicted as Q1, or R&B and Religious (such as choir music) being predicted as Q4. With this in mind, it makes sense to give some notion of music genre to a network to hopefully unveil some patterns relevant to emotion recognition.

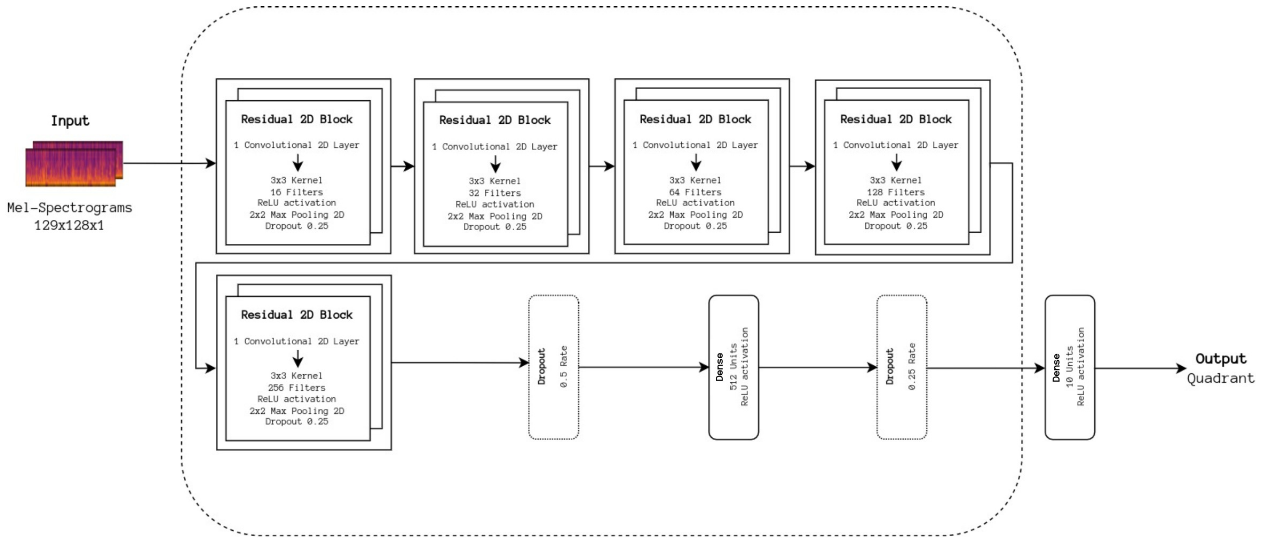


Figure 4.5: Diagram of the Genre CNN architecture.

For this experiment, the GTZAN dataset [57] was used to train a well performing genre classification model[2], reporting a **83.20% F1 Score**. It can be seen in Fig. 4.5. This dataset is composed of 1000 songs, each having a corresponding genre, which includes blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae and rock. Details on the process of collecting annotations for compiling the dataset are not disclosed.

After training, the model's weights are frozen and the last layer, which outputs to genre, is replaced by three Dense layers, which last layer outputs to one of the four quadrants, and is once again trained, resulting in a model that predicts emotional quadrants.

The evaluation process was conducted by splitting full samples, each 30 seconds, to 1.5 seconds segments, feeding the segments to the network. Full sample predictions are obtained by majority voting, meaning that the mode of all segments' predictions is considered the full sample prediction.

Looking at Table 4.11 shows an obtained **19.23% F1 Score**, much in line with the reported results. This experiment shows how different the relevant features for determining genre and emotion, and that simple using out-of-domain knowledge is not the most promising direction for improving performance.

Table 4.11: Confusion Matrix obtained for CNN model pre-trained on GTZAN.

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**164**	539	503	1044	**14.23%**
Q2	80	**522**	535	1113	**15.78%**
Q3	136	476	**556**	1082	**16.28%**
Q4	108	541	565	**1036**	**24.37%**
				F1 Score Total	**19.23%**

Despite the mediocre prediction ability of this network, the New-MERGE dataset was experimented for possible improvements, as shown in Table 4.12.

[2]https://github.com/Hguimaraes/gtzan.keras

Table 4.12: Results and comparisons obtained for New-MERGE dataset on CNN model pre-trained on GTZAN.

	F1 Score	Statistical Significance Test Against:		
		Baseline	Balanced	Balanced-Genre
Complete	**16.96%**	Significant ($1.0961\text{x}10^{-04}$)	Significant (0.0061)	Significant (0.0049)
Balanced	**18.40%**	Not Significant	//	Not Significant
Balanced-Genre	**18.55%**	Not Significant	//	//

The Complete candidate significantly decreases the performance of the model, which may be explained by its unbalanced nature. This is further corroborated with the results obtained for the balanced candidates, which maintain the performance of the model. Naturally, when more genres are equally represented the more accurate the model's performance is.

4.1.7 Voice CNN

As was reported by Panda et al. [8] and mentioned by Sá, voice-specific features tend to be correlated with sad or calm songs, which fall in low valence quadrants, namely, Q3 and Q4. In order to further explore this finding, Spleeter[3], an audio source separation tool developed by Deezer, a very large music streaming service akin to Spotify, is used to separate voice from all samples and used in tandem with the full sample as Mel-spectrograms to train a network using the baseline as foundation.

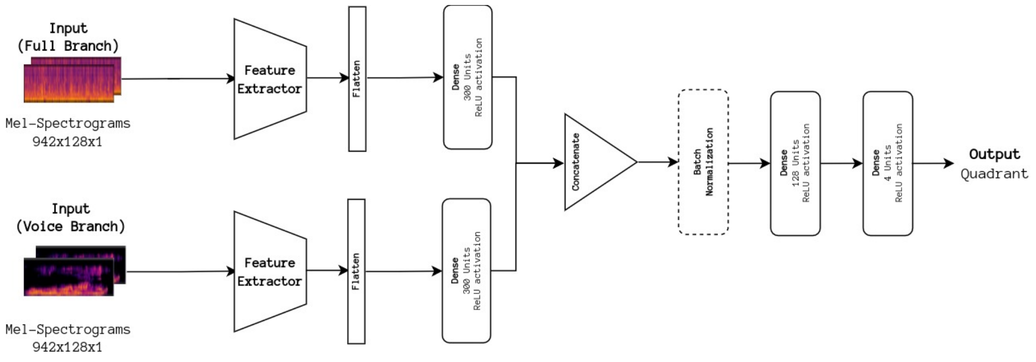


Figure 4.6: Diagram of the Voice CNN architecture.

Different from the previous experiments, the feature extractor portion is used for the full sample and for the voice isolated sample, each one an input branch for the network. The output from each branch is concatenated into a feature vector and the classification portion outputs the prediction.

Results show an obtained **61.35% F1 Score** for Legacy-MERGE, has seen in Table 4.13, which is not far from the reported results. New-MERGE results are shown in Table 4.14.

Performance did not change as the statistical results were not significant. This may indicate that the voice branch may not introduce enough relevant information for improving the

[3]Spleeter repository: https://github.com/deezer/spleeter

Table 4.13: Confusion Matrix obtained for Simple CNN with isolated Voice branch.

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**1347**	481	198	224	**60.05%**
Q2	300	**1826**	82	42	**77.56%**
Q3	295	93	**1125**	737	**51.23%**
Q4	254	65	622	**1309**	**56.56%**
				F1 Score Total	**61.35%**

Table 4.14: Results and comparisons obtained for New-MERGE dataset on Simple CNN with isolated Voice branch.

	F1 Score	Statistical Significance Test Against:		
		Baseline	Balanced	Balanced-Genre
Complete	**61.44%**	Not Significant	Not Significant	Not Significant
Balanced	**60.58%**	Not Significant	//	Not Significant
Balanced-Genre	**60.63%**	Not Significant	//	//

performance.

4.1.8 Double Branch CNN Regressor

Beyond the discrete quadrant labels, our dataset also provides continuous arousal and valence values in the AV plane, meaning that it is possible to train a regressor on the data. These AV values were obtained by obtaining the tags available for each sample from All Music and comparing the relevant ones with Warriner's Adjective List, which lists 13,915 English words and their respective rankings in arousal, valence and dominance, this last one being irrelevant for the purposes of the dataset. The rankings for each tag were averaged for each sample and these were taken as the placement of each sample in the plane, as elaborated in [10].

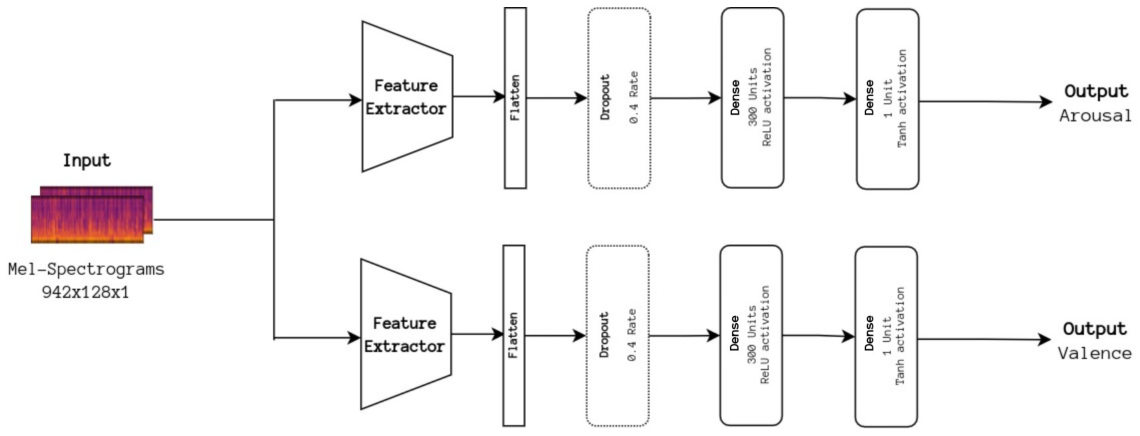


Figure 4.7: Diagram of the Double Branch CNN Regressor architecture. The feature extractor portion is the same as the baseline.

Table 4.15: Confusion Matrix obtained for Double Branch CNN Regressor

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**986**	359	123	772	**47.67%**
Q2	430	**1405**	146	169	**68.97%**
Q3	262	106	**509**	1433	**28.80%**
Q4	169	24	351	**1756**	**54.31%**
				F1 Score Total	**49.93%**

The network itself, presented in Fig. 4.7, is composed of two branches, each with its own feature extraction portion, which takes a full sample Mel-spectrogram as input, feeding into the classifier portion, outputting either arousal or valence values depending on the branch. Each branch tries to predict arousal or valence, being trained against the AV values already discussed. Here, not only is the F1-Score relevant, but also the Root Mean Squared Error between the predictions and ground-truth data.

The results show that the feature extraction portion does not feed relevant features to the regressor with a significant decrease in performance, obtaining a **49.93% F1 Score** at best, as shown in Table 4.15. Despite the match rate for A-V mapped values and actual quadrants being around 96%, the extracted features do not reflect this. The bad performance of the A-V values should be further tested against verification to assess the emotional tags accuracy for each song in our dataset.

Do to missing emotional tags, some songs were unable to be mapped to the A-V plane for the all candidates of the New-MERGE dataset. Due to time constraints, it was decided to proceed experimentation with the available songs. This decreases encompasses 282, 226 and 322 songs for the Balanced and Balanced-Genre candidates, resulting in 1347, 1178 and 1050 songs. The obtained results can be seen in Table 4.16.

Table 4.16: Results and comparisons obtained for New-MERGE dataset on Double Branch CNN Regressor

	F1 Score	Statistical Significance Test Against:		
		Baseline	Balanced	Balanced-Genre
Complete	**45.94%**	Significant ($2.0355\text{x}10^{-11}$)	Not Significant	Significant ($2.1544\text{x}10^{-04}$)
Balanced	**45.94%**	Significant ($4.1245\text{x}10^{-12}$)	//	Significant (0.0399)
Balanced-Genre	**43.19%**	Significant ($4.4920\text{x}10^{-17}$)	//	//

After some investigation, the significant decrease in performance appeared to be explained by the low match rate of mapped A-V values with the actual quadrant labels. More specifically, around 50% of the songs do not correctly match the mapped A-V values for the Complete candidate, which makes this model highly inaccurate. There is a need to understand why this low rate was encountered.

4.1.9 DNN

These features, as for the SVM, were used to train a DNN with the same sets as mentioned previously: all features, top 100 and top 100 + Spotify's 11 features. These last 11 Spotify

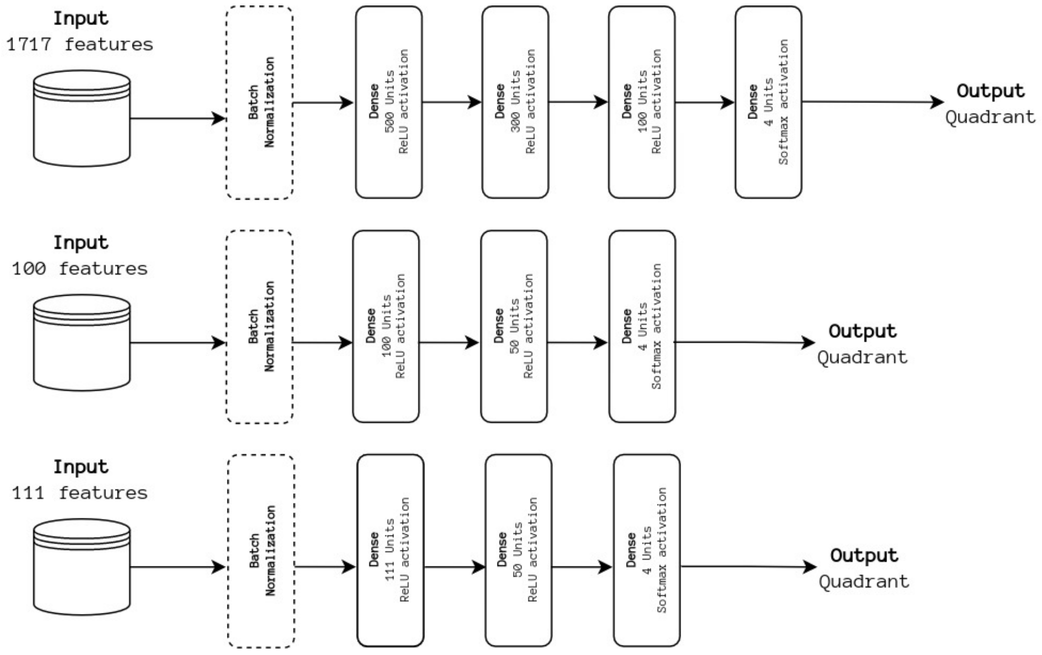

Input
1717 features
Batch Normalization
Dense 500 Units ReLU activation
Dense 300 Units ReLU activation
Dense 100 Units ReLU activation
Dense 4 Units Softmax activation
Output
Quadrant
Input
100 features
Batch Normalization
Dense 100 Units ReLU activation
Dense 50 Units ReLU activation
Dense 4 Units Softmax activation
Output
Quadrant
Input
111 features
Batch Normalization
Dense 111 Units ReLU activation
Dense 50 Units ReLU activation
Dense 4 Units Softmax activation
Output
Quadrant

Figure 4.8: Diagram of the DNN models. From top to bottom: 1714 Decorrelated features, 100 Top Features and 100 Top + 11 Spotify features.

Table 4.17: Confusion Matrix obtained for DNN with All 1714 Decorrelated Features

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**1622**	288	136	204	**71.06%**
Q2	277	**1847**	74	52	**81.28%**
Q3	115	106	**1457**	572	**64.52%**
Q4	292	54	580	**1324**	**60.00%**
				F1 Score Total	**69.21%**

features were only available for 704 songs, considerably reducing the number of samples.

