Towards global tempo estimation and rhythm-oriented genre classification based on harmonic characteristics of rhythm

Hadrien Foroughmand Aarabi

THÈSE DE DOCTORAT DE SORBONNE UNIVERSITÉ

Spécialité Informatique
ED130 - Ecole doctorale Informatique, Télécommunications et Electronique (Paris)
Sciences et Technologie de la Musique et du Son (UMR 9912)
Institut de Recherche et de Coordination Accoustique Musique
Equipe Analyse/Synthèse des Sons.

TOWARDS GLOBAL TEMPO ESTIMATION AND RHYTHM-ORIENTED GENRE CLASSIFICATION BASED ON HARMONIC CHARACTERISTICS OF RHYTHM

FOCUS ON ELECTRONIC/DANCE MUSIC

PAR HADRIEN FOROUGHMAND AARABI
DIRIGÉ PAR : GEOFFROY PEETERS

January 2021

Hadrien Foroughmand Aarabi: *Towards global tempo estimation and rhythm-oriented genre classification based on harmonic characteristics of rhythm, Focus on Electronic/-Dance Music,* Ph.D Student Researcher, © November 2020

"For me, electronic music is the classical music of the 21st century." — Jean-Michel Jarre

"One good thing about music, when it hits you, you feel no pain." — Bob Marley

ABSTRACT

Rhythm is one of the major characteristics when talking about music. When we want to describe a piece of music, we often base ourselves on its rhythmic structure. Automatically detecting this type of information within the music is one of the challenges of the Music Information Retrieval research field. In recent years, the advent of technology dedicated to art has allowed the emergence of new musical trends generally described by the term Electronic/Dance Music (EDM). EDM is an umbrella term that encompasses a plethora of sub-genres often composed with electronic instruments such as synthesizers, sequencers but also computer-aided composition software. The music denoted by this term is often dedicated to dance and is therefore characterized by its rhythmic structure. With the popularity of EDM, new sub-genres are emerging every week and are no longer necessarily defined by their musical aspects.

In this manuscript, we propose a rhythmic analysis that defines certain musical genres including those of EDM. For this purpose, we wish to perform two tasks, (i) a task of automatic estimation of the global tempo; and (ii) a task of rhythm-oriented genre classification. Tempo and genre are two inter-leaved aspects since genres are often associated with rhythm patterns which are played in specific tempo ranges.

Some handcrafted tempo estimation systems have shown their efficiency based on the extraction of rhythm-related features. Among other things, it has been shown that the harmonic series at the tempo frequency of the onset-energy-function of an audio signal accurately describes its rhythm-pattern and can be used to perform tempo or rhythm pattern estimation. Recently, with the appearance of annotated datasets, data-driven systems and deep-learning approaches have shown progress in the automatic estimation of these tasks. In the case of multi-pitch estimation, the depth of the input layer of a convolutional network has been used to represent the harmonic series of pitch candidates. We use a similar idea here to represent the harmonic series of tempo candidates.

In this thesis, we propose a method at the crossroads between handcrafted and data-driven systems. We propose the Harmonic-Constant-Q-Modulation which represents, using a 4D tensor, the harmonic series of modulation frequencies (considered as tempo frequencies) in several acoustic frequency bands over time. This representation is used as input to a convolutional network which is trained to estimate tempo or rhythm-oriented genre classes. This method, called Deep Rhythm,

allows us, using the same representation and model, to accomplish the two tasks of tempo and rhythm-oriented genre estimation automatically.

Then, we propose to extend the different aspects of Deep Rhythm. To allow the representation of the relationships between frequency bands, we modify this system to process complex-valued inputs through complex-convolutions. We study the joint estimation of both tasks using a multitask learning approach. We also investigate the addition of a second input branch to the system. This branch, applied to a mel-spectrogram input, is dedicated to the representation of timbre.

Finally we propose to analyze the effect of a metric learning paradigm on our representation. Through the study of the rhythmic similarities between tracks, we try to achieve rhythm-oriented genre classification. We propose for each of the methods a thorough evaluation of the commonly-used datasets but also of two datasets annotated in EDM genres that we have created.

RÉSUMÉ

Le rythme est l'une des caractéristiques majeures lorsqu'on parle de musique. Lorsque nous voulons décrire un morceau de musique, nous nous basons souvent sur sa structure rythmique. La détection automatique de ce type d'information au sein de la musique est l'un des défis du domaine de recherche "Music Information Retrieval". Ces dernières années, l'avènement de la technologie dédiées aux arts a permis l'émergence de nouvelles tendances musicales généralement décrites par le terme d'"Electronic/Dance Music" (EDM). L'EDM est un terme général qui englobe une pléthore de sous-genres souvent composés avec des instruments électroniques tels que des synthétiseurs, des séquenceurs mais aussi des logiciels de composition assistée par ordinateur. La musique désignée par ce terme est souvent dédiée à la danse et se caractérise donc par sa structure rythmique. Avec la popularité de l'EDM, de nouveaux sous-genres apparaissent chaque semaine et ne sont plus nécessairement définis par leurs aspects musicaux.

Dans ce manuscrit, nous proposons une analyse rythmique de ce qui définit certains genres musicaux dont ceux de l'EDM. Pour ce faire, nous souhaitons réaliser deux tâches, une tâche d'estimation automatique du tempo global et une tâche de classification des genres axée sur le rythme. Le tempo et le genre sont deux aspects entremêlé puisque les genres sont souvent associés à des motifs rythmiques qui sont joués dans des plages de tempo spécifiques.

Certains systèmes d'estimation du tempo dit "handcrafted" ont montré leur efficacité en se basant sur l'extraction de caractéristiques liées au rythme. Il a notamment été démontré que la série harmonique à la fréquence du tempo de la fonction d'énergie d'attaque d'un signal audio décrit avec précision son motif rythmique et peut être utilisée pour effectuer une estimation du tempo ou du motif rythmique. Récemment, avec l'apparition de base de données annotées, les systèmes dit "data-driven" et les approches d'apprentissage profond ont montré des progrès dans l'estimation automatique de ces tâches. Dans le cas de l'estimation de fréquence fondamentale multiple, la profondeur de la couche d'entrée d'un réseau convolutif a été utilisée pour représenter les séries harmoniques des fréquences fondamentales candidates. Nous utilisons ici une idée similaire pour représenter la série harmonique des candidats au tempo.

Dans cette thèse, nous proposons une méthode à la croisée des chemins entre les systèmes "handcrafted" et "data-driven". Nous proposons la "Harmonic Constant Q Modulation" qui représente, en utilisant un tenseur 4D, la série harmonique des fréquences de modulation (considérées comme des fréquences de tempo) dans

plusieurs bandes de fréquences acoustiques au cours du temps. Cette représentation est utilisée comme entrée d'un réseau convolutif qui est entraîné pour estimer le tempo ou des classes de genre axées rhythme. Cette méthode, appelée "Deep Rhythm", nous permet, en utilisant la même représentation et le même modèle, d'accomplir les deux tâches.

Nous proposons ensuite d'étendre les différents aspects du "Deep Rhythm". Pour permettre la représentation des relations entre les bandes de fréquences, nous modifions ce système pour traiter des entrées à valeur complexe par le biais de convolutions complexes. Nous étudions également l'estimation conjointe des deux tâches en utilisant une approche d'apprentissage multitâche. Puis nous étudions l'ajout d'une deuxième branche d'entrée au système. Cette branche, appliquée à une entrée de type mel-spectrogramme, est dédiée à la représentation du timbre.

Enfin, nous proposons d'analyser l'effet d'un paradigme d'apprentissage métrique sur notre représentation. Par l'étude des similitudes rythmiques entre les pistes, nous essayons de réaliser notre tache de classification en genres rythmiques. Nous proposons pour chacune des méthodes une évaluation approfondie des bases de données de l'état de l'art mais aussi de deux bases de données annotées en genres d'EDM que nous avons créés.

PUBLICATIONS

Foroughmand, Hadrien and Geoffroy Peeters (2017). 'Multi-source musaicing using non-negative matrix factor 2-d deconvolution.' In: *18th International Society for Music Information Retrieval (ISMIR) Late-Breaking Demo Session*.

— (2018). 'Music retiler: Using NMF2D source separation for audio mosaicing.' In: *Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion*. (ACM), p. 27.

— (2019). 'Deep-Rhythm for Tempo Estimation and Rhythm Pattern Recognition.' In: *20th International Society for Music Information Retrieval (ISMIR) Conference*.

— (2020). 'Extending deep rhythm for tempo and genre estimation using complex convolutions, multitask learning and multi-input network.' In: *Proceedings of Joint Conference on AI Music Creativity, Royal Institute of Technology (KTH), Stockholm, Sweden*. (MuMe).

ACKNOWLEDGMENTS

First of all, I would like to express my sincere gratitude to my director and thesis supervisor Prof. Geoffroy Peeters for the continuous support of my Ph.D study, for his scientific expertise, his sound advice and his involvement but also for his humor and kindness.

I would also like to thank the members of the FuturePulse project for trusting me by funding my research work.

I sincerely thank all the IRCAM staff who have welcomed me over the past four years. I could not have imagined a better working environment where the excellence of research dedicated to music has greatly inspired me.

I would particularly like to thank the sound analysis/synthesis team including Axel Roebel, Rémi Mignot, Frédéric Cornu and Nicolas Obin.

I would like to thank my office partners and now friends Alice Cohen-Hadria, Gabriel Meseguer Brocal and Guillaume Doras who supported me all this time and provided me with valuable advice in the course of my research. Thanks to Yann Teytaut for his kindness and for taking the time to proofread this manuscript.

I warmly thank my ATIAM crew Vincent, Tristan, Virgile, Charles, Andrea and Lou who accompanied me since the beginning of my research and with whom I enjoy sharing around this common passion for music.

I would like to thank Mathieu, my long-time partner in higher education, for his friendship and support, without which I would certainly not have gotten this far.

Thanks to my friends Marie, Valentine, Camille, Thomas, Etienne, Jules, André, Victor, Judith, Quentin, Yannick, Romain, Yoni, Nicolas, Martin and Valentin who supported me during these years of hard work. Thank you to Adrien and his unfailing friendship who knew how to cheer me up in difficult moments.

I also thank my mother, my sister, my grandparent and my cousin Aurélien who never doubted me.

My last thanks go to Adèle who did everything she could to help me, who supported me and above all gave me the necessary strength to go through with what I had undertaken.

CONTENTS

List of Figures xiii
List of Tables xvi
Acronyms xvii
Motivations 1
1 INTRODUCTION 3
1.1 Context - Electronic Dance Music . . . 4
1.1.1 A definition . . . 4
1.1.2 History and taxonomy . . . 4
1.1.3 Electronic/Dance Music Musical Characteristics . . . 6
1.2 Dissertation Organization and main contributions . . . 8
2 FUNDAMENTALS AND STATE OF THE ART 10
2.1 Introduction . . . 10
2.2 Core Definitions . . . 11
2.2.1 Rhythm . . . 11
2.2.2 Musical genres . . . 15
2.3 Handcrafted systems . . . 16
2.3.1 Signal Fundamentals . . . 16
2.3.2 Rhythm handcrafted features . . . 18
2.3.3 Musical genre handcrafted features . . . 21
2.4 Data-driven systems . . . 22
2.4.1 Machine learning fundamentals . . . 22
2.4.2 Data-driven tempo estimation . . . 31
2.4.3 Data-driven genre classification . . . 32
2.5 Electronic/Dance Music in Music Information Retrieval . . . 33
2.6 Conclusion . . . 35
3 DATASETS 37
3.1 Introduction . . . 37
3.2 Commonly used datasets . . . 38
3.3 Electronic Dance Music Datasets . . . 44
3.4 Discussion . . . 50
4 DEEP RHYTHM 53
4.1 Introduction . . . 53
4.2 Motivations . . . 54
4.2.1 Harmonic representation of rhythm components . . . 54
4.2.2 Adaptation to a deep learning formalism . . . 55

4.3 Harmonic Constant-Q Modulation . . . 58
4.3.1 Computation . . . 58
4.3.2 Visual identification of tempo . . . 60
4.4 Deep Convolutional Neural network . . . 60
4.4.1 Architecture of the Convolutional Neural Network . . . 60
4.4.2 Training . . . 63
4.5 Aggregating decisions over time . . . 64
4.5.1 Oracle Frame Prediction . . . 65
4.5.2 Attention Mecanism . . . 65
4.6 Evaluation . . . 66
4.6.1 Tempo Estimation . . . 67
4.6.2 Rhythm-oriented genre classification . . . 75
4.7 Conclusion . . . 79
5 DEEP RHYTHM EXTENSIONS 81
5.1 Introduction . . . 81
5.2 Complex Deep Rhythm . . . 82
5.2.1 Why complex representation/convolution? . . . 82
5.2.2 Complex HCQM. . . . 83
5.2.3 Complex Convolution . . . 83
5.2.4 Evaluation . . . 84
5.3 Multitask Learning . . . 90
5.3.1 Why multitask learning? . . . 90
5.3.2 Multitask Deep Rhythm . . . 91
5.3.3 Evaluation . . . 92
5.4 Multi-Input network . . . 95
5.4.1 Why multi-input network? . . . 95
5.4.2 Multi-input Network . . . 95
5.4.3 Evaluation . . . 98
5.5 Conclusion . . . 100
6 METRIC LEARNING DEEP RHYTHM 102
6.1 Introduction . . . 102
6.2 Metric learning principles . . . 102
6.3 Losses . . . 104
6.3.1 Evolution of metric learning losses . . . 104
6.3.2 Triplet loss . . . 106
6.4 Triplet Loss Deep Rhythm . . . 108
6.4.1 Architecture . . . 108
6.4.2 Training . . . 109
6.5 Evaluation and Analysis . . . 109
6.5.1 Datasets . . . 110

6.5.2 Embedding space 110
6.5.3 Classifiers 114
6.5.4 Classification Results 115
6.6 Conclusion 116
7 CONCLUSION 118
7.1 Summary and main contributions 118
7.2 Future Works 121
7.3 Overall conclusion 122
BIBLIOGRAPHY 123

LIST OF FIGURES

Figure 2.1 Rhythm in music. . . . 12
Figure 2.2 System of automatic analysis of rhythm. . . . 18
Figure 2.3 Schematic of the "ideal-case" Attack Decay Sustain Release (ADSR) for a given note. . . . 19
Figure 2.4 Bias and variance trade-off contributing to total error. . . . 25
Figure 2.5 Flowchart of a single neuron non-linear operation, vector to scalar, applied to an input vector. . . . 25
Figure 2.6 Updating weights in a fully-connected neural network. . . . 26
Figure 2.7 Rectifier non-linearities activation functions. . . . 27
Figure 2.8 Convolution operation steps. . . . 28
Figure 2.9 Max-pooling operation steps 29
Figure 2.10 Flowchart of a Convolutional Neural Network (CNN) architecture for a classification task. . . . 30
Figure 2.11 Early stopping principle. . . . 31
Figure 3.1 ACM tempo distribution. . . . 39
Figure 3.2 BR tempo distribution. . . . 39
Figure 3.3 BR genre distribution. . . . 39
Figure 3.4 tEBR, tempo distribution. . . . 40
Figure 3.5 gEBR, genre distribution. . . . 40
Figure 3.6 tGS tempo distribution. . . . 41
Figure 3.7 GTzan Tempo, tempo distribution. . . . 41
Figure 3.8 Hains. tempo distribution. . . . 42
Figure 3.9 ISMIR04, tempo distribution. . . . 42
Figure 3.10 tLMD, tempo distribution. . . . 42
Figure 3.11 tMTG, tempo distribution. . . . 43
Figure 3.12 gMTG, tempo distribution. . . . 43
Figure 3.13 gMTG, genre distribution. . . . 43
Figure 3.14 SMC tempo distribution. . . . 44
Figure 3.15 Combined, tempo distribution. . . . 44
Figure 3.16 Drum-n-bass (110 - 150 BPM) rhythm pattern example. . . . 47
Figure 3.17 Dubstep (140 BPM) rhythm pattern example. . . . 47
Figure 3.18 Hip-Hop (Boom bap, 80 - 120 BPM) rhythm pattern example. 48
Figure 3.19 House (115 - 135 BPM) rhythm pattern example. . . . 49
Figure 3.20 Reggae-dancehall-dub (stepper style, 60 - 90 BPM) rhythm pattern example. . . . 49