The architectures for the networks reported are rather small due to the small number of samples, which is more accentuated in this experiment, since large DNN's are mentioned to overfit very rapidly [7]. The architecture for processing all features has only five layers, a Batch Normalization at the start and four Dense layers at the end for classification, having 500, 300, 100 and 4 output units respectively, outputting to one of the four quadrants, as can be seen in Fig. 4.8. To process the top 100 features, the previous architecture is maintained, but the first Dense layer is removed and the rest's number of units are reduced, to 100, 50 and 4 respectively. For the last set, the previous network is reused, but the Batch Normalization layer is removed and the first Dense layer output units are matched with the number of input features.

The models with all 1717 decorrelated features (Table 4.17), top 100 features without (Table 4.18) and with 11 Spotify features (Table 4.19) obtained a **69.21%**, **72.48%**, and **73.73% F1 Score** respectively for Legacy-MERGE. Some of the decorrelated features still seem to introduce some noise, due to the better performance of the top 100 features model. Surprisingly, the reduced number of samples did not matter since the last model outperforms the ones before significantly.

There was a reduction of the available samples for the balanced datasets due to an oversight, which was already mentioned back in 4.1.1. The experiment proceeded with a decrease

Table 4.18: Confusion Matrix obtained for DNN with Top 100 Features

	Q1	Q2	Q3	Q4	F1-Score Per Quadrant
Q1	**1749**	187	112	202	**74.11%**
Q2	281	**1852**	68	49	**84.18%**
Q3	156	88	**1494**	512	**67.17%**
Q4	278	20	512	**1440**	**64.45%**
				F1 Score Total	**72.48%**

Table 4.19: Confusion Matrix obtained for DNN with Top 100 Features + 11 from Spotify

	Q1	Q2	Q3	Q4	F1-Score Per Quadrant
Q1	**1427**	122	90	161	**76.33%**
Q2	148	**1446**	40	36	**87.33%**
Q3	121	53	**1105**	431	**66.00%**
Q4	239	15	394	**1212**	**65.28%**
				F1 Score Total	**73.73%**

that encompasses 226 and 322 songs for the Balanced and Balanced-Genre candidates, resulting in 1178 and 1050 songs. Results for the 1714[4] Decorrelated features and top 100 features for the New-MERGE dataset can be seen in Tables 4.20 and 4.21.

Table 4.20: Results and comparisons obtained for New-MERGE dataset on DNN with All 1714 Decorrelated Features.

	F1 Score	Statistical Significance Test Against:		
		Baseline	Balanced	Balanced-Genre
Complete	**68.63%**	Not Significant	Not Significant	Not Significant
Balanced	**68.10%**	Not Significant	//	Not Significant
Balanced-Genre	**68.05%**	Not Significant	//	//

Table 4.21: Results and comparisons obtained for New-MERGE dataset on DNN with Top 100 Features.

	F1 Score	Statistical Significance Test Against:		
		Baseline	Balanced	Balanced-Genre
Complete	**67.40%**	Significant ($1.5032\text{x}10^{-14}$)	Not Significant	Not Significant
Balanced	**67.41%**	Significant ($1.6082\text{x}10^{-13}$)	//	Not Significant
Balanced-Genre	**67.41%**	Significant (2.143110^{-13})	//	//

An overall decrease in performance is significant in the Top 100 features model, which

[4]10 of these features had missing values, and for this reason were removed. It was latter found that this was also the case for the Legacy-MERGE dataset, were these missing values were changed to 0. Future experiments will be conducted to assess the performance differences.

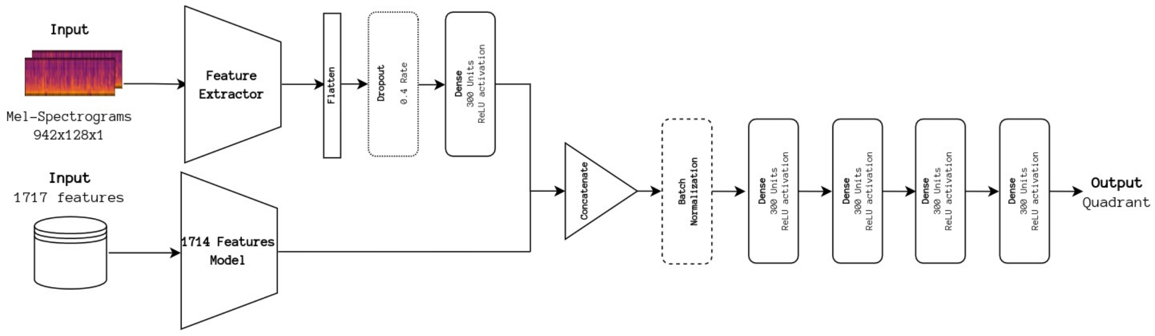


Figure 4.9: Diagram of Hybrid CNN + DNN.

Table 4.22: Confusion Matrix obtained for Hybrid CNN + DNN

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**1635**	234	130	251	**70.07%**
Q2	322	**1786**	91	51	**81.33%**
Q3	164	83	**1272**	731	**58.65%**
Q4	242	33	565	**1410**	**59.85%**
				F1 Score Total	**67.63%**

strongly suggests that the most relevant features are not the same as the ones found for the Legacy-MERGE dataset. As future work, feature selection needs to be redone in order to improve performance.

4.1.10 Ensemble - Hybrid CNN + DNN

As a way to take advantage of the different information that can be leveraged from the spectral representation of an audio sample and the hand-crafted features calculated from it, an ensemble model, comprised of the models seen in Fig. 4.9, fusing both information is experimented with.

As shown above, the ensemble has a CNN branch and a DNN branch. The DNN portion is pre-trained previously in order to find the best performing model from the various training folds and the weights are saved. To train the whole model, first the best weights for the DNN are loaded and its layers are frozen. Then, the model is fed both the features and the full samples Mel-spectrograms as input, finally outputting to one of the four quadrants.

The **67.63% F1 Score** result, as shown in Table 4.22, indicates a significant improvement compared with the baseline, but a significant decrease against the DNN only approach, which means makes this model not state of the art as previously reported. This was found to be a bug on the code available for these experiments, which after being fixed, resulted in the mentioned score Reasons behind this may include the lower performance of the feature extraction portion negatively impacting the whole model.

New-MERGE candidates' results are shown in Table 4.23.

No significant changes in performance are noted by using the New-MERGE candidates.

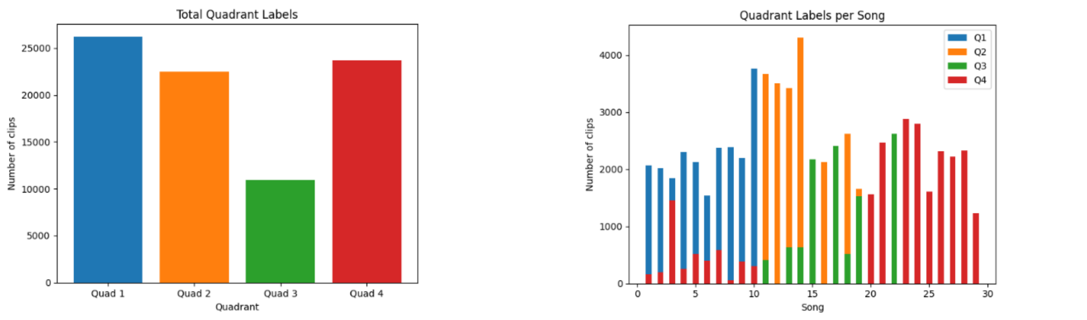
Table 4.23: Results and comparisons obtained for New-MERGE dataset on Hybrid CNN + DNN

	F1 Score	Statistical Significance Test Against:		
		Baseline	Balanced	Balanced-Genre
Complete	**67.34%**	Not Significant	Not Significant	Not Significant
Balanced	**66.54%**	Not Significant	//	Not Significant
Balanced-Genre	**67.30%**	Not Significant	//	//

4.1.11 MEVD Experiments

Although it was not pursued in the present work, in the foundation work there were also some methodologies developed and evaluated for solving MEVD.

These were conducted using the dynamic dataset from Panda et al. [63] which derived from a previous dataset presented by Yang et al. [70] containing 194 clips of songs, each 25 seconds, annotated with arousal and valence values. The dynamic dataset contains 29 entire songs, with continuous arousal and valence annotations. The process for obtaining these 29 songs started by removing the oriental songs, keeping 57 of the original 194. From here, the full songs were extracted and continuously annotated by two subjects. Should the agreement between the annotations of one song not reach 80%, the song would be removed, eventually leading to the 29 songs on the dataset. This necessary process to ensure the quality of the samples came with balancing issues, as it can be seen in Fig. 4.10, Q1 is overrepresented and Q3 is heavily underrepresented, which makes it difficult to properly train a DL model without bias to the more represented quadrants, as well as keeping it from rapidly overfitting. Another problem is the quadrant distribution of annotations in a concrete song, as can also be seen in Fig. 4.10, there are many songs that have annotations for only a quadrant, which may also impact the predictions of the model. The difficulty in not only balancing the number of samples for each quadrant, but also ensuring some balance in same song annotations and the process of obtaining these annotations, show how much more complex solving MEVD is compared to static MER.



(a) Quadrant distribution of the MEVD dataset, as seen in Sá [7].

(b) Quadrant distribution of all segment-level clips on the MEVD dataset, as seen in Sá [7].

Figure 4.10: MEVD dataset quandrant distribution.

The conducted experiments' architecture are almost the same between each other, using the feature extraction portion without the last layer. For the classifier portion however, experiments were conducted with a unidirectional and a bidirectional LSTM unit, which output is fed in both to two dense layers for classification. The input for this architecture are 2 second clips with an overlap of 100 milliseconds, meaning that each clip contains 100 milliseconds of the previous and/or next clip belonging to the same song, outputting the predicted quadrant.

Another important point of the evaluation process of these models was the cross-validation methodology used. Instead of the 10 splits 10 repetitions methodology followed until this point, the number of splits was reduced to 4. This was necessary to maintain a good proportion between the train and test sets, since the train set should be considerably larger than the test set, and to ensure that there is a reasonable quadrant distribution in the test set.

Results are much in line with the reported, achieving a **21.22%** and a **20.95% F1 Score** for the networks utilizing LSTM and Bi-LSTM units respectively. Has was already mentioned in the foundation work, MEVD is a very complex task for which there is not nearly enough data for DL approaches to take advantage.

Taking into account the very limited size of the dataset and the very unbalanced distribution, it was decided to not proceed with any experiments due to the impossibility of finalizing the updated MEVD dataset, since it is expected that the results would not differ too much with any newly developed methodology for the current dataset.

Despite no MEVD experiments were conducted, it was still important to understand the experiment for future work and the reported results were also validated.

4.2 New Architectures

In this section, we experimented with current implementations researched in the state of the art. We begin by studying segment-level methodologies, one using segments of Mel-spectrograms from full song clips by averaging the predictions of each segments, and then proceeding to another directly extracting features from the raw signal of the song clip.

4.2.1 ShortChunk CNN

Despite some earlier literature mentioning that the length of a sample that contains the whole emotional envelope of a song is about 30 seconds, more recent studies have shown that this is not always the case. As shown in various works [71] [72] [25], the length of a sample can be reduced as low as 3 seconds and still produce state of the art performance in some datasets, such as the MTG-Jamendo dataset, and recently in MediaEval [40]. Previous work on Legacy-MERGE has also shown that by using half of a sample as a full sample, performance increases despite the decrease of the sample length [7].

The approach in question has been experimentally evaluated using the above mentioned dataset by Won et al. Taking Mel-spectrogram representations of the samples as input, this approach uses 7 Residual 2D layers in sequence, a Global Max Pooling 2D layer and a sequence of Dense, Batch Normalization, Dropout and another Dense to make the classification. It is important to note that a direct comparison cannot be made between the reported results and the results from Legacy-MERGE since the problem is not the same. The original experiments approached MER as a multi-tag classification problem, meaning that multiple tags can be attributed to one sample, whereas here we approach

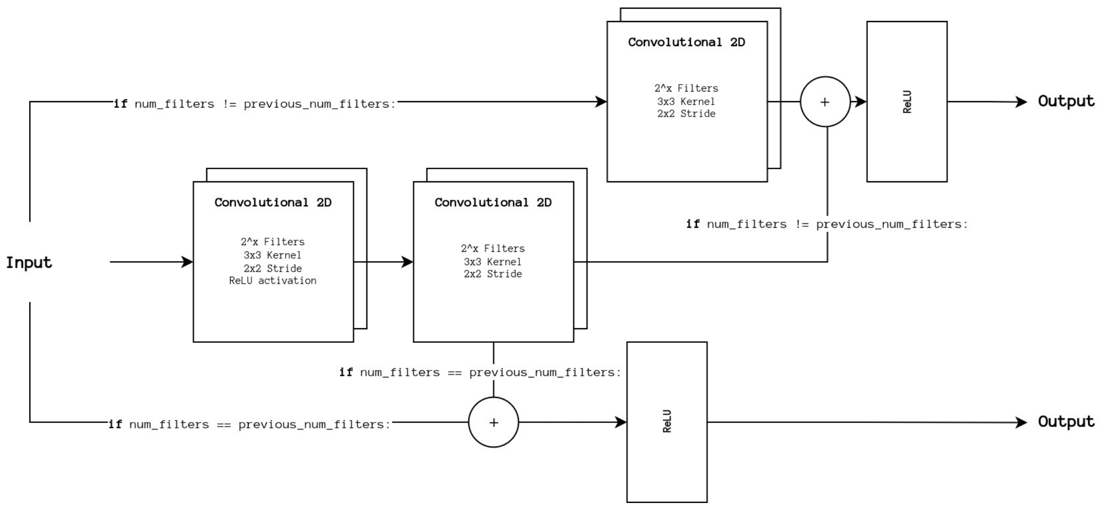


Figure 4.11: Diagram of the Residual 2D Block in the Sample CNN architecture.

it as a multi-class classification problem, meaning that only one class can be attributed to a sample. Moreover, by using smaller segments for training, the amount of training samples increases significantly, indirectly applying some Data Augmentation to the dataset, hopefully, increasing the accuracy at this level and extending it to the entire sample.

The implementation of this network is available in Won's repository[5]. The implementation was translated from PyTorch, as found in the repository, to Tensorflow, which was accurately done with precious help from the paper's author, Minz Won. In this repository, pre-trained weights for the various models are also available, which are discussed more in depth in Section 4.5.2, where these are experimented with.

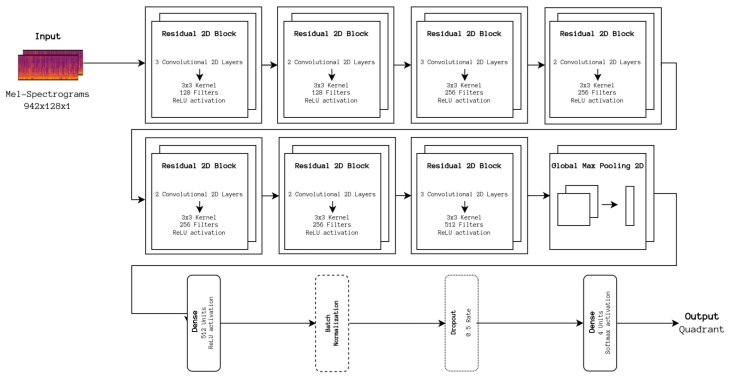


Figure 4.12: Diagram of the ShortChunk CNN architecture. The feature extraction portion comprehends the first Residual 2D block until the Global Max Pooling 2D Layer, the rest is part of the classifier portion.

Due to the smaller input size, the classification of the full sample is done through the average of the classification for the chunks of the sample. Experimenting with majority voting showed a decrease in performance with Legacy-MERGE.

A Residual 2D layer, depicted in Fig 4.11, for our purposes, is a sequence of two Convolutional 2D layers, with an additional Convolutional 2D layer for when the number of filters in the current Residual 2D layer is different from the previous. The input is passed through the two Conv 2D layers, which output is summed with the initial input before being passed to a ReLU activation layer, producing the final output. In case of the already mentioned

[5]https://github.com/minzwon/sota-music-tagging-models

Table 4.24: Confusion Matrix obtained for ShortChunk CNN.

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**989**	859	334	48	**52.16%**
Q2	312	**1810**	112	16	**64.23%**
Q3	134	356	**1342**	378	**52.71%**
Q4	99	276	1011	**854**	**46.90%**
				F1 Score Total	**54.00%**

difference in filters, the input is passed through the remaining Conv 2D layer before being summed with the output of the other layers. This architecture is depicted in Fig. 4.12. The use of the Residual 2D layer helps preserve information that would otherwise be lost during the convolutional operations [73].

The experiments using Legacy-MERGE Corrected accomplished at best a **54% F1-Score**, as can be seen in Table 4.24, which is considerably lower than the baseline Simple CNN. The explanation may lie in the considerably lower number of samples in comparison with the MGT-Jamendo dataset, from the 54.380 samples used in the reported experimentation to the 893 samples of Legacy-MERGE, meaning that the convolutional filters may have a hard time to adapt to less data.

The results in Table 4.25 show a similar performance for the Complete candidate of the New-MERGE dataset, but there is a significant decrease in performance for both balanced candidates. As other experiments have shown before, the balanced candidates seem to have worse performance in general.

Table 4.25: Results and comparisons obtained for New-MERGE dataset on ShortChunk CNN.

	F1 Score	Statistical Significance Test Against:		
		Baseline	Balanced	Balanced-Genre
Complete	**53.39%**	Not Significant	Significant ($1.1313 \text{x} 10^{-04}$)	Significant ($1.3298 \text{x} 10^{-04}$)
Balanced	**50.35%**	Significant ($2.5724 10^{-06}$)	//	Not Significant
Balanced-Genre	**49.60%**	Significant ($3.4229 \text{x} 10^{-06}$)	//	//

4.2.2 Sample CNN

End-to-end learning applies convolutional networks for extracting features directly from raw data, finding the best representation for classifying given samples. Previous studies were conducted with full length samples, failing to reach similar performance when using Mel-spectrograms instead, concluding that the model was not adequate enough for extracting the same level of features present in a pre-computed representation. Despite this, there was some promise due to the tested models being able to extract some relevant features, such as frequency decompositions.

With this idea in mind, Lee et al. [71] presents the Sample CNN architecture, which employs feature extraction at the sample-lavel by using small chunks of the sample, such as the precious architecture. This architecture is very deep, comprising 11 Convolutional

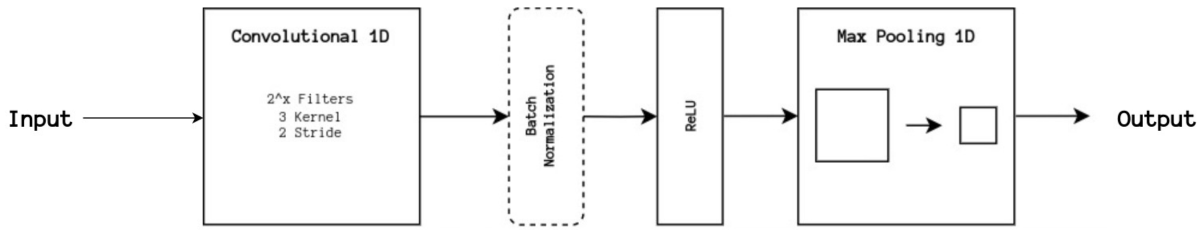


Figure 4.13: Diagram of the Conv 1D Block in the Sample CNN architecture.

Table 4.26: Confusion Matrix obtained for Sample CNN.

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**891**	889	372	78	**47.70%**
Q2	330	**1712**	192	16	**63.14%**
Q3	76	285	**1451**	398	**54.09%**
Q4	86	233	1082	**839**	**45.50%**
				F1 Score Total	**52.60%**

1D layers, a Dropout layer and a Dense layer for classification, as seen in Fig. 4.14. As with the previous network, Mel-spectrogram representations are also used as input to the network.

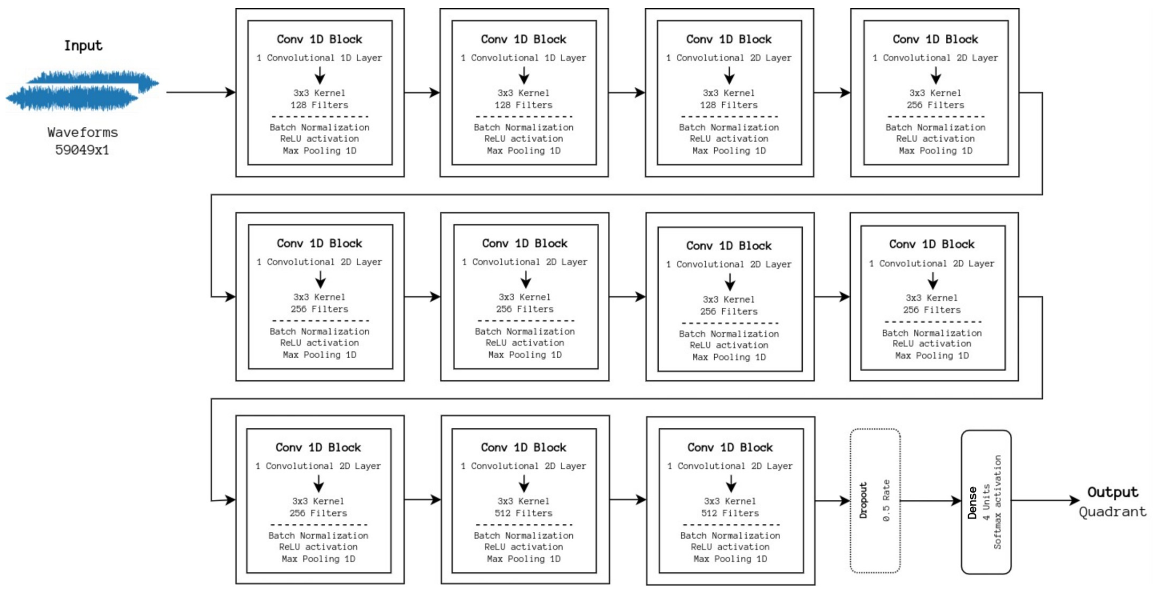


Figure 4.14: Diagram of the Sample CNN architecture. The feature extraction portion comprehends the first to the last Conv 1D Block, the rest is part of the classifier portion.

Due to the already large network and limited resources, changing the Convolutional 1D layers to Residual 1D was not considered.

The experiments using Legacy-MERGE Corrected accomplished at best a **52.60% F1-Score**, as shown in Table 4.26, still lower than the baseline Simple CNN. The same line of thought for the previous architecture stands: the very complex compositions of music are hard to effectively model with such little data.

As seen in the results displayed in Table 4.27 for the New-MERGE dataset, the same pattern of the Complete dataset candidates maintaining similar performance and the balanced candidates underperforming.

Table 4.27: Results and comparisons obtained for New-MERGE dataset on DNN with All 1714 Decorrelated Features.

	F1 Score	Statistical Significance Test Against:		
		Baseline	Balanced	Balanced-Genre
Complete	**52.00%**	Not Significant	Significant (3.6075x10^{-05})	Significant (0.8166)
Balanced	**47.51%**	Significant (2.295910^{-05})	//	Not Significant
Balanced-Genre	**47.44%**	Significant (3.3740x10^{-06})	//	//

4.3 Improvements to Baseline Network

In this section, an improvement to the baseline architecture from the foundation work is proposed, as well as experimenting with the CRNN architecture to study the impact of time-distributed features.

4.3.1 Improving Simple CNN with GRU

After trying some new architectures, it is obvious that our data does not play well with them. The baseline architecture continues to perform better, however there may be some room for improvement. For instance, the architecture in question only leverages sample-level features, directly classifying them with Dense layers. There is also some information that can be found by analyzing the resulting features from the CNN feature extractor, or frontend, as time-series data using a variant of an RNN. Due to recent findings that GRU units have very similar performance as LSTM units while using less computational resources, these were considered to improve our baseline.

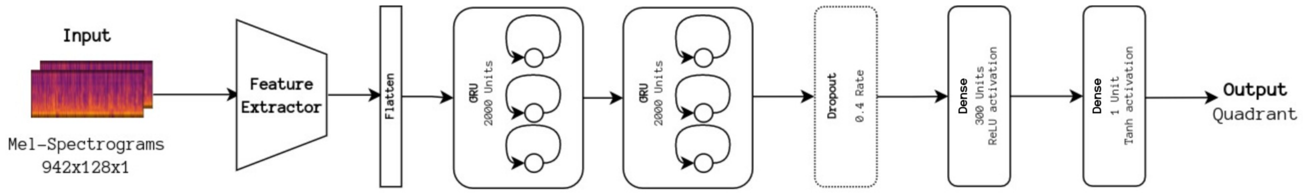


Figure 4.15: Diagram of the Simple CNN architecture with added GRU units in the classifier portion of the architecture.

The architecture is mostly unchanged from the already presented baseline, but two GRU units were introduced at the top of the classifier, or backend. The new proposed architecture can be seen in Fig. 4.15. Various numbers of units were tested, 2000 units in each layer provided the best results out of them.

Results for the Legacy-MERGE Corrected dataset showed no improvement, achieving at most a **60.07% F1-Score**, as seen in Table 4.28, which is a small dip in performance compared to the baseline results.

However, as it can be seen in Table 4.29, there is very significant improvement on the results for the New-MERGE Complete candidate, indicating that an increase in samples positively impacts the recurrent portion of the network, while the other candidates perform

Table 4.28: Confusion Matrix obtained for Improvement with GRU.

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**1307**	556	205	162	**58.45%**
Q2	265	**1872**	76	37	**77.06%**
Q3	311	118	**1116**	665	**49.99%**
Q4	292	71	887	**1210**	**54.79%**
				F1 Score Total	**60.07%**

Table 4.29: Results and comparisons obtained for New-MERGE dataset on Improvement with GRU.

	F1 Score	Statistical Significance Test Against:		
		Baseline	Balanced	Balanced-Genre
Complete	**61.99%**	Significant (0.0105)	Significant (0.0334)	Significant ($3.6075\text{x}10^{-05}$)
Balanced	**60.66%**	Not Significant	//	Significant (0.0178)
Balanced-Genre	**58.85%**	Not Significant	//	//

similarly. As shown by comparing between candidates, the number of samples seems to be the only driving factor for improving performance using this architecture.

4.3.2 CRNN

The idea of adding recurrent units to process the extracted features from the frontend came from the already discussed CRNN architecture from Choi et al. in Section 2. In the repository presented in Section 4.2.1, there is code provided for a variety of state of the art models, including the CRNN. As discussed in the article by Won et al. [25], the LSTM units are replaced by GRU units much for the same reason as mentioned in the previous expression.

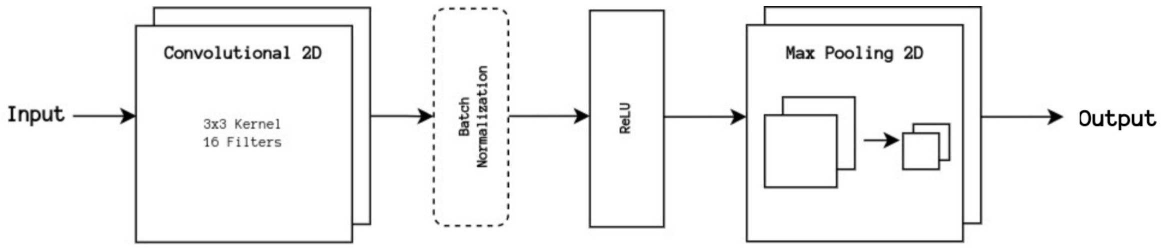


Figure 4.16: Diagram of the Conv 2D Block in the CRNN architecture.

The architecture, which can be seen in Fig. 4.17, is comprised of an initial Batch Normalization layer, followed by four Conv 2D Blocks, as the depicted in Fig. 4.16, the first with 64 filters and the rest with 128, two GRU units, a Dropout layer and a final Dense layer for classification. The model accepts a Mel-spectrogram of the full sample as input, outputting the prediction as one of the four quadrants.

Results for Legacy-MERGE show a **60.35% F1 Score**, displayed in Table 4.30, not improving, but maintaining the performance in a similar way to the previous experiment.

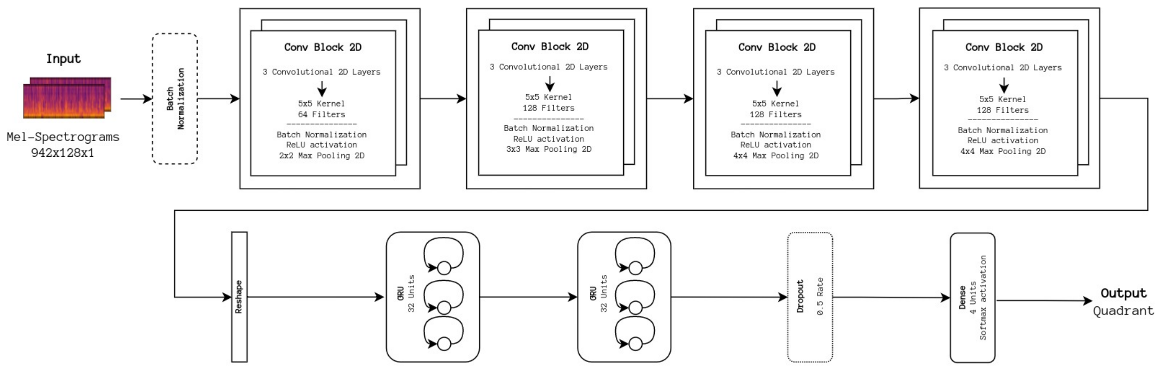


Figure 4.17: Diagram of the CRNN architecture. The feature extraction portion comprehends the Batch Normalization layer to the last Conv Block 2D, the rest is part of the classifier portion.

Table 4.30: Confusion Matrix obtained for CRNN.

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**1372**	402	253	203	**60.05%**
Q2	355	**1742**	111	42	**76.41%**
Q3	241	98	**1223**	648	**52.07%**
Q4	264	46	728	**1202**	**52.89%**
				F1 Score Total	**60.35%**

Table 4.31: Results and comparisons obtained for New-MERGE dataset on CRNN.

	F1 Score	Statistical Significance Test Against:		
		Baseline	Balanced	Balanced-Genre
Complete	**63.33%**	Significant (0.0031)	Not Significant	Not Significant
Balanced	**62.50%**	Significant (0.0264)	//	Not Significant
Balanced-Genre	**62.54%**	Significant (0.0262)	//	//

However, we obtained very significant increases in performance for all candidates of the New-MERGE dataset as shown in Table 4.31. These positive results indicate that extracting time-distributed features is a very promising direction for improving future architectures.

4.4 Data Augmentation

In this section, we experiment with different audio augmentations, using both classical audio augmentation techniques and a technique for generating new artificial samples by exploiting the a learned latent space.

4.4.1 Classical Audio Augmentations

Further expanding on the work from Sá, other audio augmentation techniques were explored beyond the already four experimented with before. To do this, the available augmentations on the audiomentations library[6] were explored, from which the five more promising ones were picked to be experimented with. Classical audio augmentations, although tested with other subfields of MIR, such as Music Genre Recognition [74], only a few are found in MER literature. This is mostly due to the difficulty in guaranteeing that an augmented sample, generated from a real sample by applying one or a composition of techniques, maintains the same emotion, or the same quadrant for our case. There is a need to further experiment with traditional techniques to find possible better ways to generate new samples.

Table 4.32: Confusion Matrix obtained for Simple CNN with Time-Frequency Masking Audio Augmentation.