Figure 3.21 Trance (125 - 150 BPM) rhythm pattern example. 50
Figure 3.22 lBP dataset, tempo distribution. 51
Figure 3.23 sBP dataset, tempo distribution. 51
Figure 4.1 Example of a harmonic representation of rhythm components. 56
Figure 4.2 Computation of the Constant-Q Transform (CQT)s of a harmonic signal according to different values of h. 57
Figure 4.3 Schematic process of training for fo estimation in polyphonic music using Harmonic Constant-Q Transform (HCQT) 58
Figure 4.4 Flowchart of the Harmonic Constant-Q Modulation (HCQM) computation steps. 61
Figure 4.5 HCQM example for a given temporal frame τ'. 62
Figure 4.6 Deep Rhythm (DR) model Architecture. 63
Figure 4.7 Time-varying tempo strategies flowcharts. 67
Figure 4.8 DR results scores compared to state-of-the-art methods. . . . 69
Figure 4.9 Validation of H and B parameters, results of DR for Combined dataset. 70
Figure 4.10 Results of time aggregation strategies. 72
Figure 4.11 Ground-truth smoothing principle. 73
Figure 4.12 Results of the smoothing (Smooth-DR) and the adaptive smoothing (Ada-smooth-DR) of the tempo ground-truth values. 74
Figure 4.13 Confusion matrices of the evaluation of the BR [Left] and the gEBR[Right] datasets using DR method. 77
Figure 4.14 Confusion matrix of the evaluation of the Gr dataset using DR method. 77
Figure 4.15 Confusion matrix of the evaluation of the sBP dataset using DR method. 78
Figure 5.1 Two examples of rhythm patterns. 82
Figure 5.2 Cplx-convolution applied to Cplx-HCQM flowchart and Cplx-DR model architecture. 85
Figure 5.3 Results of Cplx-DR and Oracle-Cplx-DR methods. 86
Figure 5.4 Results of Smooth-Cplx-DR and Ada-Smooth-Cplx-DR methods. 88
Figure 5.5 Confusion matrix of the evaluation of the sBP dataset using Cplx-DR method. 89
Figure 5.6 MutiTask Learning (MTL) strategies. 90
Figure 5.7 MTL model Architecture. 92
Figure 5.8 Log-mel Magnitude Spectrogram example 96
Figure 5.9 Multi-Input Network (MI) model architecture. 97

Figure 5.10 Confusion matrix of the evaluation of the sBP dataset using Cplx-MI method. 100
Figure 6.1 Metric learning principle. 104
Figure 6.2 Schematic representation of the *contrastive loss* and the *large margin nearest neighbours loss* 106
Figure 6.3 Schematic representation of the anchor/negative pairs (a, n) selection for a given anchor/positive (a, p) pair 107
Figure 6.4 Architecture of the Triplet Loss Deep Rhythm (TriDR) model. 109
Figure 6.5 Embedding space for the BR dataset. 111
Figure 6.6 Embedding space for the gEBR dataset. 112
Figure 6.7 Embedding space for the GTzan dataset. 113
Figure 6.8 Embedding space for the sBP dataset. 113
Figure 6.9 Results comparison between TriDR and previous methods in terms of average mean recall. 115
Figure 6.10 Confusion matrix of the evaluation of the sBP dataset using TriDR method. 116

LIST OF TABLES

Table 3–1 Utilization of commonly used datasets 45
Table 4–2 Results of rhythm-pattern recognition/rhythm-oriented genre classification in term of average-mean-recall $\hat{R}$. 76
Table 5–3 Results of rhythm-pattern recognition/rhythm-oriented genre classification using Cplx-DR method compared to DR method in term of average-mean-recall $\hat{R}$. 89
Table 5–4 Separate and joint estimation results of rhythm-oriented genre classification in term of average-mean-recall $\hat{R}$. 93
Table 5–5 Separate and joint estimation results of global tempo estimation in term of Accuracy1. 93
Table 5–6 Comparative and joint estimation results of rhythm-oriented genre classification in term of average-mean-recall $\hat{R}$ for all methods. 99
Table 5–7 Comparative and joint estimation results of global tempo estimation in term of Accuracy1 for all methods. 99

ACRONYMS

ACF	Auto-correlation Function
ADSR	Attack Decay Sustain Release
AM	Attention Mechanism
BPM	Beats Per Minute
BN	Batch-Normalization
CQT	Constant-Q Transform
CNN	Convolutional Neural Network
DFT	Discrete Fourier Transform
DNN	Deep Neural Networks
DR	Deep Rhythm
EDM	Electronic/Dance Music
ELU	Exponential Linear Unit
GMM	Gaussian Mixture Model
GTTM	Generative Theory of Tonal Music
HCQM	Harmonic Constant-Q Modulation
HCQT	Harmonic Constant-Q Transform
k-NN	k-Nearest Neighbors
LSTM	Long-Term Short Memory
MI	Multi-Input Network
MIR	Music Information Retrieval
MS	Modulation Spectrum
MTL	MutiTask Learning
NC	Nearest Centroid
OEF	Onset Energy Function
OSS	Onset Signal Strength
ReLU	Rectified Linear Unit
RNN	Reccurent Neural Network
SGD	Stochastic Gradient Descent

STFT Short-Time Fourier Transform

SVM Support Vector Machine

TriDR Triplet Loss Deep Rhythm

t-SNE t-distributed Stochastic Neighbouring Entities

MOTIVATIONS

My academic work in Music Information Retrieval (MIR) domain started with an internship at IRCAM with my PhD supervisor Geoffroy Peeters. The subject of this internship was the development of a creative technique called musaicing (Zils and Pachet, 2001) which consists in concatenating musical elements while preserving the structure of the music in an automatic way. The method developed consisted more precisely in the extraction of textures from a "source" track (such as buzzing bees sounds) and to automatically reconstructe a "target" track (such as "Let it Be" by The Beatles). Thus we obtained as result the song "Let it Be" "sung" by bees. This work is based on machine learning methods such as the Non-Negative Matrix Factorization and its extension to 2 dimensions that we will not detail here. The beginning of my thesis thus began with the publication of two papers on this subject (Foroughmand and Peeters, 2017; Foroughmand and Peeters, 2018). The temporal structure of a track of music was thus very present in this creative work and made me want to study more deeply one of its main factors which became the main axis of my thesis: rhythm.

The second axis of this thesis is intrinsically linked to the FuturePulse project that founded it. The FuturePulse project aims to develop and pilot test a multi-modal predictive analytic and recommendation platform. In this context, the role of the IRCAM laboratory is to provide its technologies to estimate among other things the Beats Per Minute (BPM) tempo and the automatic genre estimation of the tracks uploaded on the platform. The work carried out during these past three years has contributed to improve these technologies.

Finally, the third axis concerns the field of application of the tasks developed in the thesis, i.e. Electronic/Dance Music (EDM). On one hand, my passion for the specific genres of this "meta-genre" has been growing for several years and is reflected in many personal projects (creation of musical playlists, first steps in computer-assisted music) but also in the fact that I am an eclectic listener. On the other hand, it is a style that is little represented in the field of MIR today, although its popularity is increasingly becoming vaster all over the world.

Our work on the structure and our interest in the EDM led us to raise the following question: the structure of an EDM track is created by superimposing samples and rhythmic patterns, is it possible to categorize the tracks from this structure?

In other words, we want to extract a characteristic rhythmic representation of the structure and use it as input to an algorithm which is able to learn a categorization into genres. In addition, since tempo is a major element of the rhythmic structure, its automatic estimation allows us to bring more knowledge to such a model.

1

INTRODUCTION

Today, technological advances continue to grow in almost every field including art. The digitization of art pieces on a computer allows all kinds of creative, analytical or scientific processing. The use of technology is thus closely linked with visual art such as cinema in terms of special effects as a creativity tool for directors or photography and graphic arts. As far as music is concerned, although its digitization is not as explicit as for the image, a lot of research is also carried out. This thesis inscribes in the field of automatic analysis of music, that is to say MIR or the ability to extract information from the musical signal through algorithms. The field of MIR encompasses many tasks and applications. We are specifically interested here in the analysis of *rhythm* and of one of its components, *tempo*, and their influence on the *musical genre* of an audio track.

When it comes to describe music, we refer to fundamental aspects such as: harmony, melody, timbre and rhythm. The sound itself is a temporal entity, the word rhythm is used in musical jargon to refer to the temporal aspect of the music. Any listener is able to perceive rhythm, it is an element intimately linked to movement (one can indeed "tap" to the rhythm). Rhythm can be defined, and distinguished, in almost every kind of music – somehow independently of cultural aspects – and therefore possesses a certain universality. Musical styles or genres are strongly characterized by their rhythm. For example, a reggae track and a techno track beyond their instrumental differences, each has a characteristic rhythm which defines them.

The main objective of this thesis is to study the efficiency of a harmonic representation of the rhythm of an audio track in order to estimate its global tempo but also its musical genre. In order to represent the effectiveness of our methods, we carry out experimental protocols on various commonly-used datasets annotated in tempo, musical genre or both. We have also chosen to focus specifically on Electronic/Dance Music (EDM) whose genres are strongly defined by their rhythm structure.

In this introduction, we set the context of the thesis in Section 1.1 with an historical and musicological description of EDM. Then, in Section 1.2, we will present the structure of the manuscript and the main contributions of our work.

1.1 CONTEXT - ELECTRONIC DANCE MUSIC

1.1.1 *A definition*

The term EDM is used to refer to a grouping of heterogeneous musical styles composed with computers and electronic instruments such as synthesizers, drum machines, sequencers and samplers. Because of the broad spectrum of different styles that EDM denotes, we can use the term "meta-genre". Slash in "Electronic/Dance Music" is important here and is used as "and/or" because not all of the music in this meta-genre is produced exclusively for dancing.

(Butler, 2006), in his book, made the most recent and general analysis of EDM. In particular, he highlights four aspects that characterize what he considers to be a heterogeneous group of practices, styles, contexts and geographical situations. The first one is the central position of the recording as the primary element of the performance. The second one is the fact that the performing audience are the dancers in opposition of the majority of other styles of music. The third one is the location of the performance with dedicated places to dance to recorded music like clubs, bars, discotheques, warehouse, raves, festivals, and so on. Finally, the last aspect is the historical common origin of the sub-genres that constitute this meta-genre: disco.

1.1.2 *History and taxonomy*

Since the 1950s, recorded music (as opposed to live music) has been played in clubs to get listeners dancing. Rock-n-roll was the popular genre to dance at that time before it evolved in the 60s into rock, a music that was disseminated through live concerts.

The use of synthesizers in popular music in the early 1970s led to the advent of disco in the United States (Ward, Stokes, and Tucker, 1986, p. 524). Disco is composed of repetitive beats created via electronic instruments mixed with traditional instruments and vocals. Before becoming mainstream, this genre was associated with urban Black and gay culture as opposed to rock music associated with White culture.

During the 1980s, disco took on a more underground aspect and became the house music of which the city of Chicago was the hometown. It is during this period that the status of the Disc-Jockey (DJ) evolved as an artist and took the place of central element of the discotheques. His instruments were as they are today: two turntables and the musical skills of managing the timing of the two

tracks in order to make smooth transitions according to their beats (beat-/tempo-matching). The goal was to make the music last during club performances, so DJs began recording extended mixes on vinyl. At the same time, and with the evolution of the technology, they started remixing tracks and creating music from scratch. They were then called *producers*.

In the late 1980s, house music crossed the Atlantic to reach the United Kingdom and drastically increased its audience. To distinguish itself with the "house" label, the name has been changed to "acid-house". It is strongly associated with the rave culture in the UK but also with drug use at all-night parties (Gilbert, Pearson, et al., 1999). As a musical characteristic, acid is much more synthetic than house with a more frenetic rhythm, instrument sounds and vocals are replaced by synthesized sounds.

In 1988 the "techno" label was created, so named by a recording company, and was carried by artists such as Derrick May and Juan Atkins in the city of Detroit. The "techno" term has been used to detach itself from the one of "acid" which was tainted by its link with drugs (Thornton, 1996). The term emphasizes the principle of technology at the service of music by exploiting the expressive power of drum machines and synthesizers. However, the dark rave party counter-culture continued to develop leading to the emergence of sub-genres such as hardcore in the early 1990s. This music has even faster beats and almost no instrumentation with dense sounds and poly-rhythmic.

The 90s were also marked by the advent of the jungle in 1992, a music influenced by the black and urban culture of the time. It is characterized by a much more complex rhythm than house, less repetitive and with a higher BPM. By 1994, some artists have abandoned this complexity and rather focused on simpler rhythms associated with *drum'n'bass*.

From that time on, sub-genres began to appear that were the antithesis of dance or rave music. These sub-genres had little or no beats and were intended to relax the listener in clubs and concerts. It includes "down-tempo", "ambient" and "chill-out".

From the beginning of the 70's and in parallel to the sub-genres mentioned above, hip-hop has also developed as being included in EDM. It is a rhythm-driven music that has influenced other genres of EDM just as much as it has been influenced by them. Its productions by "beat-makers" are also composed with synthesizers and reflect the urban culture of street dance. Hip-hop DJ's were the stars in the 70's but due to the popularity of the genre it became more like a recorded medium.

Much more extensive histories of the first twenty years of EDM can be found in the literature. (Reynolds, 2013) describes the entire history of EDM, its origin and its evolution over time. Shapiro and Lee, 2000 provides more detail for each sub-genre. (Brewster and Broughton, 2014; Fikentscher, 2000) study the role of the DJ in society. Finally, Butler focuses his analysis on the origins of EDM, particularly its geographical origins. In his paper, (McLeod, 2001) presents a brief history and describes the causes that led to the evolution of the taxonomy of the sub-genre of which it is composed.

Today, despite the fact that EDM is a fairly recent genre (i.e. less than 50 years of history), it has a tremendous number of sub-genres. According to the specialized journalists, the producers and the consumers, sub-genres evolve very quickly and new micro-genres[1] proliferate more than in any other musical style (Collins et al., 2013). This can be explained first of all by the fact that anyone can improvise himself as a composer from home with the constant evolution of sound technologies such as computer assisted music software but also machines to control them as well as synthesizers. Second, on online music shop, there are more than thousands new songs that are upload every single day. Those shops have to categorize the uploaded tracks into sub-genre to make it easier for the users to find what kind of sound they are looking for, before buying or listening to them. Third, this taxonomy is also influenced according to a marketing strategy of music broadcasting and recording companies but also of specialized media. Another aspect described by Vitos, 2014 is that it also obeys the requirements of DJs. The exact categorization into sub-genres allows them on the one hand to use similar tracks and on the other hand to inform the listeners about the style of musical performance they are attending.

Finally, all of these reasons imply that the taxonomy can be qualified as excessive. This over-categorization results in a taxonomy that is sometimes arbitrary, independent of precise rules. Genres and sub-genres are often defined beyond the musical aspects that characterize them and can therefore be redundant or ill-defined. We thus wish to focus our work on a purely musical automatic categorization.

1.1.3 *Electronic/Dance Music Musical Characteristics*

Certain musicological aspects allow the identification of sub-genres. As in all "meta-genres" such as rock or jazz (which also include many sub-genres), aficionados of the musical style will be the most likely to notice a difference between two

1 Or sub-sub-genres

tracks belonging to the same genre. And conversely, someone who is insensitive to musical style will tend to say: "it is always the same thing". A music listener will identify the audio features that characterize a piece of music while listening to it. These features will allow them to naturally get an idea of the musical genre to which it belongs.