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**1366**	464	168	232	**60.70%**
Q2	324	**1818**	72	36	**77.63%**
Q3	266	90	**1121**	733	**52.22%**
Q4	244	41	624	**1331**	**57.56%**
				F1 Score Total	**62.03%**

Table 4.33: Confusion Matrix obtained for Simple CNN with Tanh Distortion Audio Augmentation.

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**1362**	436	203	229	**60.77%**
Q2	291	**1805**	90	64	**77.40%**
Q3	259	84	**1028**	839	**48.25%**
Q4	264	71	572	**1333**	**55.96%**
				F1 Score Total	**60.59%**

Table 4.34: Confusion Matrix obtained for Simple CNN with Seven-band Parametric Equalization Audio Augmentation.

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**1410**	509	154	157	**62.17%**
Q2	255	**1893**	69	33	**77.53%**
Q3	277	144	**1154**	635	**53.28%**
Q4	326	70	636	**1208**	**55.50%**
				F1 Score Total	**62.12%**

The techniques from the audiomentations library, which were already explained in Section 4.4, experimented with were Time-Frequency Masking (Table 4.32), Random Gain (Table 4.35) and Seven-band Parametric Equalization (Table 4.34), which significantly outperformed the baseline results, achieving a 62.03%, 62.24% and 62.12% F1-Score respectively, with 0.0451, 0.0340 and 0.0230 p-values. However, Tanh Distortion (Table 4.33) and Background Noise (Table 4.36) all underperformed, achieving a 60.59% and 60.84% F1-Score

[6]https://github.com/iver56/audiomentations

respectively. Overall, the augmentations that performed better despite changing the sample in some ways, the actual content of the songs persist, along with the emotional content in them. This may explain the bad performance of the other augmentations, since Tanh Distortion may introduce some information to samples that is encountered very frequently in rock and metal music, which are genres mostly associated with the Q2 quadrant, leading to a bias towards it, while Background Noise may make the context of songs become more jarring by introducing dissonant noises.

Table 4.35: Confusion Matrix obtained for Simple CNN with Random Gain Audio Augmentation.

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**1582**	422	95	131	**63.51%**
Q2	331	**1833**	62	24	**78.68%**
Q3	419	87	**1046**	658	**51.21%**
Q4	382	55	599	**1204**	**55.57%**
				F1 Score Total	**62.24%**

Table 4.36: Confusion Matrix obtained for Simple CNN with Background Noise Audio Augmentation.

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**1334**	480	161	255	**60.71%**
Q2	261	**1838**	74	77	**77.71%**
Q3	251	80	**1000**	879	**48.75%**
Q4	260	63	548	**1369**	**56.20%**
				F1 Score Total	**60.84%**

Due to resource limitation and the amount of methodologies to still be developed, the approach seen in the foundation work, were all audio augmentations were applied simultaneously to obtain a better performance than using these individually, was not able to be tested. It was decided to further experiment with the ones that showed visible improvement. With this said, the results obtained for the New-MERGE dataset are shown in Tables 4.37, 4.38 and 4.39 for the best performing augmentation techniques.

Table 4.37: Results and comparisons obtained for New-MERGE dataset on Simple CNN with Time-Frequency Mask Audio Augmentation.

	F1 Score	Statistical Significance Test Against:		
		Baseline	Balanced	Balanced-Genre
Complete	**61.82%**	Not Significant	Significant (0.0251)	Not Significant
Balanced	**60.58%**	Significant (0.0264)	//	Not Significant
Balanced-Genre	**61.39%**	Not Significant	//	//

The great majority of the results show similar performance between the New-MERGE candidates and the Legacy-MERGE dataset, with an exception of the Balanced candidate when Time-Frequency Masking is applied. Since it has been found that there are some songs that are exclusive to each of the candidates and taking into account the randomness

Table 4.38: Results and comparisons obtained for New-MERGE dataset on Simple CNN with Seven-band Parametric Equalization Audio Augmentation.

	F1 Score	Statistical Significance Test Against:		
		Baseline	Balanced	Balanced-Genre
Complete	**61.73%**	Not Significant	Not Significant	Not Significant
Balanced	**61.31%**	Not Significant	//	Not Significant
Balanced-Genre	**61.01%**	Not Significant	//	//

Table 4.39: Results and comparisons obtained for New-MERGE dataset on Simple CNN with Random Gain Audio Augmentation.

	F1 Score	Statistical Significance Test Against:		
		Baseline	Balanced	Balanced-Genre
Complete	**62.08%**	Not Significant	Not Significant	Not Significant
Balanced	**61.17%**	Not Significant	//	Not Significant
Balanced-Genre	**61.36%**	Not Significant	//	//

of the factor, more or less accentuated, and location in the sample where the augmentations are applied, may explain this lower value compared with the Balanced-Genre dataset. This can only be verified by using a different random seed, changing how the augmentations are applied to each sample, and making a comparison with the obtained results, but as it stands, there does not to seem any considerable changes in this approach.

4.4.2 Deep SMOTE

As discussed in the embeddings section 4.6.2, deep audio embeddings have recently seen success in the MIR field. Using a higher level representation not only makes it possible to use more traditional ML algorithms, such as SVM as an example, but also other techniques, leading us to SMOTE.

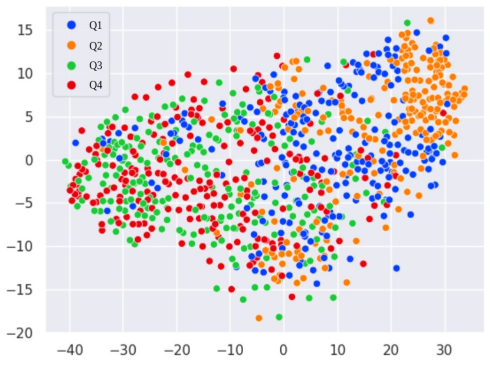
Synthetic Minority Oversampling Technique, or SMOTE, is a widely used Data Augmentation technique, consisting, very briefly, of finding clusters of minority data samples and generating new samples within these clusters. The original implementation consists of using the K-Nearest Neighbor classifier to find the above clusters, adjusting k as needed, and randomly sampling new data points through interpolation of already existing points in the same cluster.

From the inception of SMOTE [75] until the moment of writing, there have been a myriad of variants introduced in order to address different problems with the original approach, which adjust the methodology of achieving both above mentioned points.

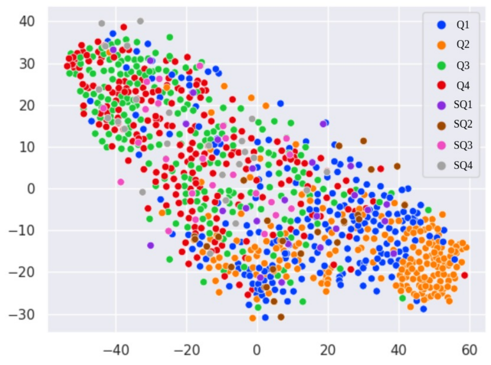
Due to the nature of SMOTE, very high-dimensional datasets tend to produce low quality or almost copies of the already existing points, which leads to redundant or noisy data

[76]. With this in mind, the idea of reducing the dimensionality of our audio samples using the autoencoder previously trained for the GAN experiment was experimented with, which coincidentally, had already been proposed by Dablain et al. [77]. The idea is the same: reduce the dimensionality using an autoencoder, apply SMOTE to the embedded data and finally get the images, in our case Mel-spectrograms, from the resulting SMOTE'd embeddings through the same autoencoder.

As to understand the distribution of the samples in the latent space of the embeddings, tSNE, a technique that projects high-dimensional samples into a lower dimensional plane for visualization purposes, was used and can be seen in Fig. 4.18.



(a) Distribution of the samples as calculated from their embeddings.



(b) Distribution of the samples as calculated from their embeddings, including generated samples.

Figure 4.18: Visualization of the samples' distribution using tSNE, without, left, and with, right, generated samples.

The SMOTE'ing process was done using the smote_variants package [78], introduced in the work by Kovacs [79], where a total of 85 variants of SMOTE were tested against each other using 104 imbalance datasets. The library also supports oversampling with multi-class problems, not found in other libraries which only support binary problems, as the original implementation. Something that was not referred to until now, is that SMOTE was developed in order to balance unbalanced datasets, which was not exactly the case for our already balanced Legacy-MERGE corrected dataset. Despite this, the library offers the option to oversample any class up to twice the proportion between the class in question and the other classes. This feature was taken advantage of to generate new samples from our already balanced dataset.

For our purposes, some of the most mentioned SMOTE variants across the reviewed literature were explored: SMOTE, Borderline-SMOTE and Adasyn. Some preliminary testing found that due to the distribution in the latent space, the performance difference between each of the tested variants was very minor and it was decided to proceed with Borderline-SMOTE due to its theoretical basis, consisting on the same KNN clustering as the original, but instead of randomly sampling from within the clusters, possible new samples which are closer to the borderline with other classes have priority on being chosen. The number of new generated samples was also tested, finding that the 25 new samples for each class produced the best performance, since more than these always produced worse results. The best F1-Score found for 25 samples generated using Borderline-SMOTE ended up at **60.70%**, shown in Table 4.40, which is below the baseline results for the Legacy-MERGE Corrected dataset.

Table 4.40: Confusion Matrix obtained for Simple CNN with Deep SMOTE generated samples.

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**1138**	715	224	153	**55.34%**
Q2	217	**1930**	70	33	**75.73%**
Q3	231	124	**1231**	624	**55.03%**
Q4	215	101	689	**1235**	**56.72%**
				F1 Score Total	**60.70%**

The low results may be explained by some factors. The dimension of the latent space is one of the most probable reasons for the poor performance. The paper by Dablain et al. presents an autoencoder that reduces the dimensionality to 600, whereas the lowest our data is capable of going before being unable to reconstruct the samples is 60416, a lot higher than the previously mentioned dimensionality. Using SMOTE on such high dimensions as already mentioned, tends to produce lower results and it could be seen here and further increase the necessity for more data even for Data Augmentation approaches. The choice of the SMOTE variant may also contribute to this, since according to Kovacs, despite his findings, there is not an objectively better variant and may change depending on the problem at hand. Further investigating other variants may also lead to better performance.

Results obtained for New-Merge shown in Table 4.41 indicate that contrary to what was seen for the GAN generated sample, the increase in samples did not impact the SMOTE generated samples.

Table 4.41: Results and comparisons obtained for New-MERGE dataset on Simple CNN with Deep SMOTE generated samples.

	F1 Score	Statistical Significance Test Against:		
		Baseline	Balanced	Balanced-Genre
Complete	**61.47%**	Not Significant	Not Significant	Not Significant
Balanced	**61.52%**	Not Significant	//	Not Significant
Balanced-Genre	**60.48%**	Not Significant	//	//

4.5 Transfer Learning

As already discussed in Chapter 2, Transfer Learning has recently been applied in MER due to the mostly small publicly available datasets. Beyond this, Transfer Learning can also be used to transfer domain knowledge from one field to, hopefully, aid models to understand underlying patterns in the data that datasets from the field are not prepared for.

Table 4.42: Confusion Matrix obtained for Transfer Learning with Artists CNN.

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**854**	960	363	53	**45.85%**
Q2	390	**1606**	205	49	**58.87%**
Q3	93	367	**1262**	488	**50.43%**
Q4	105	267	949	**919**	**48.25%**
				F1 Score Total	**50.85%**

4.5.1 Artists CNN

Park et al. [80] present two different methodologies. First, a simple model using CNN as the frontend, similar to our baseline model. Second, a siamese neural network, which consists of at least two twin networks, denominated this way due to having the same exact architecture and sharing weights and other parameters, trained using a sample and positive and negative items, which should make the network prioritize the patterns that are similar between the sample and positive items, doing the opposite between the sample and negative items. Both accept Mel-spectrogram as inputs, output to between 1000 to 10000 artist label depending on the configuration, and was trained using the MSD dataset. This last methodology was considered to be experimented with, but the reported results comparison shows that most of the time the simple methodology performs the same or better than the siamese one, which led us to experiment with the simpler model.

The Transfer Learning model was built using the architecture from the simple methodology, discarding the last layer, substituting it with a final classification layer for our four quadrants and freezing the remaining layers, which can be seen in Fig. 4.19. A dropout layer and more Dense layers before the classification layer were tested with, degrading the performance with the parameters tested with. This may be due to the already small feature vector produced by the frozen layers, having dimensionality of 256.

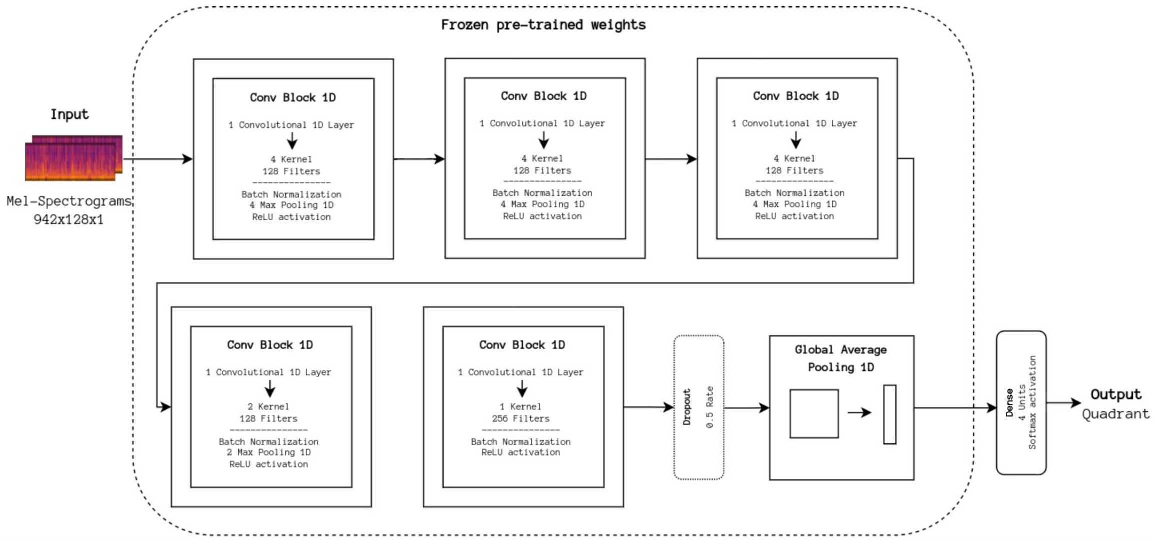


Figure 4.19: Diagram of the Artists CNN architecture. The feature extraction portion comprehends the first Conv 1D Block to the Global Max Pooling, the rest is part of the classifier portion.

Experiments using the Legacy-MERGE corrected dataset achieved at best a **50.85% F1-Score**, lower than the baseline results. Transfer learning has not provided any improvements despite different approaches taken, and as it can be seen in Table 4.42, the same is observed when using the New-MERGE dataset as can be seen in Table 4.43.

Table 4.43: Results and comparisons obtained for New-MERGE dataset on Transfer Learning with Artists CNN

	F1 Score	Statistical Significance Test Against:		
		Baseline	Balanced	Balanced-Genre
Complete	**50.27%**	Not Significant	Not Significant	Not Significant
Balanced	**50.63%**	Not Significant	//	Not Significant
Balanced-Genre	**50.22%**	Not Significant	//	//

4.5.2 CRNN Pre-trained on MTAT, Jamendo, MSD

As already mentioned in the previous experiment, Transfer Learning may be done to transfer the knowledge from a domain where data is abundant to a related domain with less available data. At some point, the article by Choi et al. [81] was considered to be used as part of a Transfer Learning experiment. The model used a CNN to extract features, something very familiar at this point, however, the output of each layer except the last one is passed through an Average Pooling layer, all outputs being concatenated at the end, including the output from the last layer, to form a feature vector. This feature vector is then classified using a variety of classifiers and regressors in order to find the best results. The model accepts The idea behind using the output of the intermediate layers of the CNN is to capture lower-level features that may ultimately be lost in the final layer, which features are thought to be more high-level. Despite the implementation being available here, using the provided weights appears to be impossible, since there is not an accurate way to replicate the environment where these were used. Contacting the author of the article and repository owner, Keunwoo Choi, pointed us to the provided models in Won's repository mentioned in Section 4.2.1 as more viable candidates for our experiment.

As way of comparing performance between using bigger datasets for pre-training a network or only using our data, the CRNN model, which was already discussed in Section 2, and tested in 4.3.2 using a translated Tensorflow implementation and our own data. The architecture of the model is the same as already discussed, but the original implementation in PyTorch is used to be able to use the pre-trained weightgs.

Upon building the model, the weights for one of the tested datasets, MTAT, Jamendo, and MSD, are loaded, and the layers are frozen, replacing the last one with a classification layer that outputs one of the four quadrants. Despite the models being trained with Mel-spectrograms with 96 Mel bins, after experimentation, it was found that 128 Mel bins, used for the remaining experiments, still provided the best results. Another important point is the optimizer used for training this Transfer Learning model, which is not static. This method, discussed in the article mentioned back in Section 4.2.1, consists of training the model for 200 epochs and changing the current optimizer or/and learning rate depending on the current epoch. This is done to more rapidly converge to the best values for the weights of the model, while reducing the number of epochs necessary for a static optimizer to reach similar performance.

Table 4.44: Confusion Matrix obtained for Transfer Learning with CRNN using MTAT

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**267**	349	589	1019	**22.38%**
Q2	151	**1045**	342	697	**74.28%**
Q3	127	123	**813**	1147	**42.16%**
Q4	154	79	642	**1356**	**48.95%**
				F1 Score Total	**46.40%**

Table 4.45: Confusion Matrix obtained for Transfer Learning with CRNN using MTG-Jamendo

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**531**	735	381	568	**28.86%**
Q2	164	**1833**	94	162	**69.71%**
Q3	374	222	**465**	1144	**25.29%**
Q4	192	190	364	**1511**	**53.09%**
				F1 Score Total	**43.73%**

The best results were achieved for a batch size of 16 and using the pre-trained weights for the MTAT dataset (Table 4.44), reaching a 46.87% F1-Score with the Legacy-MERGE corrected dataset, lower than the baseline and the results obtained for the CRNN trained with our data, while using the Jamendo (Table 4.45) and MSD (Table 4.46) reached 43.73% and 41.29% respectively. Interestingly, we would expect that a network trained with larger datasets would produce better results, but has it can be seen, this is not the case. Of course this may not be the only factor, as the lower quality of MSD as already been discussed in 2.2, seemingly confirming it, and the same may be true for the Jamendo dataset. Despite this, there is a considerable decrease in performance in comparison with training the network only using our dataset, which evokes the interest in further understanding which dataset is indeed the less accurate in terms of annotations, since this cannot be extrapolated with the current data. Due to this clear better performance seen with the MTAT pre-trained weights, these were the only ones considered when experimenting with New-MERGE.

Table 4.46: Confusion Matrix obtained for Transfer Learning with CRNN using MSD

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**1061**	541	308	304	**44.33%**
Q2	221	**1717**	121	198	**60.82%**
Q3	576	647	**388**	598	**21.83%**
Q4	641	478	258	**873**	**40.14%**
				F1 Score Total	**41.29%**

The results for New-Merge, displayed in Table 4.47, show an improvement across all candidates when compared with Legacy-MERGE.

4.6 Multiple Spectral Representations

Using multiple representations for classification came from the idea that different spectral representations have different information that can be extracted. By having a network branch to extract features for each representation, that branch adapts to the representation,

Table 4.47: Results and comparisons obtained for New-MERGE dataset on Transfer Learning with CRNN using MTAT

	F1 Score	Statistical Significance Test Against:		
		Baseline	Balanced	Balanced-Genre
Complete	**49.81%**	Significant (4.434510^{-06})	Not Significant	Not Significant
Balanced	**49.46%**	Significant (3.580710^{-05})	//	Not Significant
Balanced-Genre	**49.35%**	Significant (3.580710^{-05})	//	//

which would in theory, extract relevant features from each one [40] [82].

A Short Time Fourier Transform, referred to STFT in the literature, is a transformation applied to a raw signal to extract the time-frequency information present in it, with the intent of plotting it as a spectrogram of the signal. This is done by segmenting the signal in small segments, or windows, and applying a Fourier Transform over each individual window. These windows normally have a small overlap to ensure that no information is lost in the calculations. STFT is also used as an intermediate step for other spectral representations, such as the Mel-spectrogram.

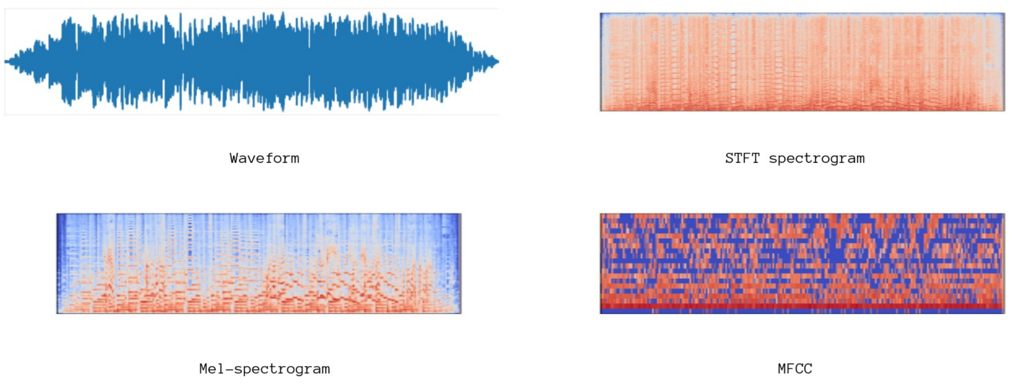


Figure 4.20: Waveform, STFT, Mel-spectrogram and MFCC representations of the same 30 second song clip. For each representation, excluding the waveform, the previous representation is a necessary step for obtaining it, meaning that for obtaining the MFCC representation, the Mel-spectrogram is needed, which is obtained from the STFT spectrogram, and so on.

The Mel Frequency Cepstral Coefficient, or as more commonly known, MFCC, is widely used in the field of Automatic Speech Recognition and Speech Emotion Recognition [83], since it accurately represents the structure of a human's vocal tract, but more recently as seen use in various subfields of MIR, due to studies finding that it represents the timbre of a audio signal quite well. The process starts by getting an STFT spectrogram of the signal, converting it to the Mel scale, and finally applying the same procedure of STFT over the spectrogram, producing the STFT representation. The resulting representation is a cepstrogram, the name given to the spectrogram of the log of the STFT. All of the mentioned representations can be seen in Fig. 4.20.

4.6.1 Segment-level Representations

Bathigama et al. [84] presents an approach which uses the idea in the beginning of this subsection. The architecture, as seen in Fig. 4.22, takes as input small samples, such as 4.2, these being longer, each one 5 seconds, and is trained as a regressor using the values in the AV plane for each sample. The feature extraction portion is composed by three branches, one for each representation used as input: STFT, Mel-spectrogram, and MFCC.

The sample rate used for getting the representations is 22kHz, above the 16kHz used as baseline for the experiments before. In theory, there would be more information per time window when using a higher sample rate, but both are used in the literature depending on the architecture in question.

The branches are very similar, being composed by a certain number of Conv Blocks 2D, depending on the representation, an Adaptive Average Pooling layer and a Flattening layer before being passed to a Concatenation layer. The backend, a regressor in this case, is composed of a Dense Block with Dropout, a Dense Block without Dropout, which outputs are passed to two branches, one for arousal and another for valence. Besides F1-Score, RMSE is also a relevant metric to assess how well each branch can correctly predict samples.

It is also important to note that a custom Early Stopping strategy was used for this experiment. Until this point, most of the criteria necessary for ensuring good performance of the models, was to limit the accuracy of the training to 0.9, mitigating the possibility of the model overfitting. For this experiment and the next, the training of the models was stopped after 10 epochs without any improvements, keeping the weights found as having the best performance until that point. This drastically reduced the necessary time to train these very complex models and mitigates the not surprising unstability of the training process, due to the very complex nature of them.

The experiments using Legacy-MERGE Corrected produced a 43.06% F1-Score, a lot lower than the baseline. Interestingly, a 0.2114 RMSE was obtained for Arousal, which is better than the reported results of 0.2301 RMSE, but for valence, the obtained result was a 0.4318 RMSE, which is considerably worse than the reported 0.1938 RMSE.

Having used a different dataset for the experimentation, it was expected that the results would not be the same, but the difference in valence is higher than would be expected, since it is lower than arousal in the reported results. Using 5-fold Cross Validation could explain some differences, but the lack of details for generating the spectral representations may be the factor with more impact. Further tests with the parameters for the representations should be made.

Results for the New-MERGE dataset, show in Table 4.49, an increase for the Complete dataset, while maintaining the same performance for the Balanced dataset. This shows that these models need a lot more data for considerable improvements to be made, as can be seen by the underperforming Balanced-Genre candidate.

4.6.2 Sample-level Representations

Interpolating the idea from [84], the baseline frontend architecture was used as the foundation for STFT and MFCC feature extractors. The architecture, as seen in 4.22, is very similar to the previous model, only varying the number of Conv Block 2D. This experiment was done to assess how well the architecture from the previous experiment suited our data.

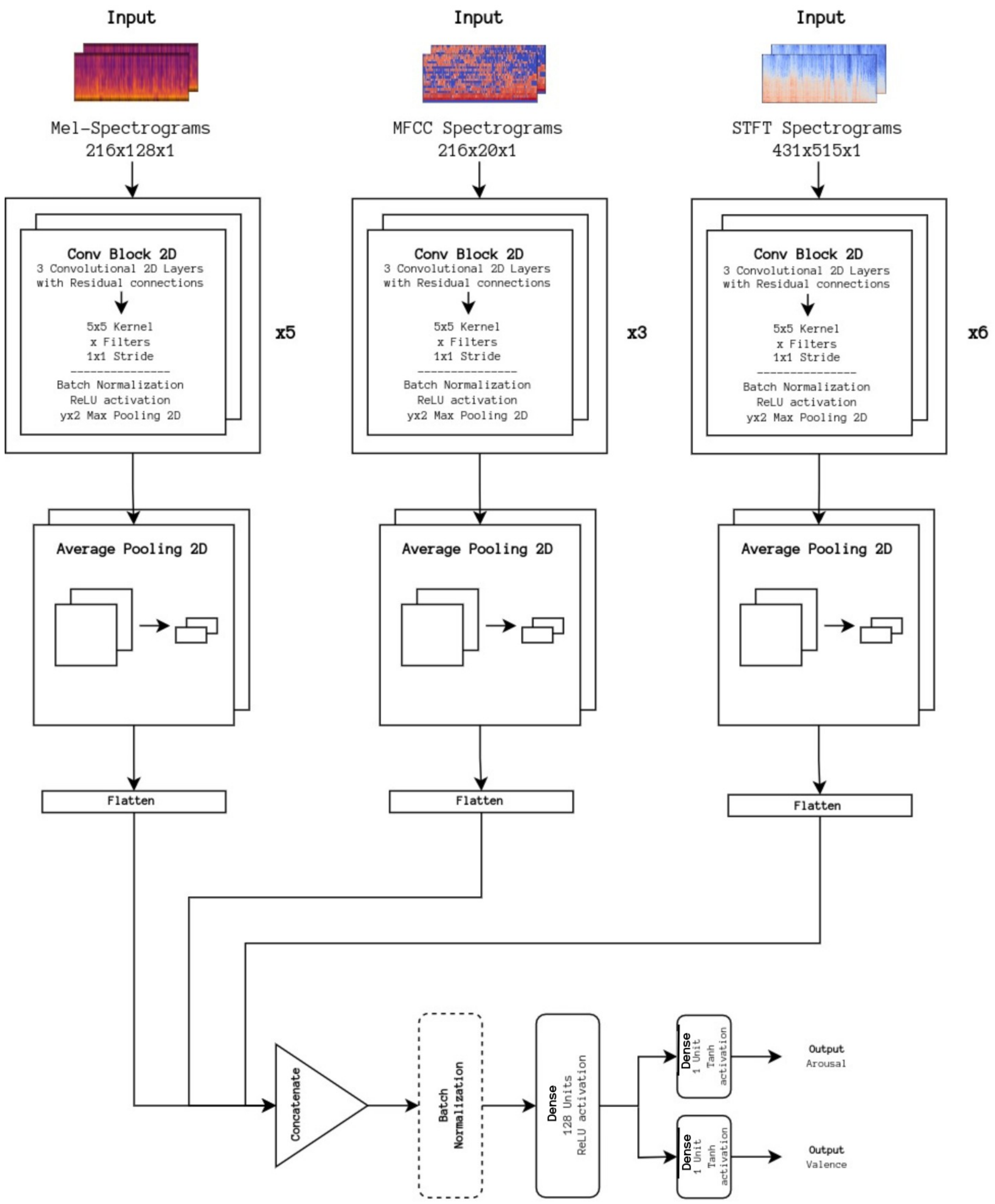


Figure 4.21: Diagram of the Multiple Representations Segment-level CNN architecture. The feature extraction portion comprehends the various Conv Block 2D used in each representation, which after concatenated is feed to the classifier portion.

The best results obtained was a 49.80% F1-Score, as seen in Table 4.50, higher than the segmented approach, but still lower than the baseline. Arousal reached 0.2596 RMSE and Valence 0.4799 RMSE, higher than the segmented approach, despite a better F1-Score.

A possible explanation for a worse F1-Score but better RMSE for both arousal and valence, is the method for classifying the full sample emotion. As for the first experiment, the mean

Table 4.48: Confusion Matrix obtained for Segment-level Multiple Representations.

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**1069**	812	281	68	**44.25%**
Q2	402	**1643**	182	23	**58.03%**
Q3	512	584	**830**	284	**36.87%**
Q4	460	421	791	**568**	**32.85%**
				F1 Score Total	**42.99%**

Table 4.49: Results and comparisons obtained for New-MERGE dataset on Segment-level Multiple Representations.