A distinguishing feature of EDM is that it is generally instrumental (with the exception of hip-hop), singing voices are often sampled from other musical pieces. Beyond instrumentation, all the sub-genres of EDM have in common their structural organization based on repetitions. The essential primary unit of their structure is the *loop*, a short musical excerpt that is rhythmically aligned and repeated on different layers. It can be isolated in the track or combined with other loops. To summarize, a typical EDM track is composed by superimposing and assembling loops of different sizes on different musical layers.

The musical characteristics that allow to identify a sub-genre of EDM are mainly oriented on the rhythm and the beat. (Butler, 2006, p. 78) thus makes a distinction between two major rhythmic streams that encompass the main sub-genres of EDM: the *four-on-the-floor* and the *breakbeat-driven* styles. The *four-on-the-floor* sub-genres are influenced by the disco's steady bass-drum pattern evolving in techno or trance through house. The associated rhythm pattern is a steady, uniformly accented beat in which the bass drum is hit on every beat in a 4/4 meter. The *breakbeat-driven*, as we have seen in the historic of EDM, appeared during the exportation of house in UK with sub-genres like jungle or drum'n'bass. It is characterized by the presence of highly syncopated binary rhythms and the intense use of polyrhythms. In the words of Butler (2006): "breakbeat patterns exploit the irregularity with an emphasis on the metrical weak location". The tempo of the different sub-genres of EDM cover a very broad spectrum. It is nevertheless on average very fast from 120 to 150 BPM. The trip-hop, down-tempo or chill-out have a tempo of about 80 BPM and represents the "slowest" styles. The gabber (a micro-genre derived from hardcore) and some techno tracks can reach more than 220BPM (Butler, 2006). We give more details on the rhythmic characteristics of EDM main genres in Section 3.3. The separation into two main rhythmic categories motivated us to focus on this type of music.

Beyond the rhythm, an EDM track can be analyzed by its instrumentation on its other musical layers.

Regarding the four-on-the-floor styles, a house track will often have a very clear harmonic layer borrowed from other musical styles such as jazz, funk or pop music. This type of instrumentation also gives rise to new micro-genres (such as deep-

house). The trance also has this harmonic borrowing and has despite a very high BPM and supported beats a particular musicality (psy-trance). On the other hand, some styles such as techno have much more limited harmonic and melodic layers. Minimal, for instance, is free of any form of pitched material.

Breakbeat-driven styles are also more sparse in terms of harmonic content. These musics are oriented on the manipulation of samples interspersed with imposing basses and complex rhythms. The samples thus used are drum breaks but also vocals borrowed from sub-genres such as hip-hop or reggae and dub (reggae-jungle).

Through our work, we wish to demonstrate that rhythmic features analysis allows automatic categorization of EDM genres. For this we have developed in our work different methods based on the harmonic characteristics of the rhythmic structure. The two tasks mentioned, the estimation of tempo and the classification in rhythm-oriented genre, are two elements that allow us to illustrate the importance of rhythm in the description of EDM. In terms of application, tempo estimation is essential to the work of the DJ since tempo is the main reference of transition. It is also an integral part of EDM creation process, particularly with regard to computer-assisted music software.

1.2 DISSERTATION ORGANIZATION AND MAIN CONTRIBUTIONS

Chapter 2 of this manuscript is dedicated to the definition of the main concepts and the presentation of research from the state of the art related to our work. We thus describe in Section 2.2 the fundamentals related to the tasks we wish to perform, namely rhythm (including tempo) and the notion of musical genre. State of the art methods are presented under two categories. The first concerns so-called handcrafted methods based on knowledge of the field. It describes various features used for automatic tempo detection and genre classification. We detail that in **??**. The second concerns more recent data-driven methods, based on large-scale deep learning networks. We also introduce the basic concepts of deep-learning and specifically those we have used as tools to develop our methods in Section 2.4. Finally, in Section 2.5, we emphasize the different works of MIR linking EDM and rhythm.

In Chapter 3 we present the various datasets that we have used in order to evaluate the method we have developed. In Section 3.2, we describe in details the commonly used datasets annotated in tempo and/or in genre to perform the two tasks. Among these, few are specific to the EDM. This leads us to our first contribution in Section 3.3: the creation of two annotated datasets in EDM genre.

We propose to detail the musicological characteristics of the selected genres with a particular emphasis on their rhythmic structure.

In Chapter 4 we present the Deep Rhythm method developed to perform the two tasks of tempo estimation and rhythm-oriented genre classification. This method is based on the use of a representation of the periodic elements of rhythm at the input of a deep-learning network. For its development, we were inspired by various works that we describe in Section 4.2. In Section 4.3, we define the harmonic representation of the rhythm we have created, the HCQM. In Figure 4.6, we present the architecture of the CNN that we are training in order to achieve the two tasks. In addition, we describe different ways to exploit the temporal dimension of the HCQM. This is detailed in Section 4.5. Finally, we explain in Section 4.6 our evaluation protocols and the results obtained for tempo estimation and genre classification. For this purpose we compare the results obtained according to several parameterizations of the DR method with three state-of-the-art methods.

In Chapter 5 we present three extension of the Deep Rhythm method. The first one in Section 5.2 is the use of a complex-value HCQM as input of a complex convolution network to allow taking into account the inter-relationship between accoustic frequency bands. The second one in Section 5.3 is the use of a multitask learning approach in order to perform the two tasks simultaneously. The third one in Section 5.4 is the use of a multi-input/multi-branch network in order to add timbre information in addition to rhythm ones during the learning. For each of these proposed improvement, we describe our motivations, the method itself, its evaluation protocol and its results.

In Chapter 6 we present a genre classification method based on metric learning adapted to DR in order to analyze rhythm similarities between audio tracks. First, in Section 6.2 the principle of metric learning is exposed. Then in Section 6.3 we describe the triplet loss which we then adapt to the DR in Section 6.4. Finally, Section 6.5 is dedicated to the analysis protocol we have set up and comment on the results.

We conclude in Chapter 7 and propose future works.

2

FUNDAMENTALS AND STATE OF THE ART

2.1 INTRODUCTION

In this chapter, the fundamentals related to our work are exposed as base for the rest of this manuscript. We also present the state of the art methods related to automatic tempo estimation and music genre classification.

We begin by highlighting general concepts of music description. In Section 2.2, we define many of the rhythm aspects including the tempo. We also detail the concept of musical genre which arguably comes hand in hand with music description.

We then describe the different methods of the state of the art, specific to the field of MIR, which can automatically process the aspects of tempo, rhythm and musical genre. We make a distinction between the two types of systems that have been used for these purposes. On one hand, the *handcrafted* systems are based on signal processing schemes applied to music and sound. Features crafted from musical knowledge are extracted from the audio signal in order to describe as best as possible its aspects and are then classified thanks to statistical models. Such techniques for tempo estimation or genre classification are described in Section 2.3. On the other hand, *data-driven* systems rely on the use of machine learning (including deep learning algorithms) applied to large annotated databases as described in Section 2.4. If these two types of systems are illustrated chronologically in relation to each other – data-driven systems have indeed appeared more recently with the emergence of large annotated databases – they are now intimately linked in music research. In the upcoming sections, beyond the introduction of state of the art methods, we also present the tools that have been used in various research and that are somehow connected to our work. For the handcrafted methods, we insist on the features oriented towards the representation of rhythm components. For the data-driven methods, general and more specific machine learning concepts are presented to better understand the tool we will be manipulating in this thesis.

Finally, we present in Section 2.5 the different works in MIR related to EDM, whether for rhythmic analysis or for genre classification.

2.2 CORE DEFINITIONS

2.2.1 *Rhythm*

DEFINITIONS. One of the first known definitions of rhythm dates back to Plato (-350), who describes it as "order in movement". Both the passing of time (movement), and a presence of regularity, structure (order) were already exposed as two fundamental elements of the definition of rhythm. Rhythm in general can be defined in two complementary ways. It is a repetition of identical events in the course of time. This definition is applicable to any phenomenon with a temporal periodicity (e.g. from the heartbeat to the lunar cycle). In the case of music, this definition would concern the musical structure. For example, on the scale of an EDM track, we can observe the repetition of structurally identical (or really similar) parts. These parts correspond to different structure level, a kick/snare alternation at small level and a loop at higher level are common examples. On the other hand, it can be described as a non-isochronous sequence of events, but which nevertheless retains a certain regularity. This last definition is more specific to describe the "artistic" rhythm whose events that compose it respond to a temporal logic (dance, sound or silence in music, rhymes for poetry, etc.).

Some descriptions from the state of the art show that rhythm cannot be precisely defined (according to Fraisse (1982)) but at least has many relative definitions. Honingh et al. (2015) define rhythm as a very general concept like the way that elements are ordered in time. For Fraisse (1974, p107) the perception of rhythm results from the perception of structures combined with their repetitions. According to Cooper, Cooper, and Meyer (1963), to experience rhythm is to group separate sounds into structured patterns. Some of these definitions are closer to the realm of perception. According to Schubotz, Friederici, and Von Cramon (2000), Grahn and Brett (2007) or Chen, Penhune, and Zatorre (2008), rhythm can also be seen from a sensory-motor point of view in music as the dancing element, the one that makes us want to move. That is the case of EDM for instance.

RHYTHM IN MUSIC. We keep as a definition of rhythm in music the one made by (Gouyon et al., 2005) and illustrated in Figure 2.1. In the most general way, rhythm represents all the temporal relations and information in music. It includes metrical structure, timing and tempo. We define these three elements in more detail in the next paragraphs.

2.2.1.1 *Metrical structure*

The Generative Theory of Tonal Music (GTTM) is a theory proposed by Lerdhal and Jackendoff (1983) at the origin of many works in the field of music theory

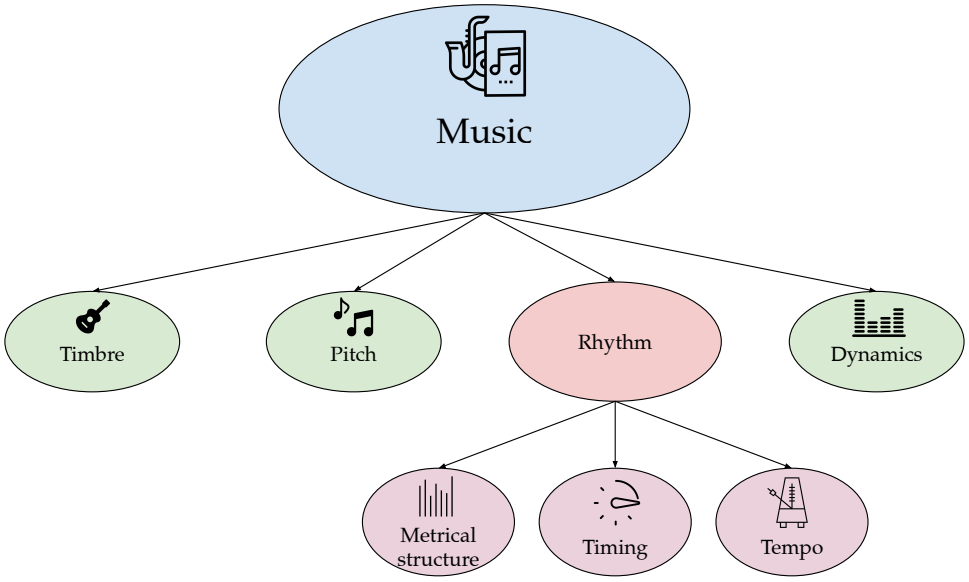

Figure 2.1: Rhythm in music.

and in the perception of music. It describes the way in which a listener, composer or producer builds an understanding of music. The music is seen as an infinite set constrained by a finite set of rules. This is also the case in other fields such as "grammar",since a sentence construction is infinite but limited by grammatical constraints. In this case, we can speak of "musical grammar", it is an abstract representation of the score. According to Lerdhal and Jackendoff (1983), it focuses on hierarchical systems that create the musical intuitions, including grouping structure (phrasing) and metrical structure that are encompassed in the rhythmic structure. The grouping structure is linked to temporal durations while the metric structure is linked to punctual elements in time: the **beats**.

BEAT. Beat and pulse are usually described as the same element. Cooper, Cooper, and Meyer (1963) define a pulse as "one of a series of regularly recurring precisely equivalent stimuli. [...] Pulses mark off equal units in the temporal continuum". Gouyon et al. (2005) uses the term pulse as a metric level (a series of beats) and the term beat as a single element of a pulse. The beats are grouped in series according to their respective accentuation. Accentuation is at the basis of any rhythm, once again according to Cooper "accent is a stimulus (in a series of stimuli) which is marked for consciousness in some way". In other words, accentuation is the human being's ability to identify events when listening to musical excerpts. The way it is perceived is still not well understood but much research on the subject tends to conclude that it is influenced by several elements: harmony, duration, intensity,

regularity, pitch or the perception of timbre (Cooper, Cooper, and Meyer, 1963; Lerdhal and Jackendoff, 1983; Dixon and Cambouropoulos, 2000). In the GTTM, Lerdhal and Jackendoff (1983) cites different rules to which the beat obeys:

- They must be equally spaced.
- There must be a beat of the metric structure for each note.
- The division of a series of beats by a specific time duration corresponds to a metric level. This allows one to identify low metric levels (short time duration) and high metric levels (long time duration).
- A high level beat must be aligned with a beat at the lowest level, so a high level beat aligned with beats at the next highest level is called a downbeat. For a given beat series at a specific metric level, the distance between two consecutive beats is called the inter-beat interval, which is the period of the series. Its phase is represented by the position of the first beat.

TATUM. Tatum is one of the many names used to describe the smallest level of the metrical structure. However, according to Bilmes (1993), a better definition would be that the **tatum** is "the regular time division that most highly coincides with all note onsets" (it is defined in details in Section 2.3.2) The tatum is often at the eighth or sixteenth note level. In a very syncopated music track it is not necessarily explicit and can be simply induced (Gouyon et al., 2005).

MEASURE. In a music score, the measure refers to the bar line restricted by a meter of two levels. It is associated with the time signature which represents the number of faster beat that make up one measure. Time signature is indicated as a fraction: the denominator corresponds to the basic temporal unit and the numerator to the number of basic temporal units a measure can contain. For instance for a $\frac{2}{4}$ time signature, the basic temporal unit is a quarter note and in a measure there is room for two of them. The GTTM specifies that there must be a beat of the metric structure for each note, quantized durations are used to match these sound events to a score (in Western notation). They are rational numbers (1, $\frac{1}{4}$, $\frac{1}{6}$, ...) relative to a chosen time interval (the denominator of the time signature). The use of quantized duration simplifies the reading of the rhythmic structure of a music track.

DYNAMIC ATTENDING THEORY. A theory other than the GTTM, the Dynamic Attending theory (DAT) proposed by (Jones and Boltz, 1989), shows the importance of the context of perception of a temporal event. Indeed, according to the DAT, each event will be anticipated by the listener according to the events already passed. It thus summarizes the attention mechanism of the brain which will pro-

duce the minimum effort possible in order to minimize the duration of attention. In the case of sound, when listening to a piece of music, the attention will be amplified at the moment an event occurs, such as a pulse or the presence of a downbeat, and will diminish the rest of the time. This finally shows that the human being is capable of perceiving the musical structure. However, it is also shown that the musical education (Drake and Palmer, 2000) of a listener will strongly influence his ability to analyze the rhythmic structure.

2.2.1.2 *Tempo*

The tempo represents generally the speed of a musical excerpt. It is defined as the rate of the beat at a given metrical level in a given metrical structure.

TACTUS is the name given to the tempo of the most intense pulsation or beat. It is the metric level that corresponds to the tempo. Cooper defines it as one of a series of identical, regular and recurring stimuli. First indicated on musical scores in an informative way, it is from the appearance of the metronome in the 19th century that the composer was able to precisely quantify the tempo of one of his compositions. It is measured in beats per minute (BPM). One of its definitions is purely perceptive (Scheirer, 2000) and corresponds to the speed at which a listener will clap his hands, on an object or with his feet while listening to a piece of music.

PREFERED AND PERCEIVED TEMPO. Tempo is still today a concept that can be described as ambiguous. Indeed, depending on a given piece of music it can be perceived differently by several listeners and also by one and the same listener. McKinney and Moelants, 2004; McKinney and Moelants, 2006 highlight the use of a resonance model to identify the reasons for the appearance of a preferred tempo. Perceptual experiments in which listeners were asked to "tap" the tempo of a piece of music showed that the preferred tempo was around 120BPM (Moelants and McKinney, 2004). If one of the metric levels contains musical excerpts containing this resonant tempo then listeners will tend to estimate the same tempo. On the other hand, if an excerpt contains a metric level that has two beats on either side of the resonance tempo then the listeners will not be unanimous on their estimation. To date few methods have been developed for the estimation of a perceptual tempo and those despite more recent experiments (Zapata et al., 2012; Holzapfel et al., 2012; Peeters and Flocon-Cholet, 2012; Peeters and Marchand, 2013). Differences in tempo perception correspond in the majority of cases to an estimate in a different metric level. It can thus be estimated at half or double the inter-beat interval or at a third or triple of the inter-beat interval. This is referred to as **octave error**, a problem that is being actively studied in the field of tempo estimation, with the aim of reducing it.

2.2.1.3 *Timing*

In all styles of music, the timing of the metric structure does not have a fixed starting point and presents certain variations that are not taken into account by the GTTM. There are two types of non-metric timing, one based on tempo, the other on timing. Starting from a series of isochronous beats placed in a strictly metric way, we can differentiate between two types of timing deviations:

- The shift of a single beat of the metric structure. We speak of systematic deviation such as the "swing" that has its origin in jazz music (Friberg and Sundström, 1997).
- The offset of all the beats following a particular beat in the series. In the case of a piece of music, it may occur after a pause. This is called expressive deviation used to accentuate the emotion of a listener.

In both cases, the timing change does not affect the inter-beat interval, it is a short-term change also called "timing deviation". The change in tempo is characterized by the variation of the inter-beat interval from a particular beat of the series, it is a long term change. By taking the series in real time, it is impossible to predict these changes before the shift occurs. According to (Honing, 2001), it is impossible to represent them mathematically because they are two-dimensional elements projected onto one, time. From another musically point of view, one can interpret these variations on the assumption that the tempo is constant and that the deviations are temporal and local. These deviations are an integral part of the structuring of the rhythm and are clearly defined by the musicians or the composers (Honing and De Haas, 2008).

2.2.2 *Musical genres*

As with rhythm, there is no clear and formal definition of the musical genre. According to Tzanetakis and Cook (2002), the musical genre is defined as a label created by humans to describe and characterize music. In a way, all the constitutive elements of music illustrated in Figure 2.1 are implicated in the genre description of an audio track. Historically, genre is intrinsically linked to classification: it responds to this natural human necessity to categorize what surrounds him. The categorization of music is a vague field that is extremely subjective and lacks general rules. In everyday life, the genre remains the most used and fastest way to categorize or to compare pieces of music. For example, when one talks about a concert to someone who does not know the artist, one will specify the music genre in order to give an idea of the kind of music one listens to. This is why, with

increasing popularity over the years, music genre classification is today one of the most studied tasks in MIR.

(Aucouturier and Pachet, 2003) present the different aspects of the musical genre concept in their study. In particular, they put forward two concepts that surround the utility of genre. The "intentional" concept where genre is an interpretation of a music track and give information on a community culture, on the particular epoch and on the geography. The "extensional" concept where genre represents a set of music track in an analytic way (e.g. the emotion brought by the piece).

In this thesis, we perform a music genre automatic classification. The genre that we use are defined according to their musical characteristics among the other aspects (cultural, marketing, etc.). The definition of the musical genre being intrinsically linked to its automatic analysis, the meaning of the term becomes more explicit by exploring the various works dedicated to this task. With the specific focus on EDM, we seek to reconnect the genre of a piece with its rhythmic structure.

2.3 HANDCRAFTED SYSTEMS

Early MIR systems encoded domain knowledge (audio, auditory perception and musical knowledge) by hand-crafting signal processing and statistical models. Data were at most used to manually tune some parameters (such as filter frequencies or transition probabilities). Starting from digital signal processing fundamentals, we present here the different knowledge-based tools used for rhythm description, tempo estimation and genre classification. We also present the main works dedicated to these tasks. It should be noted that these are among the most explored in the field. To go into details or to have a more precise vision of the state of the art concerning them, good overviews have been published: (Gouyon et al., 2006; Zapata and Gómez, 2011; Peeters, 2011) for rhythm description and (Aucouturier and Pachet, 2003; Ramírez and Flores, 2019) for music genre classification.

2.3.1 *Signal Fundamentals*

We only present here the main elements of signal processing that we have used in this thesis. For a more detailed description, many reference books or articles can be consulted such as (Rabiner and Gold, 1975) and more recently (Muller et al., 2011) for a specific application to music.

DISCRETE-TIME SIGNAL The basic element of any computer music operation is the audio signal. It is the signal that contains the acoustic information that we perceive as sound. It allows the transmission, the storage, the creation and, in the case of MIR, the analysis of music. Music is a continuous-time audio signal. That is, for a given signal, there is a function $f : \mathbb{R} \to \mathbb{R}$ such that each point in time $t \in \mathbb{R}$ has an amplitude $f(t) \in \mathbb{R}$. Due to the inherent constraints of digital systems, the signal cannot be processed in its continuous form. It has to be discretized (i.e. converted to a finite number of points). To do so, two operations are applied to the audio signal: sampling and quantization. It can be done using equidistant sampling (or T-sampling), given a continuous-time signal f, and a positive real number $T > 0$, the discrete-time signal x can be define as a function $x : \mathbb{Z} \to \mathbb{R}$ following:

$$x(n) := f(n \cdot T) \tag{2–1}$$

for $n \in \mathbb{Z}$. Here T is the sampling period, its inverse is computed to obtain the sampling rate in Hertz (Hz):

$$F_s = 1/T \tag{2–2}$$

According to the Nyquist-Shannon-Kotelnikov theorem, a sampling rate $F_s = 44,100\text{Hz}$ is high enough to perfectly represent sounds with frequencies up to the Nyquist frequency: $\Omega = \frac{F_s}{2} = 22,050\text{Hz}$. Human auditory system is able to perceive frequency ranges from 20 to 20,000Hz. Beyond a sampling rate of 44,100Hz, the gain in sound quality is no longer perceptible.

DISCRETE FOURIER TRANSFORM The Discrete Fourier Transform (DFT) is a digital signal processing tool that allows to decompose a discrete-time signal into its constituent frequencies (i.e. the spectrum). Given a discrete-time signal x of length L, the DFT X of x is defined as:

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-\frac{2\pi i}{N} kn} \tag{2–3}$$

for $k \in [0 : N-1]$. $X(k)$ is a complex Fourier coefficient that encodes the magnitude and the phase of the sinusoidal components of the signal with frequency:

$$F_{coef}(k) = \frac{k \cdot F_s}{N} \tag{2–4}$$

The frequency resolution (the distance between frequency bins) is expressed as $\Delta_f = F_{coef}(1)$.

SHORT-TIME FOURIER TRANSFORM In a research field like MIR, we need to represent the time evolution of the frequencies, the DFT does not allow to do that. The Short-Time Fourier Transform (STFT) is used to apply the DFT on short parts of the signal. These time segments are known as frames. In other words, the STFT allows to represent which frequency occurs in a given frame. It is defined as:

$$X(t,k) = \sum_{n=0}^{N-1} w(n) \cdot x(n+tH) \cdot e^{-\frac{2\pi i}{N} kn} \tag{2–5}$$

with $t \in \mathbb{Z}$ the frame index, $k \in [0 : L-1]$ the frequency index, w a window function centered around time position, N the frame length and H the hop size (i.e. the distance between two consecutive frames in samples). Here k corresponds to the frequency band with center frequency:

$$F_{coef}(k) = \frac{k \cdot F_s}{N} \tag{2–6}$$

The spectrogram (frequency/time representation) refers to the magnitude of the STFT: $Y = |X|$.

2.3.2 *Rhythm handcrafted features*

A classic flowchart of commonly used automatic analysis of rhythm system is presented in Figure 2.2

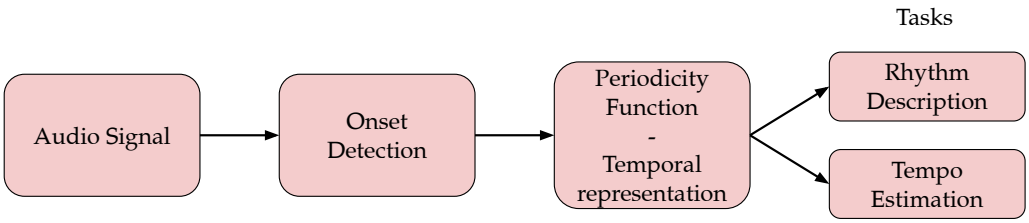

Figure 2.2: System of automatic analysis of rhythm.

2.3.2.1 *Onset*

Onsets denote the start time of events in the audio signal. Finding the accents of the musical signal is equivalent to finding its onsets[1]. Klapuri (1999) defines it as

1 The onset differs to the beat since it is a physical phenomena (versus perceptual one for the beats).

"the start of a discrete event in an acoustic signal" while Bello et al. (2005) define it as the chosen moment to mark a transient (i.e a short time interval in which the signal evolves rapidly in a non-trivial way). Thus according to the ADSR model, the onset is identified as the start time of a note, drum beat or other musical event. As illustrated in Figure 2.3, it is located at the beginning of the attack, it marks the earliest point in time where the transient can be detected.

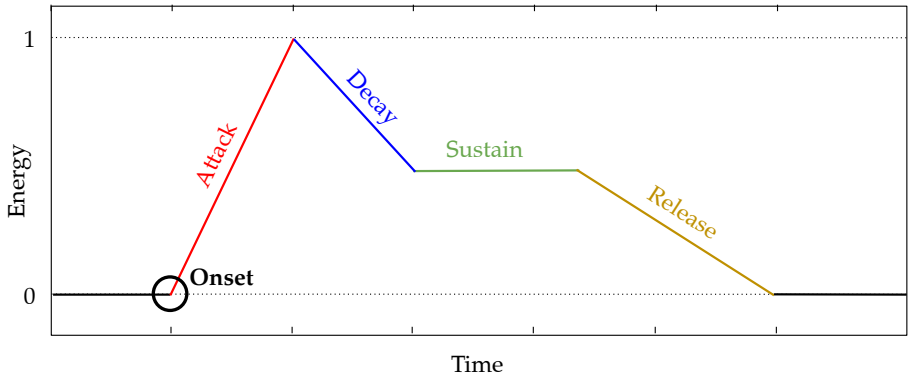

Figure 2.3: Schematic of the "ideal-case" ADSR for a given note.

2.3.2.2 *Onset detection*

The ADSR illustrated in Figure 2.3 is representative of an ideal case. In the musical context, where the signal is polyphonic, complex and noisy it becomes difficult to identify the onsets. An onset detection function is used as an intermediate representation to describe the changes in quantity or strength that occur in the signal from one moment to another. Such functions may have several denomination in the literature, such as Onset Signal Strength (OSS) or Onset Energy Function (OEF). We use the latter in this manuscript. Three steps are to be considered to perform an onset extraction of a musical signal:

1. Signal pre-processing such as multi-band separation;
2. Computation of the OEF for example spectral flux (consisting in temporally differentiating the amplitude of the spectrogram) or the amplitude-weighted phase deviation (Bello et al., 2004; Dixon, 2006);
3. Extraction of onset by searching for peaks in the OEF often performed by a search for local maxima.

Analyzing the signal over multiple frequency bands logarithmically spaced allows to better detect onset according to Scheirer (1998) which combined these an-

alyzes in a second step. This idea has been taken up by Klapuri (1999) and Paulus and Klapuri (2003) and Dixon, Pampalk, and Widmer (2003) among others.

2.3.2.3 *Temporal representations*

Analyzing the rhythmic structure often means studying the periodicity of events related to the rhythm, and in this case the onsets. Also to estimate the tempo from an OEF it is necessary to determine its dominant pulse i.e. the periodic element with the highest energy. The comb filter bank, the DFT and the Auto-correlation Function (ACF) are periodicity representations while others representation such as similarity matrix, scale transform and modulation spectrum also allows to represent rhythm.

COMB FILTER BANKS. Scheirer (1998) proposes the use of band-pass filters combined with resonating comb filters and peak picking to estimate the dominant pulse positions and so on the tempo. Klapuri, Eronen, and Astola (2006) also use resonating comb filter banks driven by band-wise accent signals The main extension they propose is the tracking of multiple metrical levels.

DFT. The DFT (presented in Section 2.3.1) has been also used for tempo estimation (Xiao et al., 2008; Grosche and Müller, 2009) It has been applied to the OEF in (Holzapfel and Stylianou, 2011; Peeters, 2011) or to others representation (Klapuri, Eronen, and Astola, 2006) for rhythm description.

ACF. ACF is the most commonly used periodicity function. It is calculated as follows:

$$ACF(m) = \sum_{n=0}^{N-1} x(n) \cdot x(n-m) \tag{2–7}$$

It has been used for beat tracking in (Goto, 2001; Seppänen, Eronen, and Hiipakka, 2006; Davies and Plumbley, 2007; Ellis, 2007) and by extension for tempo estimation in (Percival and Tzanetakis, 2014). Gainza and Coyle (2011) propose a hybrid multi-band decomposition where the periodicities of onset functions are tracked in several frequency bands using auto-correlation and then weighted. In (Peeters, 2006), a combination of DFT and ACF is applied in order to avoid octave errors.

For rhythm description, Tzanetakis and Cook (2002) defines a beat histogram computed from an ACF and Peeters (2011) proposes a harmonic analysis of the rhythm pattern using again a combination of DFT and ACF.

OTHERS. This is probably due to the difficulty of creating datasets annotated in such rhythm pattern (defining the similarity between patterns — outside the trivial identity case — remains a difficult task). The recognition of rhythm pattern has received much less attention. To create such a dataset, one may consider the equivalence between the rhythm pattern and the related dance (such as Tango). This approach led to the dataset: *Ballroom* (Gouyon et al., 2006), *Extended-ballroom* (Marchand and Peeters, 2016b) and *Greek-dances* (Holzapfel et al., 2012). Foote, Cooper, and Nam (2002) defines a beat spectrum computed with a similarity matrix of MFCCs, Holzapfel and Stylianou (2011) proposes the use of the scale transform (which allows to get a tempo invariant representation), Marchand and Peeters (2014) and Marchand and Peeters (2016a) extends the latter by combining it with the modulation spectrum and adding correlation coefficients between frequency bands.

To estimate tempo or describe rhythm, other representation have been developed such as wavelet representation (Tzanetakis and Cook, 2002), tempogram (Cemgil et al., 2000; Wu and Jang, 2014) or inter-onset interval histograms (Seppanen, 2001; Gouyon and Herrera, 2003)

2.3.3 *Musical genre handcrafted features*

Early automatic genre classification systems were based on twofold procedure. A step of handcrafted feature extraction and a step of classification (relying on a machine learning algorithms). For feature extraction, a vector of low-level descriptor is computed on an audio signal cut into frames using STFT (Section 2.3.1). As mentioned earlier, the classification of the musical genre is based on descriptors related to the characteristic concepts of music: the rhythm, the timbre and the pitch. We introduced numerous features related to rhythm in the previous section We present below an exhaustive list of timbre and pitch related features devoted to automatic genre classification.