	F1 Score	Statistical Significance Test Against:		
		Baseline	Balanced	Balanced-Genre
Complete	**49.81%**	Not Significant	Significant (0.0029)	Significant (5.113810^{-16})
Balanced	**49.46%**	Not Significant	//	Significant (7.127310^{-14})
Balanced-Genre	**49.35%**	Significant (1.554510^{-07})	//	//

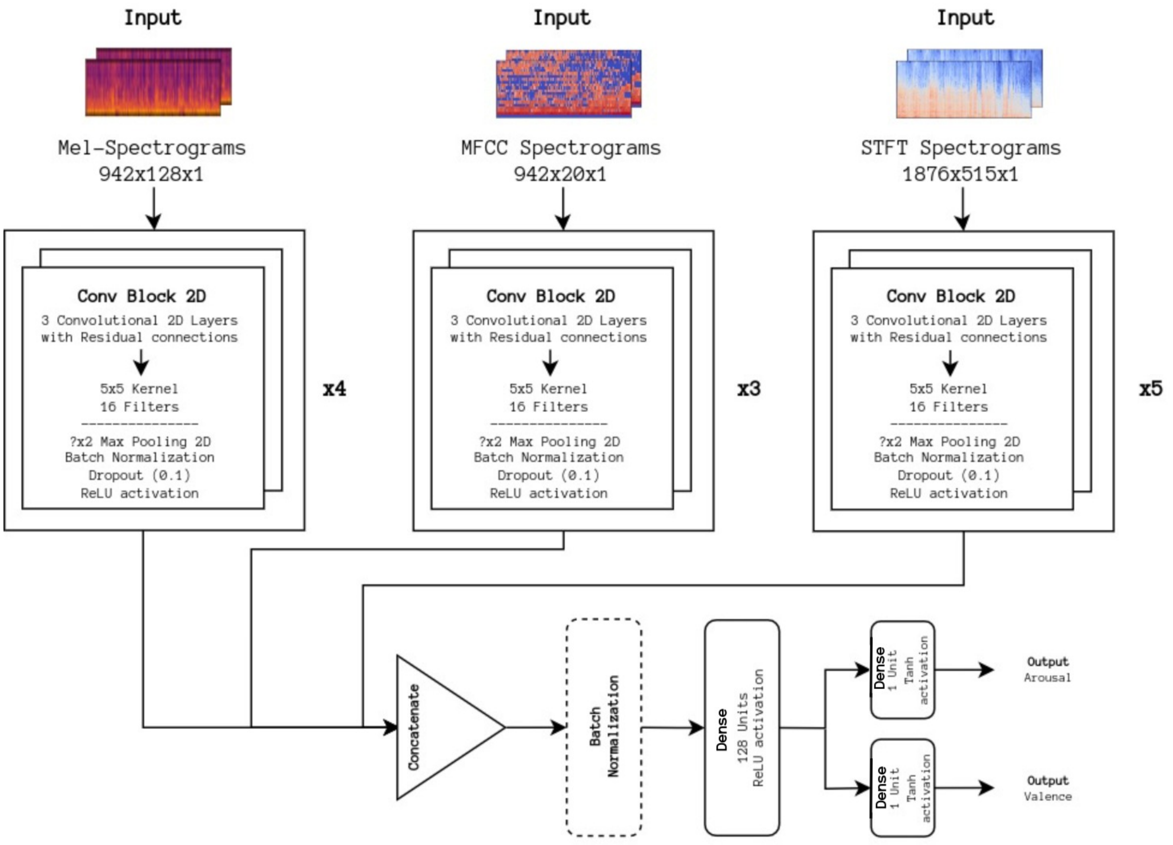


Figure 4.22: Diagram of the Multiple Representations Sample-level CNN architecture. The feature extraction portion comprehends the various Conv Block 2D used in each representation, which after concatenated is feed to the classifier portion.

of all segments' predictions of a sample determines the emotion for the whole sample, due to producing better results for that experiment, although it is not what is commonly done in the literature, using the mode over the mean of the segments. It is possible that using the mode instead may lead to better results, but it was not possible to assess. As for the experiment with the full sample, a bigger batch size may lead to better performance, but a 32 batch size was the most that could be tested with the resources available.

Table 4.50: Confusion Matrix obtained for Sample-level Multiple Representations.

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**865**	617	369	379	**46.04%**
Q2	220	**1444**	507	79	**61.43%**
Q3	170	147	**1123**	770	**43.13%**
Q4	145	83	872	**1140**	**48.60%**
				F1 Score Total	**49.80%**

There was a surprisingly good result obtained for the Balanced-Genre candidate of the New-MERGE dataset, a **74.23% F1 Score**, as seen in Table 4.51, which is on par with the state-of-the-art SVM. Although the results for the other candidates make this result suspicious, there was nothing found that would erroneously produce this high score. This seems to indicate that multiple spectral representations are indeed a very promising direction to further mitigate the small size of our datasets, as the extra information leveraged from other representations is shown to be beneficial.

Table 4.51: Results and comparisons obtained for New-MERGE dataset on Sample-level Multiple Representations.

	F1 Score	Statistical Significance Test Against:		
		Baseline	Balanced	Balanced-Genre
Complete	**31.44%**	Significant (2.678110^{-27})	Significant (6.270910^{-08})	Significant (2.719710^{-34})
Balanced	**28.68%**	Significant (1.037210^{-58})	//	Significant (4.067410^{-128})
Balanced-Genre	**74.23%**	Significant (5.882010^{-61})	//	//

4.7 Deep Audio Embeddings

In this section, deep audio embeddings approaches are presented using pre-trained deep audio embedding models and utilizing a data-driven approach using an autoencoder trained on our data.

4.7.1 OpenL3

Recently, deep audio embeddings have shown great performance such as in SER, where Cramer et al. [85] reached state-of-the-art results when compared against other popular deep embeddings architectures using a modified version of the Look, Listen, Learn deep embedding approach. Taking in account these promising results, Koh et al. [86] further compares the performance of this deep audio embedding methodology against VGGish, a

Table 4.52: Confusion Matrix obtained for Random Forest with OpenL3 Embeddings

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**1128**	449	313	340	**47.31%**
Q2	477	**1564**	115	94	**68.24%**
Q3	451	207	**822**	730	**39.56%**
Q4	459	109	659	**1013**	**45.59%**
				F1 Score Total	55.70%

popular deep embedding approach experimented by Somonyan et al. [87] using the popular VGGNet model as the foundation, using a variety of classifiers.

The experiments were conducted using a variety of datasets, including our own 4QAED dataset, using the OpenL3 library as the L3 implementation for obtaining the embeddings.

The OpenL3 library [7] is a collection of embedding models for a variety of media, including music. The authors state that the purpose of this library is to provide an implementation of the Look, Listen, Learn (L3) paper by Arandjelovic et al. [88] and to improve on some of the flaws in the original method.

The model used was trained on the music subset of the AudioSet dataset [89], consisting on more than 1M 10 sec excerpts from Youtube videos following a very thorough 632 audio classes onthology, encompassing a myriad of audio-related tasks.

In the paper by Koh et al., various configurations of the embedding model for music in this library are experimented with and evaluated with a variety of classifiers (SVM, RF, DNN,...) using the Legacy-MERGE dataset.

The reported results show that the best classifier for this approach is Random Forest with a 72% F1-Score. There is some confusion as to which splits for the data were used, more specifically, there is mention of a 80%/10%/10% split for train, test and validation sets respectively, but also a mention of using cross-validation with 20 repetitions. For consistency, cross-validation with 10 splits and 10 repetitions was used as for other experiments presented. Attempts to contact the author asking clarification on this point, as well as details not mentioned regarding the parameters used for training the RF classifier, but there was no answer until the moment of writing.

The obtained results show a huge decrease in performance, only being able to reach a **55.70% F1 Score** using the provided details, as shown in Table 4.52. Unfortunelly, the problem with the replicability of published work is general across all science fields, being dubbed the replication crisis [90], which most of the time entails the lack of enough detail to properly replicate the presented results. Since replicability of previous work is necessary for consistently improving on any science field, we have ensured to share the necessary details for each conducted experiment, hoping these may be of use in future work, in or out of the MERGE team.

As for the New-MERGE dataset, there were some significant improvements, as can be seen in Table 4.53. This is most probably due to the increase in samples, since the performance of each candidate slightly decreases in proportion with the quantity of samples available.

[7]https://github.com/marl/openl3

Table 4.53: Results and comparisons obtained for New-MERGE dataset on Sample-level Multiple Representations.

	F1 Score	Statistical Significance Test Against:		
		Baseline	Balanced	Balanced-Genre
Complete	**53.62%**	Significant (3.7057×10^{-07})	Not Significant	Not Significant
Balanced	**52.95%**	Significant (2.3283×10^{-05})	//	Not Significant
Balanced-Genre	**52.85%**	Significant (5.4467×10^{-05})	//	//

4.7.2 Autoencoder Embeddings

Since an autoencoder was trained for the purpose of experimenting with Data Augmentation, see Section 4.4.2, classification of the resulting embeddings on that latent space may be more accurate due to the data-driven approach taken. In order to test the validity of this approach, the same classifier from the previous experiment was tested with these data points, retaining the same parameters for a direct comparison.

Table 4.54: Confusion Matrix obtained for Random Forest with Autoencoder Embeddings

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**1351**	344	318	217	**55.06%**
Q2	627	**1404**	133	86	**68.19%**
Q3	340	67	**1049**	754	**47.39%**
Q4	337	38	695	**1170**	**52.17%**
				F1 Score Total	**50.85%**

Results indicate that the OpenL3 embeddings still perform better, as shown in Table 4.54 with a lower **50.85% F1 Score**. Despite the lower score, the obtained scores for the New-MERGE dataset, shown in Table 4.55, corroborates what was already found for the Deep SMOTE approach: more samples increases the accuracy of the learned latent space. As the dataset increases in the future, we may expect even further improvements to this methodology, has more accurate latent space representations will be attainable.

Table 4.55: Results and comparisons obtained for New-MERGE dataset on Sample-level Multiple Representations.

	F1 Score	Statistical Significance Test Against:		
		Baseline	Balanced	Balanced-Genre
Complete	**53.56%**	Significant (3.7057×10^{-07})	Not Significant	Not Significant
Balanced	**52.97%**	Significant (2.3283×10^{-05})	//	Not Significant
Balanced-Genre	**53.69%**	Significant (5.4467×10^{-05})	//	//

Table 4.56: Confusion Matrix obtained for Hybrid CNN with Augmentations + DNN.

	Q1	**Q2**	**Q3**	**Q4**	**F1-Score Per Quadrant**
Q1	**1681**	270	107	172	**70.30%**
Q2	310	**1843**	61	36	**81.62%**
Q3	194	91	**1319**	606	**61.31%**
Q4	332	50	570	**1288**	**58.93%**
				F1 Score Total	**68.18%**

4.8 Ensemble Method: Hybrid CNN with Augmentations + DNN

The last methodology developed, was an improvement on the Ensemble method presented in the foundation work, previously discussed in Section 4.1.10. Data augmentation has had some success when using classical audio augmentation techniques, which can be seen in the experiments concerning these.

To further assess their impact, the CNN portion of the Hybrid model was pre-trained using the set of augmentation studied in the foundation work. Due to the already mentioned resource constraints, this could not be tested with the techniques experimented with in the present work.

Various approaches were taken, firstly only pre-training the CNN portion, but latter, improvements were noted when pretraining the DNN portion of the model as well.

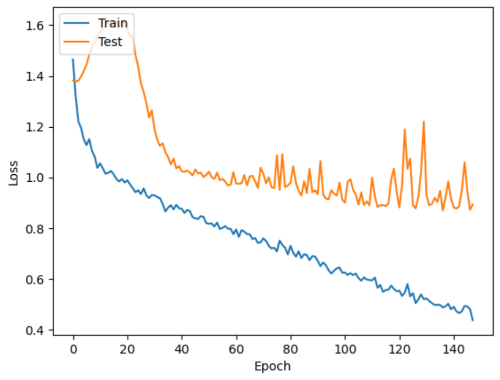
The results in Table 4.56 show a significant improvement over the baseline, achieving a **68.18% F1-Score** for the Legacy-MERGE dataset, but it did not significantly improve over its non-augmented counterpart. However, when using the New-MERGE dataset, as seen in Table 4.57, there were very significant improvements on the balanced datasets, achieving a **80.22%** and **80.24% F1 Score**, which make them state of the art for this dataset. These improvements are very exciting and urges for more effort in Data Augmentation for any future methodologies.

Table 4.57: Results and comparisons obtained for Hybrid CNN with Augmentations + DNN.

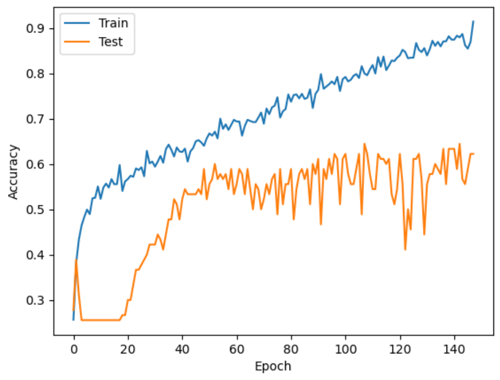
	F1 Score	Statistical Significance Test Against:		
		Baseline	Balanced	Balanced-Genre
Complete	**67.85%**	Significant ($3.7057\text{x}10^{-07}$)	Not Significant	Significant ($1.1712\text{x}10^{-56}$)
Balanced	**80.22%**	Significant ($5.5791\text{x}10^{-47}$)	//	Significant ($4.0674\text{x}10^{-128}$)
Balanced-Genre	**80.24%**	Significant ($1.2784\text{x}10^{-45}$)	//	//

4.9 A note on the Evaluation Methodology

Due to the impossibility to experiment with all possible hyperparameters for each model evaluated, a methodology was defined and followed when developing and evaluating each new approach presented in this work to ensure the best performance under the resources' constraints.



(a) Example of a train and test loss plot.



(b) Example of a train and test accuracy plot.

Figure 4.23: Examples of loss and accuracy plots

For the approaches replicated from the literature, if any set of optimal hyperparameters is provided, these are used as the starting point. After first training the model, the resulting metrics, loss and accuracy are plotted in order to assess what changes may be made, which can be seen in Fig. 4.23. Taking as an example a plot of the loss during training that plateaued at a relatively early epoch, various paths can be taken, such as reducing the number of epochs to save on resources or experimenting with a larger batch size to get better performance. In the case a plot of the accuracy during training seems to reach a plateau at a very early epoch, it is possible that a change to the optimizer used needs to be made, changing it entirely or reducing the learning rate. For approaches that were not directly replicated from the literature or approaches from the literature that do not provide a set of hyperparameters, the optimal parameters from similar already evaluated models were used.

For training and testing, cross-validation using 10 splits and 10 repeats is used, meaning that the dataset in question is split into 10 random slices, 9 are used for training and the remaining one is used for testing, one particular train and test split as described is known as train and test folds, and at some point, all slices are utilized as train and test data. The process of randomly selecting the slices is done 10 times, resulting in a total of 100 different folds that are used to evaluate each model. This is done to circumvent the low number of samples on the studied datasets, as was done in previous work by the team [8].

4.10 Summary of presented approaches

As to put into perspective the different methodologies studied in this work, a summary of the most important details of each one are presented in Tables 4.58, 4.59 and 4.60. The presented F1 Scores are the best ones found between the Legacy- and New-MERGE datasets, as to provide an idea on the limitations of each methodology. The time estimations provided are very rough, since the development environment was not the most stable. Not only where the used GPUs dependant on the load of these environment, as well as the speed at which computations could be made, impacted by other users' usage.

To end this section, Figs. 4.24 and 4.25 present the F1-Scores obtained between the datasets by ascending order of the dataset where the methodologies were developed, meaning that the best performing methodologies for the Legacy-MERGE dataset for the foundation work and New-MERGE dataset for the newly explored methodologies are to the right of the respective figures, as to give an idea on the limits of the datasets themselves.

Table 4.58: A Summary of Replicated Approaches.