TIMBRE RELATED FEATURES. The features related to timbre are the most used for automatic genre classification because they are dedicated to the spectral distribution of the signal. In other words, timbre features extracted for every frames encompass the sources (instrument) in the music. Among those, the MFCC which working as approximation to human auditory system are the most used (Tzanetakis and Cook, 2000; Pye, 2000; Deshpande, Nam, and Singh, 2001). Other works use spectral centroid (Tzanetakis and Cook, 2002; Lambrou et al., 1998), spectral flux, zero crossing rate, spectral roll-off (Tzanetakis and Cook, 2002),

PITCH RELATED FEATURES. Ermolinskiy, Cook, and Tzanetakis (2001) use pitch histograms features vector to represent the harmony of an audio signal. Wakefield (1999) developed a "chromogram" that describes the harmonic content of the music and can be used to determine the pitch range of the audio signal. Chroma features are the main features related to pitch. They enable the modeling of melody and harmony assuming that humans perceive different pitches as similar if they are separated by an octave (Müller, 2015).

For good overviews of automatic genre classification, early methods refer to (Aucouturier and Pachet, 2003) where the extracted features are described in details. A more recent overview of these methods is presented in (Ramírez and Flores, 2019). They also present all the works that used automatic features learning and deep learning.

In the next section, we describe deep learning as a tool for data-driven systems.

2.4 DATA-DRIVEN SYSTEMS

Data-driven systems use machine learning to acquire knowledge from medium to large-scale datasets. A step of feature extraction from the data is often applied upstream of the training step. It makes the data more suitable for learning depending on the task to be learned. In this section, we first present the machine learning fundamentals. We only describe the tools used in this thesis. Then we list some of the works using machine learning and deep learning to perform the task of automatic tempo estimation and genre classification.

2.4.1 *Machine learning fundamentals*

Machine learning is a research field that, as its name suggests, encompasses the computer algorithms designed to learn. Contrary to a classical algorithm, developed according to a certain logic, machine learning methods are expected to learn this logic from the data they process. For a given task, they automatically target the patterns contained in the data.

Supervised learning is a paradigm in machine learning that uses labeled data and where the desired output for a given input is specified. It is opposed to *unsupervised learning* for which data labels are not available. Other so-called *semi-supervised learning* methods can also be used if the data collection is partially labeled. The methods we have developed in this thesis (Chapter 4, 5, 6) all rely on supervised learning.

2.4.1.1 *Supervised learning*

PRINCIPLE. A labeled data represents an input data x_i tagged by the output response y_i that we want the model to find automatically. The learning step of the model is then to identify the patterns in the input data that lead to the desired output. Thus, we want to find a function f_S that given a set of input/output pairs $S = (x_1, y_1), (x_2, y_2), \ldots, (x_{|S|}, y_{|S|})$ captures the relationships between x and y using controllable parameters θ:

$$f_S(x_i, \theta) = \hat{y}_i \approx y_i \tag{2–8}$$

where $\hat{y}_i$ is the output of the model. The training pairs (x_i, y_i) are drawn from a unknown joint probability distribution $p(x, y)$. The goal is to approximate $p(x, y)$ thanks to f_S by knowing only S and adjusting the parameters θ. Thus, based only on the prior knowledge S, the main objective of a trained model is to take unseen input data and correctly determined its output with Equation 2–8. This process is commonly called *prediction*. With the annotated dataset available, it is common to assume that the annotations are relevant and therefore that y_i is the correct label of the input x_i. However these annotations, often assigned manually, are subject to potential errors as we will see in the Chapter 3.

VARIANCE. Having access to the labels of the data allows to evaluate a trained model by computing its accuracy. However, this does not necessarily reflect real-world performances. The *variance* error measures the variation in performance for the different sets that we can draw from p(x, y). The variance decreases as the size and representativity of the training data S increases. It is formalized as:

$$Variance = E[(f_S(\theta] - E[f(\theta)])^2] \tag{2–9}$$

where $E[f(\theta)]$ is the expected performance of the model definition function $f(\theta)$ across all possible draws from $p(x, y)$ and $f_S(\theta)$ the actual performance on S.

BIAS. The bias describes how close $f(\theta)$ is to the unknown function $f*$ that best describes the joint probability distribution $p(x, y)$. It is formalize as:

$$Bias = E[f(\theta)] - f* \tag{2–10}$$

The goal of any supervised learning model is to achieve low bias and low variance. Thus the perfect model would be able to use an infinite amount of training data in order to eliminate bias and variance errors.

LOSS FUNCTION. To adjust the complexity of the model θ, a *loss function* $\mathcal{L}$ is defined. It measures the relationship between the model output $\hat{y}$ and its ground truth y. This error function (also known as the cost function) thus allows to evaluate the performance of the model by comparing y to $\hat{y}$. The choice of $\mathcal{L}$ depends on the task performed and thus on the characteristics of the target space. The cost function $\mathcal{L}$, being the total error of variance and bias, is minimized over the training set S by adjusting θ. Since the target joint probability $p(x, y)$ is unknown, the contribution of each term (bias and variance) cannot be calculated. Also, bias and variance evolve in opposite direction as it is illustrate in Figure 2.4 on which the *bias-variance trade-off* is represented.

UNDERFITTING, OVERFITTING AND SPLIT. As depicted in Figure 2.4, a "simple" model that oversimplify the relations between x and y tends to have low variance and high bias. This phenomenon is known as *underfitting*. On the other side, a more "complex" model (with large amount of θ) tends to have high variance and low bias. It is known as *overfitting*. The performance of the model will also depend on the training set S. If the complexity is high and S is small, the model will tend to overfit i.e. it will learn a function that depends only on the data from S without learning a trend or a structure present in $p(x, y)$. In other words, the model will generalize poorly.

In order to control the evolution of the bias-variance trade-off, the available data S is split into three sets:

- a training set that will be used to train the model;
- a testing set to evaluate the performance of the model.
- a validation set that will allow to find the most optimal model complexity (i.e. with the minimum error).

2.4.1.2 *Deep Neural Network*

PRINCIPLE. Inspired by a simplified version of the brain, Deep Neural Networks (DNN) are today the most widely used models of machine learning algorithms. Neuronal connections in the brain allow the passage and analysis of information from one neuron to another. DNN are based on the transfer of information through successive non-linearity operations. The basic element is a *neuron*, it is illustrated in Figure 2.5 and is formalized as:

$$h(x) = \sigma(Wx + b) \tag{2–11}$$

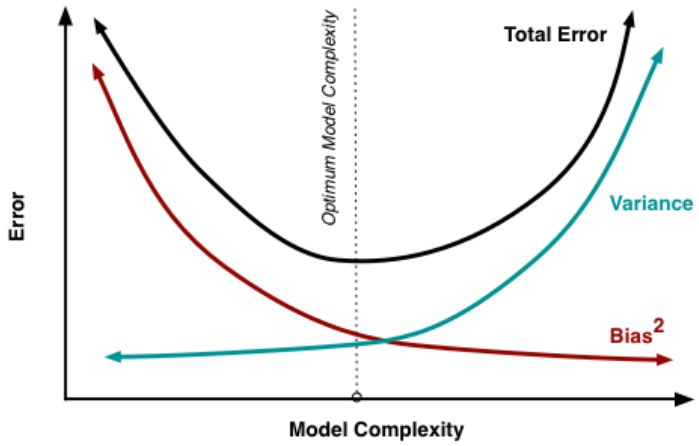

Figure 2.4: Bias and variance trade-off contributing to total error. *Figure taken from (Fortmann-Roe, 2012).*

where σ is a non-linear activation function, $x \in \mathbb{R}^D$ is the input of the layer, $W \in \mathbb{R}^{1xD}$ the weight matrix and b the bias vector. While W perform a linear transformation of the input data x, bias b allows the model to represent patterns that do not necessarily pass through the origin. The latter two represent the parameters that the model must learn during training. The activation function σ allows to induce non-linearity to the network in order to learn complex relationships between input and output.

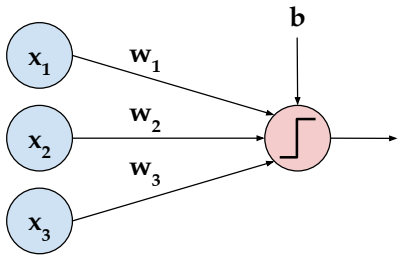

Figure 2.5: Flowchart of a single neuron non-linear operation, vector to scalar, applied to an input vector.

FULLY-CONNECTED ARCHITECTURE. A DNN is successively composed of an input layer x, hidden layers taking as input the outputs of the previous layer and an output layer. The role of the hidden layers is to capture the complex relationships between the dimensions of the previous layer. To do this, each neuron of one layer is connected to all neurons of the previous and applies its own activation function. This is called a *fully-connected* architecture (or dense layer).

BACKPROPAGATION. In a DNN, to predict $\hat{y}$, the input data are forward propagated through the successive layers to the output layer. Training the model is equivalent to adjusting the θ parameters to minimize a cost function $\mathcal{L}$. This operation is performed using Backpropagation (Rumelhart, Hinton, and Williams, 1986). The Backpropagation algorithm computes the gradient of $\mathcal{L}$ by measuring the deviation between the output $\hat{y}$ and y with respect to θ. In other words, backprofiling reflects the impact of each parameter on the overall behavior of the network. Thus, the model error is back-propagated from the output to the input of the network. The parameters θ are then updated using gradient descent, which requires the calculation of the partial derivative of the loss function $\frac{\partial \mathcal{L}}{\partial \theta}$. The negative of the gradient represents the steepest slope of the error surface. Hence, minimizing the function means going down in the direction of this slope towards a minimum error value. The most commonly used method is the Stochastic Gradient Descent (SGD). It updates, at any step t, each parameter (so called weights) by subtracting the loss gradient, calibrated by the learning rate η and is formalized as:

$$\theta_{t+1} = \theta_t - \eta \nabla \mathcal{L}(\theta_t) \tag{2–12}$$

This operation is called the gradient step in the error surface. The steps of forward propagation, Backpropagation and weights update are illustrated in Figure 2.6.

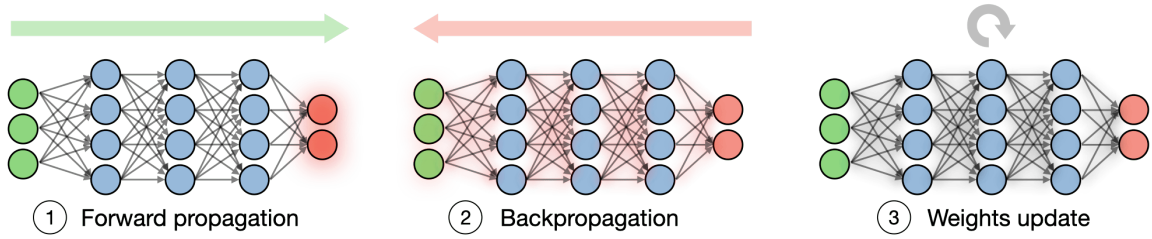

Figure 2.6: Updating weights in a fully-connected neural network. Input layer is represented in green, hidden layers in blue and output layer in red. *Figure taken from (CS 230 - Deep Learning).*

MINIBATCH. Due to the limited computational resources and processing capacity of a machine but also to the processing of the training data by the DNN model, the training set is rarely used in its entirety during an iteration. It is split into smaller subsets called *minibatch*. The size of a minibatch, commonly called batch-size, is correlated to the learning rate (which also depends on the type of S). The minibatch makes the objective change stochastically at each optimization iteration: with a small batch-size and a large learning rate, we deviate far from the

desired minimum error value and if the learning rate is too small we never reach this value.

NON-LINEARITIES ACTIVATION FUNCTIONS. The architecture of a DNN is configurable. The number of neurons restricts the number of input combinations while the number of layers is correlated with the model's ability to find hierarchical transformations. The more layers there are, the more the abstraction of representations from one layer to the next increases. These parameters have a direct influence on the complexity and thus on the bias-variance trade-off. A complex model with many layers and many neurons will tend to speed up the training process but will also tend to overfit quickly. To avoid this but also to help the model finding complex relationships, the non-linearity σ is applied to each neuron. The most commonly used are the rectifier non-linearities illustrated in Figure 2.7. They prevent the vanishing gradient problem i.e. when the loss function gradient is close to zero the network has difficulty to train. They are also efficient and help producing more sparse representations (Nair and Hinton, 2010; Clevert, Unterthiner, and Hochreiter, 2016).

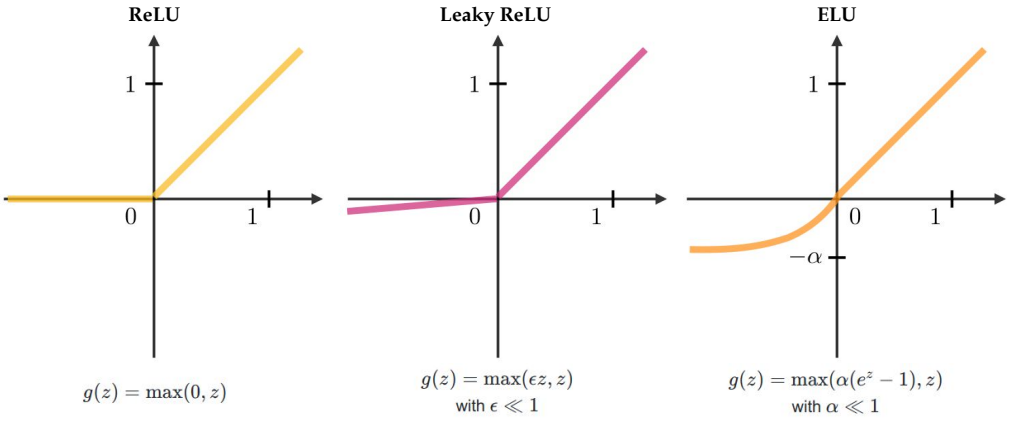

Figure 2.7: Rectifier non-linearities activation functions. Rectified Linear Unit (ReLU) is a non-linear function with a threshold (value below zero are set to zero). Leaky ReLU and Exponential Linear Unit (ELU) functions represent a non-zero derivate for negative values. *Figure taken from* (CS 230 - Deep Learning).

2.4.1.3 *Convolutional Neural Network*

PRINCIPLE. Fully-connected DNN are not appropriate to process input like images. Indeed, to process a three-dimensional image (width, height, color channel), the whole set of pixels must be considered, which drastically increases the number of parameters to process. As we have seen, too much complexity quickly leads

to overfitting. It is with the objective of directly processing images as input to a network that the CNN were developed (LeCun, Bengio, et al., 1995). They thus allow, based on the same learning process, to effectively reduce the number of parameters to be updated during training.

During image processing by a CNN, the input image can be seen as a three-dimensional function $\mathrm{img}(w, h, d)$ with w, h and d spatial coordinates. A pixel corresponds to a coordinate and its intensity corresponds to the amplitude $\mathrm{img}(w, h, d)$. The information of an image is then processed according to a given pixel and those surrounding it.

CONVOLUTION. In the field of image processing, the analysis process is based on the use of *filters* (also called kernels). They allow to detect the characteristic patterns of the image in a spatialized way. To do this, the filter is applied to the whole image by convolution i.e. it is slid over the whole image. As described in Figure 2.8, the convolution operation is equivalent to calculating an element-wise multiplication between each component of the filter and the input on which it overlaps. This results in a matrix called *feature map* that captures the applications of the filters over the entire input image i.e. it stores the filter activations. The filter is also three-dimensional. The height and width of a filter is smaller than the input image in order to be able to capture "local" information. However, the depth of the filter corresponds to the depth of the input. The stride parameter corresponds to the filter advancement step after each operation in terms of number of pixels for each dimension.