Experiment	Data (Num Samples) (Length)	Input (Sample Rate) (Dim)	Architecture Description	Num. Parameters (All) (Trainable)	Batch (Size)	Epochs (Num)	Optimizer (LR)	Time Est. (Min)	Best Overall F1 Score
SVM with Top 100 (Panda)	Legacy-MERGE (900)	100 Handcrafted Features	Support Vector Machine (Classifier)	N.A.	N.A	N.A	N.A	4.5	74.89%
Baseline - Simple CNN (Sá)	Legacy-MERGE (900) (30 sec)	Mel-spectrograms (16kHz) (942x128x1)	CNN (Feature Extractor) ->FCN (Classifier) ->Quadrant	621.360 621.264	150	200	SGD (0.01)	1269.7	61.66%
Split CNN (Sá)	Split Legacy-MERGE (1800) (30 sec)	Mel-spectrograms (16kHz) (471x128x1)	CNN (Feature Extractor) ->FCN (Classifier) ->Quadrant	1.272.028 1.271.900	150	200	SGD (0.01)	1261.3	64.77%
Simple CNN with Aug (Sá)	Legacy-MERGE + Augmented Samples (900 + 3600)(30 sec)	Mel-spectrograms (16kHz) (942x128x1)	CNN (Feature Extractor) ->FCN (Classifier) ->Quadrant	621.360 621.264	120	450	Adam (0.005)	1446	61.55%
Simple CNN with GAN (Sá)	Legacy-MERGE + GAN generated per quad (900 + (25 * 4)) (30 sec)	Mel-spectrograms (16kHz) (942x128x1)	CNN (Feature Extractor) ->FCN (Classifier) ->Quadrant	621.360 621.264	200	100	Adam (0.01)	285.3	55.89%
Pre-Trained CNN with Genre (Sá)	GTZAN (Pre-Train) Legacy-MERGE (900 / 6300) (30 sec / 1.5 sec)	Mel-spectrograms (16kHz & 0.5 sec overlap) (129x128x1)	CNN (Feature Extractor) ->FCN (Classifier) ->Quadrant	2,654,724 164.740 (After Pre -Training)	1800	20	Adam (0.05)	49.8	19.23%
Voice CNN - Full Branch + Voice Branch (Sá)	Legacy-MERGE (900) (30 sec)	2x Mel-spectrograms (16kHz) (942x128x1) [Full Sample + Voice Only Sample]	[CNN (Feature Extractor - Full) + CNN (Feature Extractor - Voice)] ->FCN (Classifier) ->Quadrant	1.320.156 1.318.764	150	200	SGD (0.01)	1660.2	61.44%
AV CNN - Arousal Branch + Valence Branch (Sá)	Legacy-MERGE (900) (30 sec)	Mel-spectrograms (16kHz) (942x128x1)	CNN (Feature Extractor) ->FCN (Classifier) ->[Arousal + Valence]	1.270.642 1.270.386	300	100	SGD (0.01)	320.3	49.93%
DNN All (Sá)	Legacy-MERGE (900)	1714 Handcrafted Features	FCN (Preprocessing) ->FCN (Classifier) ->Quadrant	1,045,160 1,041,732	450	10	Adam (0.0005)	4.1	69.21%
DNN Top 100 (Sá)	Legacy-MERGE (900)	100 Handcrafted Features	FCN (Preprocessing) ->FCN (Classifier) ->Quadrant	15.754 15.554	150	300	SGD (0.01)	28.1	72.48%
DNN Top 100 + 11 Spotify (Sá)	Legacy-MERGE - Available on Spotify (704)	111 Handcrafted Features	FCN (Preprocessing) ->FCN (Classifier) -> Quadrant	18.236 18.236	300	80	Adam (0.0005)	9.1	73.73%
Emsemble - CNN Branch + DNN Branch (Sá)	Legacy-MERGE (900) (30 sec)	Mel-spectrograms (16kHz)(942x128x1) + 1714 Handcrafted Features	[FCN (Preprocessing) + CNN (Feature Extractor)] ->FCN (Classifier) ->Quadrant	1.767.280 721.596	300	100	SGD (0.01)	222.8	67.63%

Table 4.59: A Summary of Developed Methodologies.

Experiment	Data (Num Samples) (Length)	Input (Sample Rate) (Dim)	Architecture Description	Num. Parameters (All) (Trainable)	Batch (Size)	Epochs (Num)	Optimizer (LR)	Time Est. (Min)	Best Overall F1 Score
ShortChunk CNN (Louro)	Legacy-MERGE-C (893 / 6993) (30 sec / 3.69 sec)	Mel-spectrograms (16kHz) (116x128x1)	CNN (Feature Extractor) ->FCN (Classifier) ->Quadrant	12.092.424 12.080.646	50	100	SGD (0.001)	840	54.00%
Sample CNN (Louro)	Legacy-MERGE-C (893 / 6993) (30 sec / 3.69 sec)	Raw Waveform (16kHz) (59049x1)	CNN (Feature Extractor) ->FCN (Classifier) ->Quadrant	1.852.164 1.846.276	50	150	SGD (0.001)	635	52.60%
Simple CNN + RNN Improvement (Louro)	Legacy-MERGE-C (893) (30 sec)	Mel-spectrograms (16kHz) (942x128x1)	CNN (Feature Extractor) ->RNN (Feature Extractor) ->FCN (Classifier) ->Quadrant	48.667.035 48.666.939	150	200	SGD (0.01)	700	61.99%
CRNN (Louro)	Legacy-MERGE-C (893) (30 sec)	Mel-spectrograms (16kHz) (942x128x1)	CNN (Feature Extractor) ->RNN (Feature Extractor) ->FCN (Classifier) ->Quadrant	392.480 392.582	50	200	SGD (0.001)	940.5	63.33%
Simple CNN with Time-Frequency Masking Augmentation (Louro)	Legacy-MERGE-C + Time-Frequency Masking (893 + 893)(30 sec)	Mel-spectrograms (16kHz) (942x128x1)	CNN (Feature Extractor) ->FCN (Classifier) ->Quadrant	621.360 621.264	150	200	SGD (0.01)	1482.5	62.03%
Simple CNN with Tanh Distortion Augmentation (Louro)	Legacy-MERGE-C + Tanh Distortion (893 + 893)(30 sec)	Mel-spectrograms (16kHz) (942x128x1)	CNN (Feature Extractor) ->FCN (Classifier) ->Quadrant	621.360 621.264	150	200	SGD (0.01)	696.3	60.59%
Simple CNN with Random Gain Augmentation (Louro)	Legacy-MERGE-C + Random Gain (893 + 893)(30 sec)	Mel-spectrograms (16kHz) (942x128x1)	CNN (Feature Extractor) ->FCN (Classifier) ->Quadrant	621.360 621.264	150	200	SGD (0.01)	1245.5	62.24%
Simple CNN with Background Noise Augmentation (Louro)	Legacy-MERGE-C + Background Noise (893 + 893)(30 sec)	Mel-spectrograms (16kHz) (942x128x1)	CNN (Feature Extractor) ->FCN (Classifier) ->Quadrant	621.360 621.264	150	200	SGD (0.01)	843.5	60.84%
Simple CNN with Seven-Band Parametric Eq. Augmentation (Louro)	Legacy-MERGE-C + Seven-Band Parametric Eq. (893 + 893)(30 sec)	Mel-spectrograms (16kHz) (942x128x1)	CNN (Feature Extractor) ->FCN (Classifier) ->Quadrant	621.360 621.264	150	200	SGD (0.01)	1096.2	62.12%
Simple CNN with Deep SMOTE Generated Samples (Louro)	Legacy-MERGE-C + Deep SMOTE generated per quad (893 + (25 * 4)) (30 sec)	Mel-spectrograms (16kHz) (942x128x1)	CNN (Feature Extractor) ->FCN (Classifier) ->Quadrant	621.360 621.264	200	150	SGD (0.01)	557.4	61.47%

Table 4.60: Continuation of Summary of Developed Methodologies.

Experiment	Data (Num Samples) (Length)	Input (Sample Rate) (Dim)	Architecture Description	Num. Parameters (All) (Trainable)	Batch (Size)	Epochs (Num)	Optimizer (LR)	Time Est. (Min)	Best Overall F1 Score
Pre-Trained Artist CNN (Louro)	Legacy-MERGE-C (893 / 8930) (30 sec / 3 sec)	Mel-spectrograms (22.05kHz) (129x128x1)	CNN (Feature Extractor) ->FCN (Classifier) ->Quadrant	267.01 21.028 (After Pre-Training)	100	200	Adam (0.01)	593.2	50.85%
Pre-Trained CRNN with MTAT (Louro)	Legacy-MERGE-C (893) (30 sec)	Mel-spectrograms (16kHz) (942x128x1)	CNN (Feature Extractor) ->RNN (Feature Extractor) ->FCN (Classifier) ->Quadrant	392.582 392.582	16	200 (As per Won Optimizer)	Adam (0.001) SGD (0.0001) SGD (0.00001)	748.3	49.81%
Pre-Trained CRNN with JAM (Louro)	Legacy-MERGE-C (893) (30 sec)	Mel-spectrograms (16kHz) (942x128x1)	CNN (Feature Extractor) ->RNN (Feature Extractor) ->FCN (Classifier) ->Quadrant	392.582 392.582	16	200 (As per Won Optimizer)	Adam (0.001) SGD (0.0001) SGD (0.00001)	748.3	43.73%
Pre-Trained CRNN with MSD (Louro)	Legacy-MERGE-C (893) (30 sec)	Mel-spectrograms (16kHz) (942x128x1)	CNN (Feature Extractor) ->RNN (Feature Extractor) ->FCN (Classifier) ->Quadrant	392.582 392.582	16	200 (As per Won Optimizer)	Adam (0.001) SGD (0.0001) SGD (0.00001)	748.3	41.29%
Open-L3 Embeddings (Louro)	Legacy-MERGE-C (893) (30 sec)	Deep Audio Embeddings (N.A.)(298x512)	Random Forest (Classifier)	N.A.	N.A.	N.A.	N.A.	1.66	55.70%
Autoencoder Embeddings (Louro)	Legacy-MERGE-C (893) (30 sec)	Deep Audio Embeddings (N.A.)(60416)	Random Forest (Classifier)	N.A.	N.A.	N.A.	N.A.	1.66	53.69%
Multiple Sample-Level Representations (Louro)	Legacy-MERGE-C (893) (30 sec)	Mel-spectrograms (16kHz)(942x128x1) MFCCs (16kHz)(942x20x1) STFTs (16kHz)(1876x515x1)	[CNN (Feature Extractor - Mel) + CNN (Feature Extractor - MFCC) + CNN (Feature Extractor - STFT)] ->FCN (Classifier) ->Quadrant	4.008.374 4.006.318	32	100	SGD (0.01)	2342.8	74.23%
Multiple Segment-Level Representations (Louro)	Legacy-MERGE-C (893 / 5358) (30 sec / 5 sec)	Mel-spectrograms (22.05kHz)(216x128x1) MFCCs (22.05kHz)(216x20x1) STFTs (22.05kHz)(431x515x1)	[CNN (Feature Extractor - Mel) + CNN (Feature Extractor - MFCC) + CNN (Feature Extractor - STFT)] ->FCN (Classifier) ->Quadrant	2.054.274 2.054.274	16	50	SGD (0.01)	2673.6	49.81%
Emsemble - CNN Augmented Branch + DNN Branch (Louro)	Legacy-MERGE-C (893 + 3572) (30 sec)	Mel-spectrograms (16kHz)(942x128x1) + 1714 Handcrafted Features	[FCN (Preprocessing) + CNN (Feature Extractor)] ->FCN (Classifier) ->Quadrant	1.767.280 721.596	300	100	SGD (0.01)	188.4	80.24%

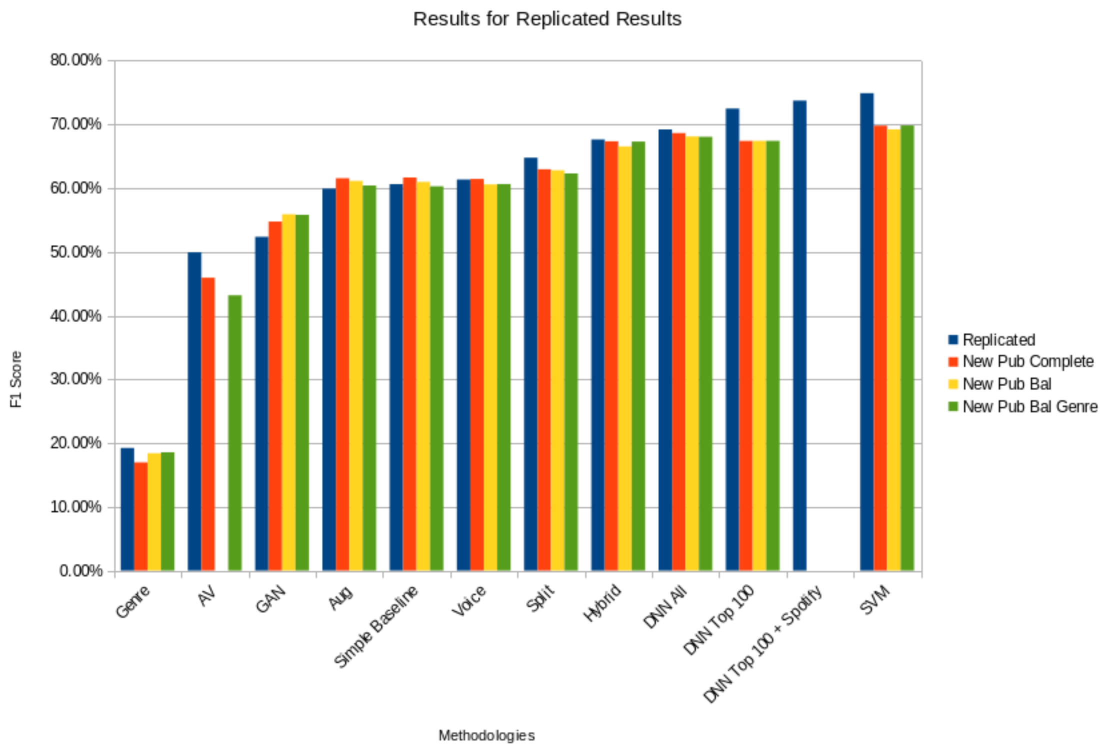


Figure 4.24: Results for Replicated Work.

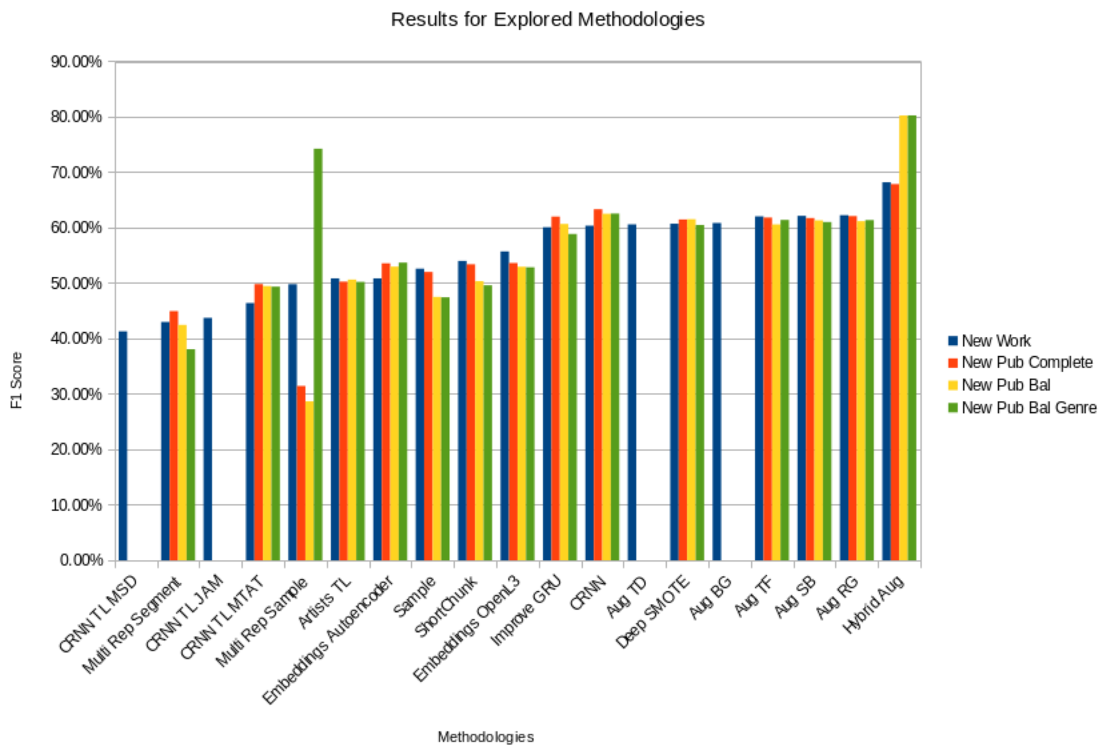


Figure 4.25: Results for Explored Methodologies.