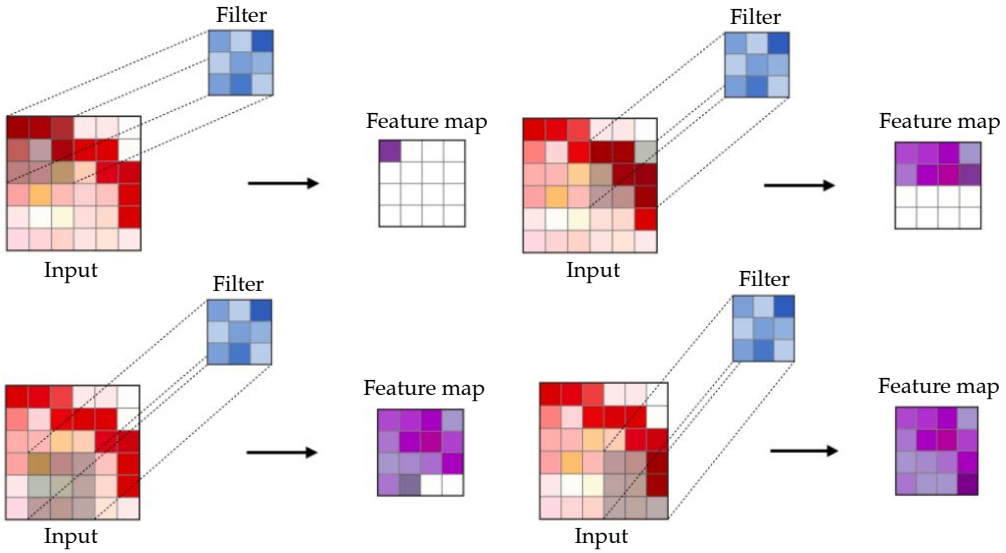

Figure 2.8: Convolution operation steps. *Figure taken from (CS 230 - Deep Learning).*

Another type of filter called *pooling* is usually applied after a convolution operation. Pooling is equivalent to a subsampling operation, i.e. the size of the feature map (only height and width) is reduced and only the most important features are kept. For example, among the pooling strategies, max pooling allows to keep the maximum values as shown in Figure 2.9.

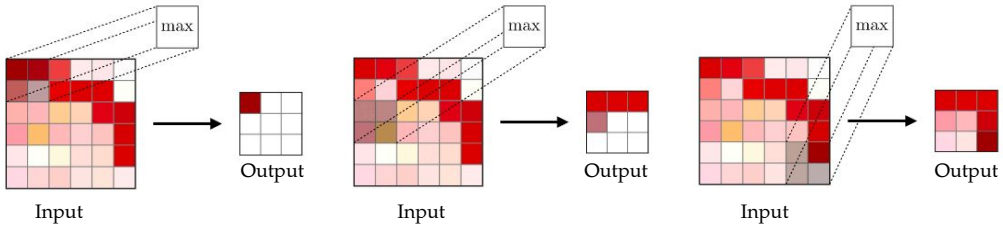

Figure 2.9: Max-pooling operation steps. *Figure taken from (CS 230 - Deep Learning).*

ARCHITECTURE. As for the DNN architecture, a CNN is composed of several successive layers. Each layer is generally composed of a convolution operation, a non-linearity operation and sometimes a downsampling phase successively. At a given layer, many different convolutions are processed in parallel, each one with different filters. The specificity of a CNN, compared to others image processing systems, is that filters are not handcrafted but they are learned during the training process using the Backpropagation algorithm. The weights correspond to the filters' value that are trained to detect important features automatically without human supervision. After a convolution operation, the resulting feature maps are stack along the depth axis as a tensor. Each convolutional layer learns filters of increasing complexity while adding many layers increases the abstraction capacity of the network. The first convolution layer extract low-level features such as edges, lines or corners. The middle layers learn filters that detect part of the object such as a wheel in the case of a car. The last layers learn higher-level representations such as the full object in different shapes and positions.

For a classification task such as automatic musical genre classification, it is usual to add several dense layer after the last convolutional layer in order to learn pattern from high-level features. A *flatten* layer makes it possible by reducing the tridimensional output of a convolutional layer to a one-dimensional vector. In this case, the last fully-connected layer represents all the possible classes. An overall CNN architecture for a classification task is presented in Figure 2.10.

All the methods we present in this thesis are based on the use of CNN to perform classification.

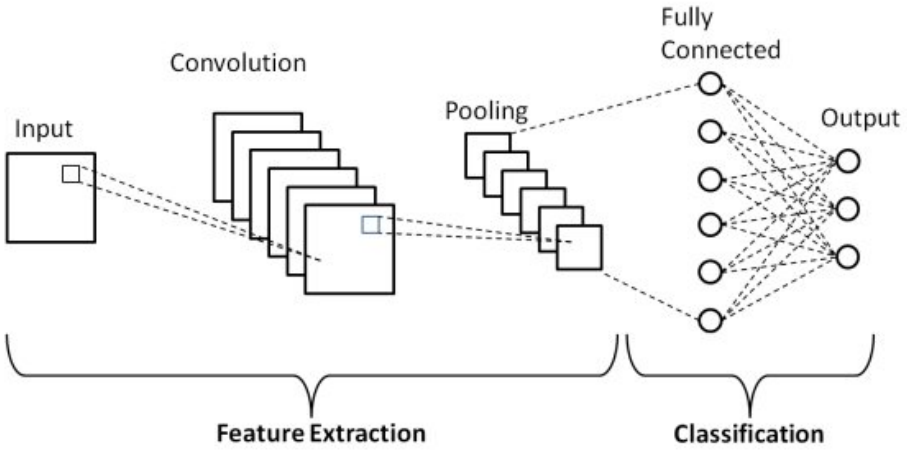

Figure 2.10: Flowchart of a CNN architecture for a classification task. *Figure taken from (Phung, Rhee, et al., 2019).*

2.4.1.4 *Additional deep neural network elements*

In our methods, we exploit the benefits of the following techniques to reinforce our neural networks.

BATCH-NORMALIZATION. The Batchnormalization algorithm introduced by Ioffe and Szegedy (2015) improves the performance and stability of neural networks. The input data are generally standadize using a zero mean and a standard deviation of one in order to ensure that each feature has the same contribution, which reduces the sensitivity of the model to small variations. It is based on this principle and allows to normalize and standardize the output activations of each layer of the network. Batch normalization adds two learnable parameters: a shift factor γ and scale factor β. These parameters encourage the ability of the network to take advantage of the non-linearity function in case it cannot learn with this constraint of zero mean and unit variance. They also control the needed mean and the variance of the layer which helps the optimization algorithm. It optimizes the training because networks converge quickly, allows higher learning rates, reduces the sensitivity to the initial starting weights and keeps a controllable range of values avoiding saturations for some non-linearity activations (Goodfellow et al., 2016).

DROPOUT. Dropout is a regularization operation that was introduced by Srivastava et al. (2014) It allows to prevent over-fitting during training time. At each training iteration, some randomly selected neurons are disabled and are discarded from the training process. Dropped neurons at one step are usually active at the

next step. Dropout prevents neurons from being highly dependent on only a small number of previous neurons.

EARLY STOPPING. In the context of model training, given a training set S divided into minibatch, *epoch* is a term reffering to one iteration where the model processes the whole training set i.e. all the minibatch to update its weights. Early stopping (Prechelt, 1998) is a regularization technique that stops the training process as soon as the validation loss reaches a plateau or starts to increase (Figure 2.11) instead of keep decreasing. In other words, it allows the model to automatically stop training when it reaches the optimal complexity.

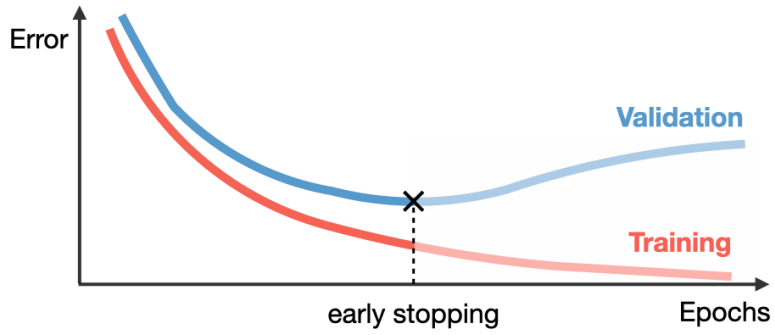

Figure 2.11: Early stopping principle. *Figure taken from (CS 230 - Deep Learning).*

ADAM OPTIMIZATION. An optimizer allows to optimize the convergence during the training process by adaptively changing the parameters θ. The current most popular method is the ADAM optimizer (Kingma and Ba, 2014). While SGD maintains a single learning rate η for all weight updates and the learning rate does not change during training, ADAM computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients.

2.4.2 *Data-driven tempo estimation*

Since the pioneer works of (Goto and Muraoka, 1994), many audio datasets have been annotated into tempo. This therefore encourages researchers to develop data-driven systems based on machine learning and deep learning techniques.

Such machine learning models have been used for automatic tempo estimation. They all rely on a twofold procedure, a first step of handcrafted rhythmic features extraction such as the ones presented in Section 2.3.2 and a second step of classification using machine learning algorithms. As an exhaustive list we can cite among these k-Nearest Neighbors (k-NN) (Seyerlehner, Widmer, and Schnitzer,

2007), Gaussian Mixture Model (GMM) (Xiao et al., 2008; Peeters and Flocon-Cholet, 2012), Support Vector Machine (SVM) (Chen et al., 2009; Gkiokas, Katsouros, and Carayannis, 2012; Percival and Tzanetakis, 2014) and bags of classifiers (Levy, 2011).

The first use of deep learning for tempo estimation was proposed by Böck and Schedl (2011). They propose the use of a bidirectional Long-Term Short Memory (LSTM)-Reccurent Neural Network (RNN) to perform a frame by frame beat classification of the signal. The network transforms the signal directly into a beat activation function with the spectral features of the audio signal used as input. An ACF is then used to determine the predominant tempo.

They extend this method in (Böck, Krebs, and Widmer, 2015) by using the bidirectional LSTM-RNN to predict the position of the beats inside the signal. This output is then used as the input of a bank of resonating comb filters to detect the periodicity and so the tempo. This technique still achieves the best results today in terms of Accuracy2 i.e. by taking into account the octave errors.

Schreiber and Müller (2018b) proposed a "single step approach" for tempo estimation using CNN. The network design is inspired by the flowchart of handcrafted systems: the first layer is supposed to mimic the extraction of an onset-strength-function. Their system uses as input mel-spectrograms and the network is trained to classify the tempo of an audio excerpt into 256 tempo classes (from 30 to 285 BPM). Excellent performances are to be denoted in terms of Class-Accuracy and Accuracy1. They also proposed in (Schreiber and Müller, 2019) the use of a VGG-style shallow network to estimate tempo and key. Using musically motivated directional filters, they were able to obtain good results for both tasks with the same architecture.

Recently, works that process a multi-task learning frameworks to estimate simultaneously tempo and beats (Böck, Davies, and Knees, 2019) and tempo, beats and down-beats (Böck and Davies, 2020) have been published. They showed that using such frameworks where one task take benefits of the other, allows to improve the performance.

2.4.3 *Data-driven genre classification*

As for tempo estimation, works dedicated to genre classification is based on a first stage of features extraction presented in Section 2.3.3 and a second one relying on machine learning algorithms to perform the classification. The work of Tzanetakis and Cook (2002) used probabilistic and unsupervised approaches with GMM and k-NN classifiers. In particular, k-NN is a relatively popular model even in recent

works (Pálmason et al., 2017; Iloga, Romain, and Tchuenté, 2018). They are used for extracting sequential patterns from music and generating music genre taxonomies.

Soon after the work of Tzanetakis and Cook (2002), SVM gained popularity (Li, Ogihara, and Li, 2003; Ness et al., 2009; Henaff et al., 2011). For instance, Silla Jr, Koerich, and Kaestner (2010) combine many different content-based features, using a genetic algorithm for the feature selection phase, and then use SVM for genre classification.

More recently, deep learning models have been extensively employed for this task. The use of CNN as a classifier has become very popular (Kong, Feng, and Li, 2014; Zhang et al., 2016; Schindler, Lidy, and Rauber, 2016). Choi, Fazekas, and Sandler (2016) and Choi et al. (2017) based their work on a large scale tagging experimenting various architecture of CNN and RNN. Pons, Lidy, and Serra (2016), Pons and Serra (2017), and Pons et al. (2017b) classify samples with CNN, experimenting with the dimensions of the convolutional filters. As we have seen in Section 2.4.1.3 image processing, filters are spatial, whereas in audio spectrograms, filter dimensions are related to time and frequency. Therefore, filters can be selected to make the models more sensitive to temporal features (tempo, rhythm) or frequency patterns (instruments, timbre or equalization).

Other works experiment different inputs to a network. Senac et al. (2017) show that using a fine-tuned selection of CNN input features, related to timbre, tonality and dynamics, could be more efficient and similar in accuracy than using spectrograms. Nanni et al. (2016) and Nanni et al. (2018) explore the idea of merging visual and acoustic features for music genre classification. To this end, visual descriptors are extracted using a CNN and audio features are extracted with audio signal feature extraction methods.

While all these works perform music genre classification with a first step of feature extraction, other works have shown their efficiency in the task by learning these features automatically. This process of learning audio features is often approached as an unsupervised learning problem. An overview of these techniques is given in (Ramírez and Flores, 2019), we will not detail them here. Some other works have shown better results on very large scale datasets in an end-to-end way using raw audio as input of deep learning models Pons et al., 2017a.

2.5 ELECTRONIC/DANCE MUSIC IN MUSIC INFORMATION RETRIEVAL

For more than a decade, several works in the MIR field have been dedicated to EDM. If we consider the historical evolution of this type of music (presented in Section 1.1), we can assume that the number of works concerning it is relatively

low. We present here a non-exhaustive list of these works which are related to our research. According to Butler (2006), the structural change in EDM music is generally dependent on timbre or rhythm variations. Logically, MIR works on EDM are focused on the two musical dimensions: rhythm and timbre.

Hockman, Davies, and Fujinaga (2012) perform a downbeat detection task on *breakbeat* rhythmic style genres of EDM (Hardcore, Jungle and Drum'n Bass) with a style specific model.

Panteli, Bogaards, Honingh, et al. (2014) developed a model to evaluate rhythm similarity between EDM tracks. For this, they assume that EDM rhythm is expressed via the *loop* which is a repetition of patterns associated with instruments (percussive or not). First, they segment the audio signal based on timbre features using a segmentation algorithm presented in (Rocha, Bogaards, and Honingh, 2013). It includes a set of musically informed rules to account for the fact that segment boundaries in EDM are usually on the beat. Second, they represent rhythm polyphony[2] by separating the audio into rhythmic streams. Features relative to rhythm in each of those streams are extracted after an onset detection step: the attack, the periodicity of the onsets and the metrical structure of the rhythm patterns. Finally they evaluate the rhythm similarity between the concatenated features vectors thanks to a perceptual rating.

Honingh et al. (2015) also deal with the music similarity in a multidimensional context through sub-similarities of rhythm and timbre. They lead perceptual experiments with a panel of listener to measure the similarity between rhythm, timbre and what they called general similarity. They also want to estimate the interaction between them in terms of two metrics of consistency. Their method have proved that in EDM when all the sub-similarity are high the output of their study is high, and when a sub-similarity is low, the output gets lower.

Panteli et al. (2017) present another method for similarity evaluation in EDM based on rhythm and timbre. They extract several features, some related to the periodicity of rhythmic events (the degree of repetition as well as the metrical position of the repeating pattern), others related to the presence or absence of low frequency instruments in the mix and the roughness. Using the annotated data on similarity described in (Honingh et al., 2015), they apply a linear regression model. In other words, they correlate the audio features to the perceptual data. Doing so, they perform a statistical analysis of the importance of each descriptor with rhythm and timbre similarities treated separately. Finally, they assume that the features related to rhythm are the most important to evaluate similarity in EDM.