Chapter 5

Conclusions and Future Work

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

5.1 Conclusion

As this work comes to its end, the most relevant conclusions and main contributions are presented and discussed.

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. This has been stated multiple times in this work and as the data shows, it truly is the most impactful factor for performance at this stage for solving MER.

Still on improving the performance of this methodologies, there is an interest in testing with different hyperparameters for the DL models, most importantly, the batch size. Although some methodologies seem to perform better with a higher number of epochs, batch size has been the most impactful hyperparameter. For example, there is evidence that higher batch sizes may positively impact the Sample-level Multiple Representations methodology, as was shown by the considerable performance increase when increasing the batch size, however, the sheer complexity of the model did not make this possible with the available resources.

There is some interest in continue experimenting with time-distributed features with recurrent units, as these have shown to be at its best when there is a more considerable number of samples, increasing the need to further update our datasets.

Shifting the focus on the overall methodologies employed, it was apparent that transfer learning has not improved the performance of the models in any way. This was seen in the foundation work and the tendency was sustained in the present work, should it be out-of-domain knowledge or in-domain knowledge transfer learning, as was seen with the Artists CNN architecture with pre-trained weights and the larger Static MER datasets experimented with the CRNN architecture. Nonetheless, there are other methods to employ it which have not been experimented with, some being presented in the next section.

Another overall methodology that deserves a lot more research is Data Augmentation. It has been shown that classical Data Augmentation has a positive impact in the models where these are applied, but there is still not much information regarding the effects of most of these in MER. This being more effective than Transfer Learning to mitigate lack of data quantity, it should be of great priority in future work.

Multiple representations, although not showing very promising results initially, the obtained results for the New-MERGE dataset indicate a very promising direction. Since only three representations were tested in tandem, there is still a lot of possibilities in this area, such as the work by Zang et al. [82], where the Legacy-MERGE dataset was adopted for evaluation. Here, Mel-spectrograms and the Mel Filter Banks used to generate them are used in combination to produce very impressive results. Although there was interest in pursuing this and the author contributed some pointers for replicating the experiment, it was not possible to conduct and is expected to be explored in the future.

Some of the best results obtained are still the ones that incorporate handcrafted features in some way. These should be further pursued in order to enrich DL models with intricacies that may not be apparent with such a small quantity of data.

With this work, we also have proved the viability of the extension to the static dataset created by the team. The preliminary results will be used as basis to further improve on the hyperparameters used in these methodologies, as performance gains are expected.

We also provide a very detailed run down of all replicated and developed methodologies as future reference for future work using DL approahces.

5.2 Future Work

There are a lot of aspects that can be improved in future work. Here, a summary of the most promising directions is presented. These are:

- Expand the Static MER datasets, as it has been shown that is the most crucial point for improving performance;
- Assuring augmented samples retain the emotion pertaining their original quadrant by splicing the vocals and instrumentals of songs in the same quadrants;
- Deeper study on applying classical audio augmentations with an automated search approach, looking for the best combination of augmentations possible;
- Pre-training the voice isolated branch of the Voice CNN model with data from Speech Emotion Recognition to improve performance;
- Pre-training deep embedding models, as was done for OpenL3, with other domain knowledge that may be more appropriate for the problem at hand.

Other promising direction not explored in this work include:

- Applying unsupervised learning to automatically label data, reducing the manual labor necessary;
- Introducing attention layers between the feature extraction and classifier portions of DL models may improve the performance by removing redundant features;
- Study the quality of larger datasets when projected to the A-V plane, such as the MSD dataset, which large quantity of available social tags may produce good results.

On a personal note, developing this work has been a very enriching experience, not only in a professional, but also in a personal level. The sheer complexity of the topic was very hard to get into in the beginning, but with a lot of perseverance, we were able to grasp the necessary concepts to provide a very extensive and comprehensive work, which we sincerely hope to be a reference for any future work on our team.

This page is intentionally left blank.

References

[1] A. Huq; J. P. Bello and R. Rowe. Automated music emotion recognition: A systematic evaluation. *Journal of New Music Research*, 39:227 – 244, 2010.

[2] Y.-H. Yang; Y.-C. Lin and H. Chen. Personalized music emotion recognition. pages 748–749, 01 2009.

[3] T. Zhang; A. E. Ali; C. Wang; A. Hanjalic and P. Cesar. Cornet: Fine-grained emotion recognition for video watching using wearable physiological sensors. *Sensors*, 21(1), 2021.

[4] K. Avramidis; C. Garoufis; A. Zlatintsi and P. Maragos. Enhancing affective representations of music-induced eeg through multimodal supervision and latent domain adaptation, 2022.

[5] B.-H. Sungand and S.-C. Wei. Becmer: A fusion model using bert and cnn for music emotion recognition. In *2021 IEEE 22nd International Conference on Information Reuse and Integration for Data Science (IRI)*, pages 437–444, 2021.

[6] R. Panda; R. Malheiro and R. P. Paiva. Musical texture and expressivity features for music emotion recognition. *19th International Society for Music Information Retrieval Conference*, 2018.

[7] P. Sá. Merge audio: Music emotion recognition next generation - audio classification with deep learning, October 2021.

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

[9] P. R. Kleinginna Jr. and A. M. Kleinginna. A categorized list of emotion definitions, with suggestions for a consensual definition. *Motivation and Emotion*, (5):345–379, 1981.

[10] R. Panda. *Emotion-based Analysis and Classification of Audio Music.* PhD thesis, Universidade de Coimbra, January 2019.

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

[12] T. Eerola and J. K. Vuoskoski. A comparison of the discrete and dimensional models of emotion in music. *Psychology of Music*, 39(1):18–49, 2011.

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

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

[15] P. R. Farnsworth. A stufy of the hevner adjective list. *The Journal of Aesthetics and Art Criticism*, 13:97–103, 1954.

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

[17] M. Zentner; D. Grandjean and K. Scherer. Emotions evoked by the sound of music: Characterization, classification, and measurement. *Emotion (Washington, D.C.)*, 8:494–521, 09 2008.

[18] J. Posner; J. A. Russell and B. S. Peterson. The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. 17(3), 2005.

[19] J. Russel. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178, 1980.

[20] O. Meyers. *A mood-based music classification and exploration system.* PhD thesis, Massachusetts Institute of Technology, School of Architecture and Planning, June 2007.

[21] K. Trohidis; G. Tsoumakas; G. Kalliris and I. Vlahavas. Multi-label classification of music by emotion. 2011.

[22] X. Yang; Y. Dong and J. Li. Review of data features-based music emotion recognition methods. *Multimedia Systems*, 24(4):365–389, 2018.

[23] S.-Y. Wang; J.-C. Wan; Y.-H. Yang and H.-M. Wang. Towards time-varying music auto-tagging based on cal500 expansion. volume 2014, pages 1–6, 07 2014.

[24] E. Law; K. West; M. Mandel; M. Bay and J. Downie. Evaluation of algorithms using games: The case of music tagging. pages 387–392, 2009.

[25] M. Won; A. Ferraro; D. Bogdanov; and X. Serra. Evaluation of cnn-based automatic music tagging models, 2020.

[26] T. Bertin-Mahieux; D. Ellis; B. Whitman and P. Lamere. The million song dataset. pages 591–596, 2011.

[27] R. Malheiro; R. Panda; P. Gomes and R. P. Paiva. Bi-modal music emotion recognition: Novel lyrical features and dataset. 2016.

[28] A. Aljanaki; Y.-H. Yang and M. Soleymani. Developing a benchmark for emotional analysis of music. *PLOS ONE*, 12(3):1–22, 2017.

[29] K. Zhang; H. Zhang; S. Li; C. Yang and L. Sun. The pmemo dataset for music emotion recognition. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, ICMR '18, pages 135–142, 2018.

[30] Y.-A. Chen; Y.-H. Yang; J.-C. Wang and H. Chen. The amg1608 dataset for music emotion recognition. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 693–697, 2015.

[31] D. Bogdanov; M. Won; P. Tovstogan; A. Porter and X. Serra. The mtg-jamendo dataset for automatic music tagging. In *ICML 2019*, 2019.

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

[33] J. Fan; Y.-H. Yang; K. Dong and P. Pasquier. A comparative study of western and chinese classical music based on soundscape models. International Conference on Acoustics, Speech, and Signal Processing, 2020.

[34] H. Demuth and B. Jesús. Neural network design 2nd edition, 2014.

[35] K. Choi; G. Fazekas and M. Sandler. Automatic tagging using deep convolutional neural networks, 2016.

[36] M. Malik; S. Adavanne; K. Drossos; T. Virtanen; D. Ticha and R. Jarina. Stacked convolutional and recurrent neural networks for music emotion recognition, 2017.

[37] S. Hizlisoy; S. Yildirim and Z. Tufekci. Music emotion recognition using convolutional long short term memorydeep neural networks. *Engineering Science and Technology, an International Journal*, 24:760–767, 2021.

[38] Q. Zhong; Y. Zhu; D. Cai; L. Xiao and H. Zhang. Electroencephalogram access for emotion recognition based on a deep hybrid network. *Frontiers in Human Neuroscience*, 14, 2020.

[39] D. S. Park; W. Chan; Y. Zhang; C.-C. Chiu; B. Zoph; E. D. Cubuk and Q. V. Le. SpecAugment: A simple data augmentation method for automatic speech recognition. In *Interspeech 2019*. ISCA, 2019.

[40] H. H. Tan. Semi-supervised music emotion recognition using noisy student training and harmonic pitch class profiles, 2021.

[41] K. J. Piczak. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pages 1015–1018. ACM Press.

[42] I. Goodfellow; J. Pouget-Abadie; M. Mirza; B. Xu; D. Warde-Farley; S. Ozair; A. Courville and Y. Bengio. Generative adversarial networks, 2014.

[43] D. Pinho. Music generation using generative adversarial networks, June 2018.

[44] A. Radford; L. Metz and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016.

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

[46] R. Panda; R. Malheiro and R. P. P. Audio features for music emotion recognition: a survey. *IEEE Transactions on Affective Computing*, Early Access Article, 2020.

[47] K. Markov and T. Matsui. *Speech and Music Emotion Recognition Using Gaussian Processes*, pages 63–85. Springer Japan, Tokyo, 2015.

[48] J. Wang. An intuitive tutorial to gaussian processes regression, 2021.

[49] M. Bargaje. Emotion recognition and emotion based classification of audio using genetic algorithm - an optimized approach. In *2015 International Conference on Industrial Instrumentation and Control (ICIC)*, pages 562–567, 2015.

[50] Y. Feng; Y. Zhuang and Y. Pan. Music information retrieval by detecting mood via computational media aesthetics. In *Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003)*, pages 235–241, 2003.

[51] P. Juslin. From everyday emotions to aesthetic emotions: Toward a unified theory of musical emotions. *Physics of life reviews*, 10:235–266, 05 2013.

[52] K. Choi; G. Fazekas and M. Sandler. Automatic tagging using deep convolutional neural networks, 2016.

[53] K. Choi; G. Fazekas; M. Sandler and Ky. Cho. Convolutional recurrent neural networks for music classification, 2016.

[54] J. S. Gómez-Cañón; E. Cano; P. Herrera and E. Gómez. Transfer learning from speech to music: towards language-sensitive emotion recognition models. 2021.

[55] J. S. Gómez-Cañón; P Herrera; E. Gómez and E. Cano. The emotions that we perceive in music: the influence of language and lyrics comprehension on agreement, 2019.

[56] J. Grekow. Music emotion recognition using recurrent neural networks and pretrained models. *Journal of Intelligent Information Systems*, 2021.

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

[58] S. Lang; F. Bravo-Marquez; C. Beckham; M. Hall and E. Frank. Wekadeeplearning4j: A deep learning package for weka based on deeplearning4j. *Knowledge-Based Systems*, 178:48–50, 2019.

[59] J. Yang. A novel music emotion recognition model using neural network technology. *Frontiers in Psychology*, 12, 2021.

[60] X. Zhou; J. Lu; J. Huang; M. Zhong and M. Wang. Enhancing artificial bee colony algorithm with multi-elite guidance. *Information Sciences*, 543:242–258, 2021.

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

[62] A. Gabrielsson and E. Lindström. The influence of musical structure on emotional expression. 2001.

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

[64] I. Goodfellow; D. Warde-Farley; M. Mirza; A. Courville and Y. Bengio. Maxout networks, 2013.

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

[66] Y Dong; X. Yang; X. Zhao and J. Li. Bidirectional convolutional recurrent sparse network (bcrsn): An efficient model for music emotion recognition. *IEEE Transactions on Multimedia*, 21:3150–3163, 2019.

[67] R. Orjesek; R. Jarina and M. Chmulik. End-to-end music emotion variation detection using iteratively reconstructed deep features. *Multimedia Tools and Applications*, pages 1–15, 2022.

[68] S. Dieleman and B. Schrauwen. End-to-end learning for music audio. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6964–6968, 2014.

[69] M. Robnik-Sikonja and I. Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Machine Learning*, 53:23–69, 2003.

[70] Y.-H. Yang; Y.-C. Lin; Y.-F. Su and H. H. Chen. A regression approach to music emotion recognition. *IEEE Transactions On Audio, Speech, and Language Processing*, 16(2):448–457, 2008.

[71] J. Lee; J. Park; K. L. Kim and J. Nam. Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms, 2017.

[72] J. Pons; O. Nieto; M. Prockup; E. Schmidt; A. Ehmann and X. Serra. End-to-end learning for music audio tagging at scale, 2017.

[73] K. He; X. Zhang; S. Ren and J. Sun. Deep residual learning for image recognition, 2015.

[74] R. Mignot and G. Peeters. An analysis of the effect of data augmentation methods: Experiments for a musical genre classification task. *Transactions of the International Society for Music Information Retrieval*, 2:97–110, 2019.

[75] N. Chawla; K. Bowyer; L. Hall and W. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[76] S. Maldonado; J. López and C. Vairetti. An alternative smote oversampling strategy for high-dimensional datasets. *Applied Soft Computing*, 76:380–389, 2019.

[77] D. Dablain; B. Krawczyk and N. Chawla. Deepsmote: Fusing deep learning and smote for imbalanced data, 2021.

[78] G. Kovács. smote-variants: a python implementation of 85 minority oversampling techniques. *Neurocomputing*, 366:352–354, 2019.

[79] G. Kovács. An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets. *Applied Soft Computing*, 83:105662, 2019.

[80] J. Park; J. Lee; J. Park; J.-W. Ha and J. Nam. Representation learning of music using artist labels, 2017.

[81] K. Choi; G. Fazekas; M. Sandler and K. Cho. Transfer learning for music classification and regression tasks, 2017.

[82] M. Zhang; Y. Zhu; N. Ge; Y. Zhu; T. Feng and W. Zhang. Attention-based joint feature extraction model for static music emotion classification. In *2021 14th International Symposium on Computational Intelligence and Design (ISCID)*, pages 291–296, 2021.

[83] M. B. Akçay and K. Oğuz. Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers. *Speech Communication*, 116:56–76, 2020.

[84] T. Bathigama and S. Madushika. Multi-Representational Music Emotion Recognition using Deep Convolution Neural Networks. 2021.

[85] J. Cramer; H.-H. Wu; J. Salamon and J. P. Bello. Look, listen, and learn more: Design choices for deep audio embeddings. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3852–3856, 2019.

[86] E. Koh and S. Dubnov. Comparison and analysis of deep audio embeddings for music emotion recognition. 2021.

[87] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[88] R. Arandjelović and A. Zisserman. Look, listen and learn, 2017.

[89] J. F. Gemmeke; D. P. W. Ellis; D. Freedman; A. Jansen; W. Lawrence; R. C. Moore; M. Plakal and M. Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780, 2017.

[90] A. Cockburn; P. Dragicevic; L. Besançon and C. Gutwin. Threats of a replication crisis in empirical computer science. *Communications of the ACM*, 63:70–79, 2020.