2 Since the rhythm in EDM is marked by many instruments.

Other methods dedicated to EDM analysis are based on data aggregation. The work of (Hörschläger et al., 2015) aims to address tempo estimation octave errors in Electronic Music by incorporating style information extracted from Wikipedia[3]. Knees et al. (2015) developed two datasets annotated in style (genre), tempo and key. All the annotations are extracted from Beatport and are improved by gathering and exploiting users' feedbacks. Tempo annotations have been improved in (Schreiber and Müller, 2018a) with a tapping experiment. More details of these datasets are given in Section 3.2.

As we saw in the introduction, the number of genres/sub-genres in EDM is flourishing. Some research has been dedicated to their automatic classification. They are all based on the so called handcrafted features designed from domain knowledge. Among these, we can cite the work of Diakopoulos et al. (2009), the one of Camara (2017) and more recently the one of Caparrini et al. (2020). The latter perform an automatic EDM sub-genre classification and a comparison of the classification and the taxonomy in time. The datasets and their annotations are taken from Beatport. They aggregate two datasets from the platform at two-year intervals (2016 and 2018) annotated in 23 different sub-genres. They thus show that some sub-genres appear and others disappear, illustrating once again the volatile aspect of EDM taxonomy.

To perform sub-genre classification, they apply an audio features analysis and several machine learning algorithms dedicated to the classification of those features. They first compute for each track audio features with two library: pyAudioAnalysis Giannakopoulos, 2015 and Essentia Bogdanov et al., 2013. Those libraries contain several types of audio features relative to timbre and to rhythm. Combining all of those features, their mean and their standard deviation provides 92 inputs variables for the classifiers. As classifiers they use decision tree, random forest and extremely randomized trees gradient tree boosting. They finally perform a study on the statistical importance of the features. It shows that BPM-based features are the most efficient ones for classification i.e. they have the most discriminative power for sub-genres classification.

2.6 CONCLUSION

Starting from the two main axes of the thesis, i.e. automatic tempo estimation and automatic classification into musical genres, we have presented in this chapter some concepts specific to music analysis.

Knowing that tempo is an indivisible concept in rhythmic analysis, we began by defining the different aspects of rhythm. Thus, we were able to get familiar with

3 https://fr.wikipedia.org/

the theories that shed light on the different perceptual and temporal aspects of the rhythm. For the musical genre, we have also discussed its general definitions on the basis that it is an ambiguous and subjective concept. Whether it is for rhythm or genre, we wish to bring certain answers through our work to disambiguate these concepts, particularly with regard to EDM. We have divided our state of the art overview into two distinct parts, one for handcrafted systems and one for data-driven systems.

In the first one, we first described some signal processing concepts. We started with the most general notions and then focused on the tools adapted to the description of rhythm. For this, we have precisely defined the onset and the way to detect it in a musical signal with the OEF. Through various works, we have shown that there are different temporal representations allowing to analyze the OEF. Among these we will further detail the work of (Peeters, 2011; Marchand and Peeters, 2016a) which serves as a baseline for the development of our Deep Rhythm method (Chapter 4). In addition to the handcrafted rhythmic features, we also mentioned two other types of features used for the automatic classification of musical genres related to timbre and pitch. We will get a closer look to timbre-related features for the deep rhythm multi-input method in Section 5.4.

In the second part, we have described the large-scale analysis tools allowing a supervised classification thanks to machine learning. We started by defining what supervised learning is. Then, we presented an explanation of deep learning algorithms that are DNN and CNN, both ubiquitous in this thesis. Next, we mentioned the work dedicated to the automatic estimation of tempo with the use of machine learning. We did the same for the musical genre works. Note that handcrafted features based on domain-knowledge are very often used as input to these analysis models. Our work is derived from a combination of handcrafted features used as input to a deep learning model.

We ended by presenting the various MIR works dedicated to EDM. It is a musical genre that has been relatively unexplored. Nevertheless, the work of analysis of EDM focuses on two musical aspects: rhythm and timbre. The analyses show that rhythm is a key element in the description of the musical genres in EDM. We wish to apply this type of analysis by describing certain genres in EDM using rhythmic features and data-driven genre classification. This will be the main focus of this thesis and will be found in next chapters. These approaches, however, necessitate to manipulate datasets that are annotated for the given taks. In the next chapter, we will introduce such datasets and their link to the two tasks we are aimed at accomplishing.

3

DATASETS

3.1 INTRODUCTION

Our work is geared towards two main tasks: a global tempo estimation task and a rhythm-oriented genre classification task. We discussed the fact that the nerve center of our methods is based on the combination of rhythmic handcrafted and data-driven systems. Therefore, the best way to evaluate the results of our methods is to test them on different annotated datasets.

Tempo is usually defined as (and annotated as) the rate at which people tap their foot or their hands when listening to a music piece. Several people can therefore perceive different tempi for the same piece of music (Hainsworth, 2004; Peeters, 2006; Percival and Tzanetakis, 2014). This is due to the hierarchy of the metrical structure in music (to deal with this ambiguity the research community has proposed to consider octave errors as correct) and due to the fact that without the cultural knowledge of the rhythm pattern(s) being played, it can be difficult to perceive "the" tempo (or even "a" tempo). This last point, of course, opens the door to data-driven approaches, which can learn the specifics of the patterns and their use to infer a given tempo. In this work, we do not deal with the inherent ambiguity of tempo and consider the values provided by annotated datasets as ground-truth. Since the tempo of a track can vary over time, different approaches have been proposed so that an entire piece of music can be annotated with a single tempo value. Hainsworth (2004) defines it as the mean of the inter-beat intervals whereas others (Peeters, 2006; Percival and Tzanetakis, 2014; Böck, Krebs, and Widmer, 2015) recommend to compute the median of the inter-beat intervals to counter the influence of outliers. For more than a decade, annotated datasets dedicated to the global tempo estimation have been developed. Some have been created for challenges such as the Music Information Retrieval Evaluation eXchange (MIREX) [1], others for the evaluation of new methods as we wish to do. Beyond tempo, other rhythmic features can be annotated such as beat positions.

As for datasets annotated into musical genre, their development is all the more consequent knowing that it is a very popular task. The application of deep learning methods for automatic music tagging task requires datsets on a very large scale.

1 https://www.music-ir.org/mirex/wiki/MIREX_HOME

Music tagging encompasses not only musical genres but also other tags such as mood, period or theme of a piece of music. The number of labels used in such context can be very high.

As we will see, the majority of datasets annotated into tempo are also annotated in musical genres. For some of them, this musical genre labeling is thought to stick to the rhythmic characteristics. This will allow us to analyze the musical genres through their overall tempo and rhythmic characteristics. Knowing that we do not want to carry out a classification in musical genres on a large scale, we will be satisfied with the annotations in genre of these smaller datasets. From now until the end of this manuscript we will refer to our task of "rhythm-oriented musical genre" classification simply as "genre" classification for the sake of readability.

In Section 3.2 we first describe the datasets commonly found for tempo and rhythm genre estimation. EDM datasets presented have been created specifically for the automatic estimation of tempo (or key) but not for the classification in musical genres. This is what motivated us to create two new annotated datasets wich are balanced in EDM genres. We detail those in Section 3.3.

For all datasets presented, we indicate their distribution in tempo and/or genre according to their annotations and applications. These distributions highlight the frequency of annotations for each label in the case of tempo and/or genre.

3.2 COMMONLY USED DATASETS

We first describe the different datasets commonly used for the estimation of the global tempo. While they have been created for tempo estimation, it will be possible to exploit some of them for genre classification as well. We only consider the datasets for which audio data are available and free of charge for research purposes.

ACM Mirum (ACM) proposed by Peeters and Flocon-Cholet (2012) – 1,410 tracks – tempo (perceptual tempo) annotated dataset. We use it in chapters 4 and 5 as test set for tempo estimation.

Ballroom (BR) proposed by Gouyon et al. (2004) – 698 tracks – tempo and genre annotated dataset. This dataset is annotated in rhythmic genres, which means that the classes correspond to the rhythmic patterns that define ballroom dance music. We use it as test set for tempo estimation in chapters 4 and 5 and as train/test/-validation set for genre classification in chapters 4, 5, 6.

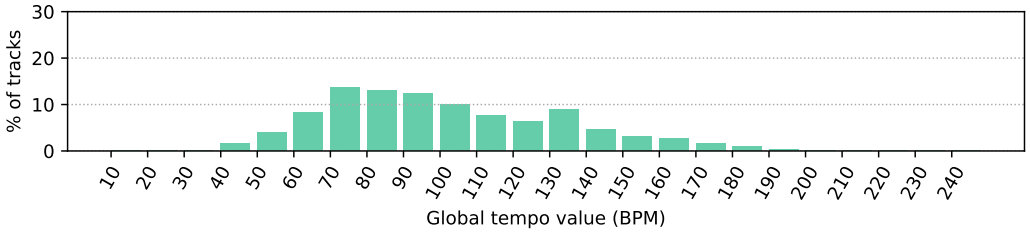

Figure 3.1: ACM tempo distribution.

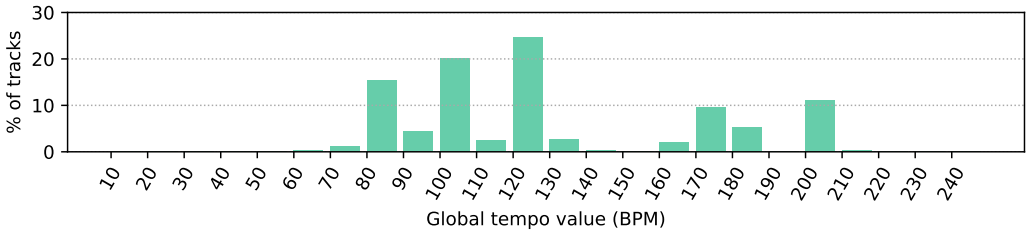

Figure 3.2: BR tempo distribution.

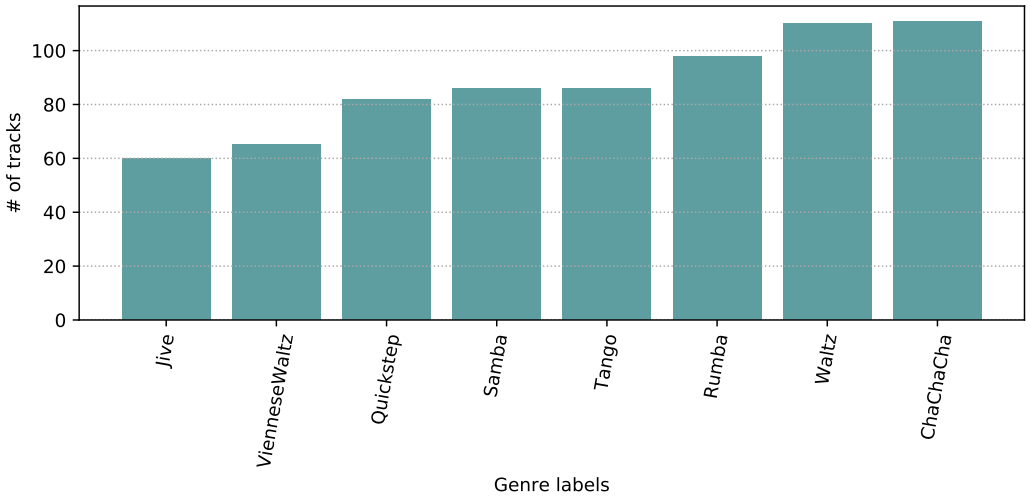

Figure 3.3: BR genre distribution.

Extended Ballroom (EBR) presented by Marchand and Peeters (2016b) – 4,180 tracks – tempo and genre annotated dataset. It is an extension of the BR dataset with additional ballroom styles. For the tempo estimation task, we remove from the dataset the tracks already present in the BR dataset for experimental training/testing purposes, as in (Schreiber and Müller, 2018b). We therefore refer to this reduced dataset as ***tempo Extended Ballroom (tEBR)*** – 3,826 tracks. For the genre classification, as in Marchand and Peeters, 2016a, we keep only the nine most

represented genres. We therefore refer to this reduced dataset as ***genre Extended Ballroom (gEBR)*** – 3,992 tracks. We use tEBR as train set for tempo estimation in chapters 4 and 5 and gEBR as train/test/validation set for genre classification in chapters 4, 5, 6.

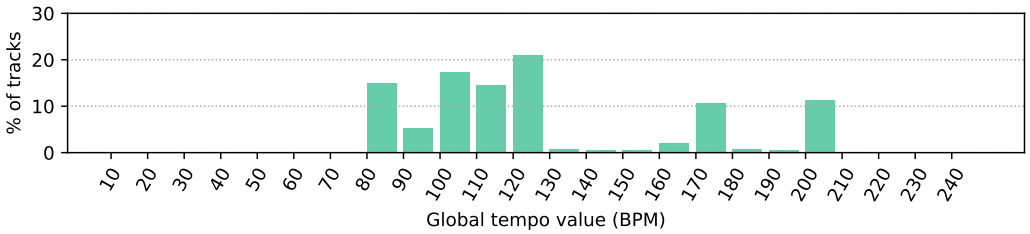

Figure 3.4: tEBR, tempo distribution.

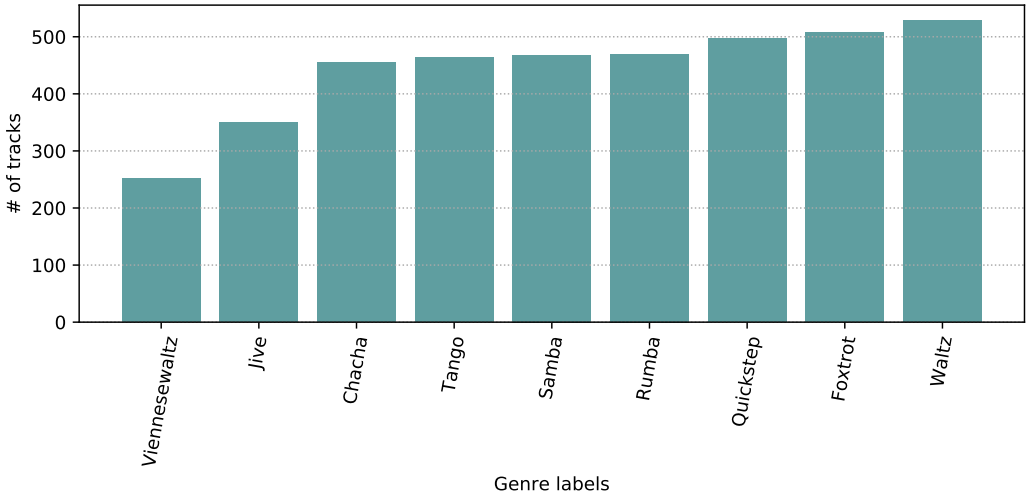

Figure 3.5: gEBR, genre distribution.

tempo GiantSteps (tGS) proposed by Knees et al. (2015) – 664 tracks – tempo and EDM genre annotated dataset. The audio files are excerpts of music taken from Beatport[2], we detail the platform in the next Section 3.3. We use the annotations corrected by Schreiber and Müller, 2018a in a perceptual tapping experiment[3]. We use it as test set for tempo estimation in chapters 4 and 5.

Greek Dance (Gr) proposed by Holzapfel and Stylianou (2011) – 180 tracks – greek dance annotated dataset. As for BR and EBR, it is annotated into rhythmic

2 https://www.beatport.com/

3 They asked a panel of participants to tap the tempo along the tracks. By aggregating the results of the experiments, they were able to define new annotations that are more perceptually relevant

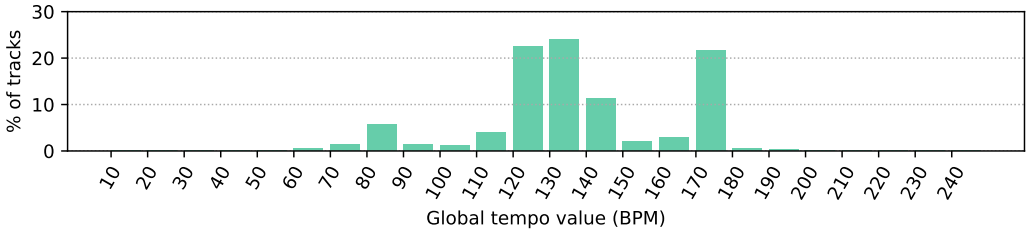

Figure 3.6: tGS tempo distribution.

genres. The tracks are divided into 6 classes equitably distributed (30 tracks per class): *kalamatianos*, *kontilies*, *maleviziotis*, *pentozalis*, *sousta* and *kritikos syrtos*. We use it as train/test/validation set for genre classification in chapters 4, 5, 6.

GTzan Tempo (GTzan) initially proposed by Tzanetakis and Cook (2002) for genre classification and later annotated into tempo by Marchand, Fresnel, and Peeters (2015) – 1000 tracks. The original dataset is popular for tagging or music genre classification tasks due to its large size. It is divided into 10 equally distributed genres (100 tracks per class): *blues*, *classical*, *country*, *disco*, *hip-hop*, *jazz*, *metal*, *pop*, *reggae*, *rock*. We use it as test set for tempo estimation in chapters 4 and 5 and as train/test/validation set for genre classification in chapters 4, 5, 6.

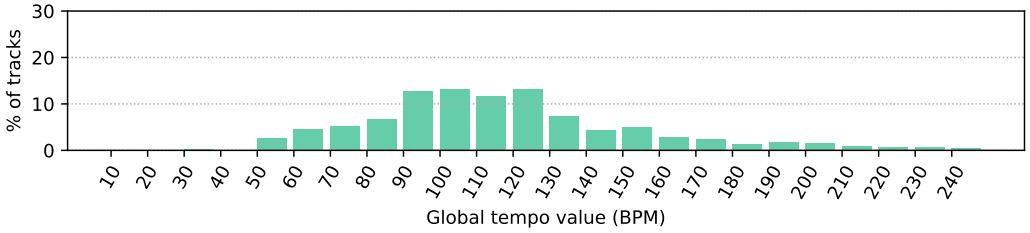

Figure 3.7: GTzan Tempo, tempo distribution.

Hainsworth (Hains.) proposed by Hainsworth, 2004 – 222 tracks – tempo annotated dataset. We use it as test set for tempo estimation in chapters 4 and 5.

ISMIR04 Songs (ISMIR04) proposed by Gouyon et al., 2006 – 464 tracks – tempo annotated dataset. We use it as test set for tempo estimation in chapters 4 and 5.

tempo LMD (tLMD) a subset of the Lack MIDI dataset proposed by Raffel (2016) and tempo annotated by Schreiber and Müller (2017) – 3,611 tracks. It is annotated in genre but the quality of the annotations does not allow us to use it

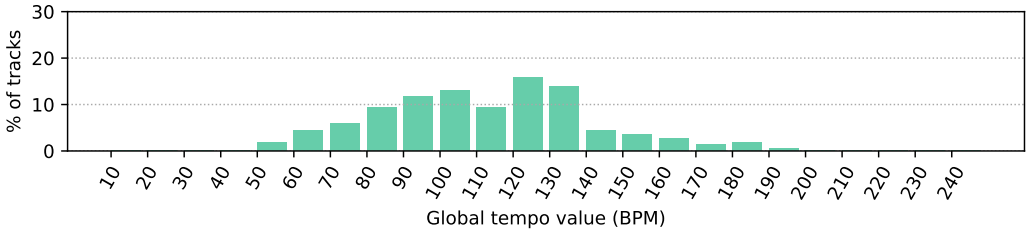

Figure 3.8: Hains. tempo distribution.

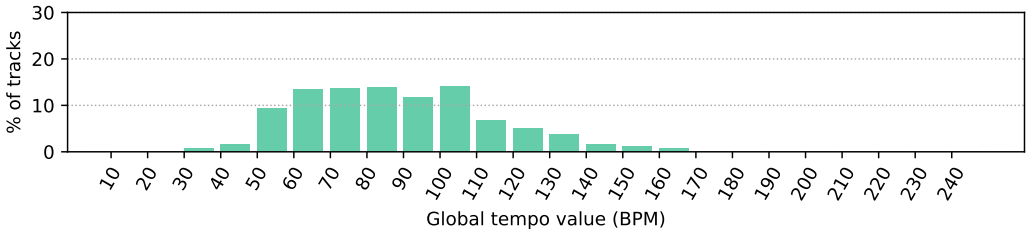

Figure 3.9: ISMIR04, tempo distribution.

in our experimental protocols (many tracks are annotated as "unknow"). We use it as train set for tempo estimation in chapters 4 and 5.

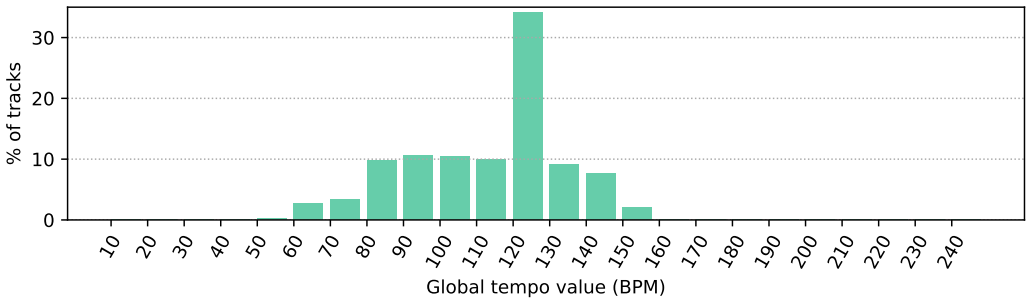

Figure 3.10: tLMD, tempo distribution.

tempo MTG (tMTG) proposed by Faraldo, Jorda, and Herrera (2017) for EDM key estimation – 1,159 tracks – tempo annotated using a tapping method by Schreiber and Müller (2018b). We use it as train set for tempo estimation in chapters 4 and 5

genre MTG (gMTG) – 1,823 tracks. We propose to merge the two EDM datasets of the state of the art, tGS and tMTG. The goal is to obtain a tempo and genre annotated dataset for our experiments in both tasks. We use it as test set for tempo

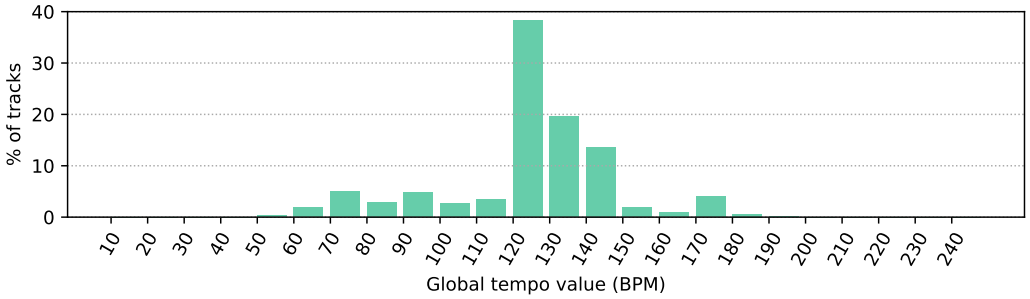

Figure 3.11: tMTG, tempo distribution.

estimation in chapters 4 and 5 and as train/test/validation set for genre classification in chapters 4, 5, 6.

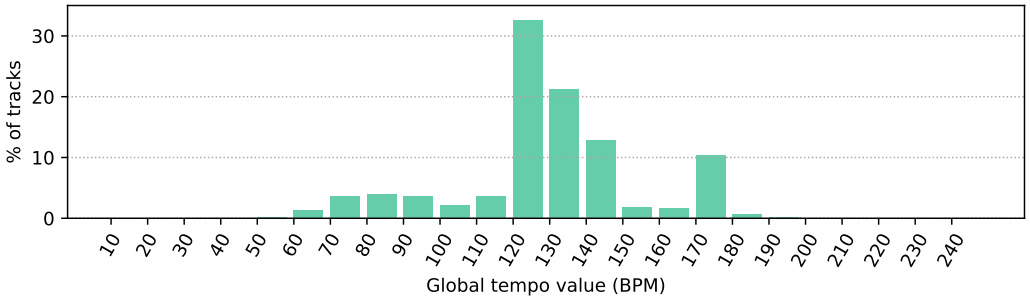

Figure 3.12: gMTG, tempo distribution.

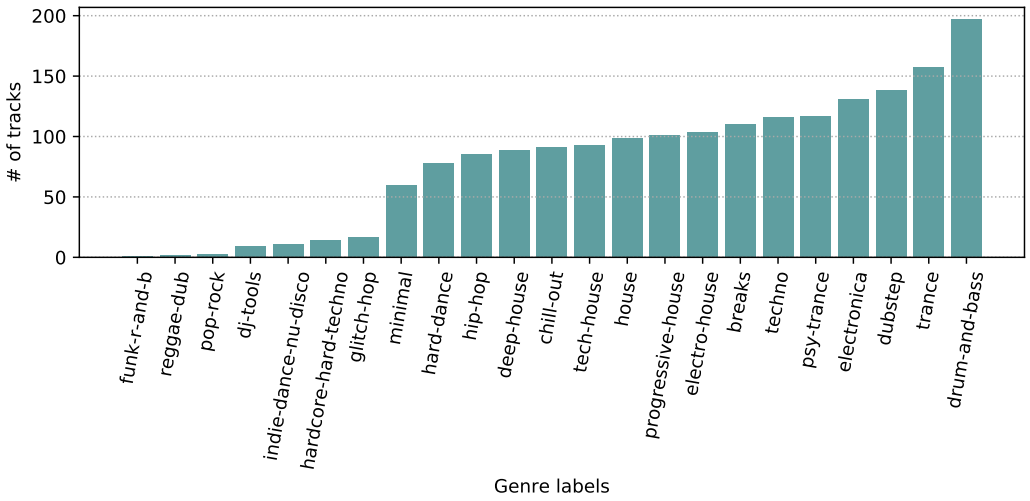

Figure 3.13: gMTG, genre distribution.

SMC proposed by Holzapfel et al. (2012) – 217 tracks – tempo annotated dataset. It is designed to make the task of beat tracking more difficult. We use it as test set for tempo estimation in Chapter 4 and 5

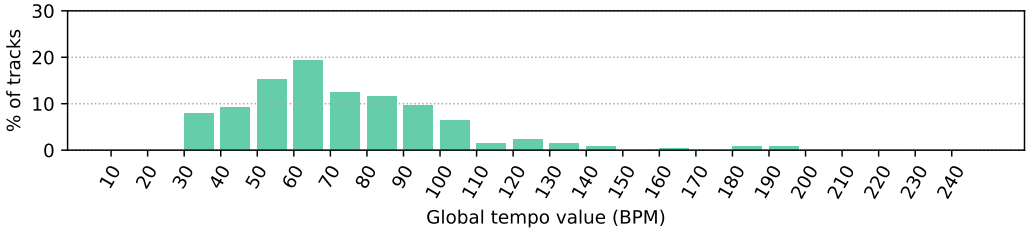

Figure 3.14: SMC tempo distribution.

Combined – 4,675 tracks. Schreiber and Müller, 2017 proposed in their experimental protocol to merge all the tempo annotated test set into one to perform a large scale analysis. This dataset is the union of ACM, ISMIR04, BR, Hains., GTzan, SMC, tGS.

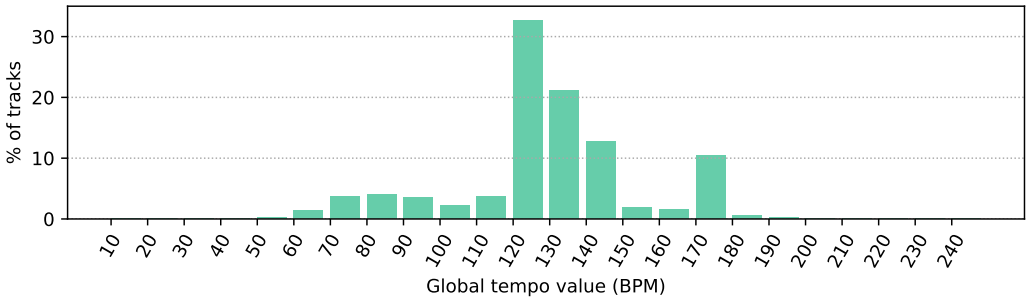

Figure 3.15: Combined, tempo distribution.

We present in Table **??** the involvement and use of the different datasets in our work.

3.3 ELECTRONIC DANCE MUSIC DATASETS

As seen, only the tGS, tMTG and gMTG datasets presented above are dedicated to EDM. The original purpose of these datasets was the automatic estimation of tempo and that is the main reason why they are unbalanced in musical genre (Figure 3.13). As a reminder, we assume that the rhythmic structure of an EDM track is a major indication of its musical genre. We want to check our assumption using data-driven methods. We therefore need datasets which are genre-balanced. Such datasets do not exist in the literature or are not publicly available.

Table 3–1: Utilization of commonly used datasets

Dataset	Tempo	Genre
ACM	test	-
BR	test	train/test/validation
tEBR	train	-
gEBR	-	train/test/validation
tGS	test	-
Gr	-	train/test/validation
GTzan	test	train/test/validation
Hains.	test	-
ISMIR04	test	-
tLMD	train	-
tMTG	train	-
gMTG	-	train/test/validation
SMC	test	-
Combined	test	-

For the three EDM datasets, tGS, tMTG and gMTG, audio excerpts and annotations have been retrieved from the same source: the *Beatport* website. It is one of the most popular websites for DJs and EDM listeners. It is both a platform for sales and the promotion of new trends. More than 25,000 new tracks are published every week. For each track uploaded, Beatport provides precise annotations into genre, key and tempo. The genre labeling is subject to current trends, despite an unbreakable core, hence some sub-genres are sometimes removed from the platform and others are added every 6 months. The taxonomy includes no less than 30 different core genres and more than 150 sub-genres. For each track, a 2-minute audio excerpt is freely available.

We have gathered from Beatport around 67,000 audio excerpts of 30 different genres and 48 different sub-genre. From these data, we have created two smaller genre-balanced EDM datasets. Because of computational resources, we have not kept all the 67,000 tracks. It should be noted that the two smaller datasets are also balanced in sub-genres (when the core genres are subdivided into sub-genres). Finally, the amount of collected data has allowed us to only use one track of each artist.

large Beatport (lBP) – 3,000 tracks with 100 tracks per genre. The 30 genres selected are all the core genres of the Beatport taxonomy: *afro-house*, *big-room*, *breaks*, *dance*, *deep-house*, *dj-tools*, *drum-and-bass*, *dubstep*, *electro-house*, *electronica-downtempo*, *funk-soul-disco*, *funky-groove-jackin-house*, *future-house*, *garage-bassline-grime*, *glitch-hop*, *hard-dance*, *hardcore-hard-techno*, *hip-hop-r-and-b*, *house*, *indie-dance-nu-disco*, *leftfield-bass*, *leftfield-house-and-techno*, *minimal-deep-tech*, *progressive-house*, *psy-trance*, *reggae-dancehall-dub*, *tech-house*, *techno*, *trance*, *trap-future-bass*.

small Beatport (sBP) – 1,100 tracks with 100 tracks per genre. For this dataset we focus on a subset of genres which have been wisely selected for their rhythmic characteristics. For some of the genres described, we illustrate below examples of basic rhythmic drum patterns using a sixteenth-note MIDI grid. This means each sequence is only one bar long. Each line of a grid corresponds to a percussive elements played by a drum-machine. In the figures, the elements are represented by icons, from bottom to top: bass-drum, snare-drum (or hand-clap), open hi-hat (or crash), closed hi-hat. We have chosen to represent only these 4 elements in order to illustrate simplified structures but many others elements are also used in the composition of an EDM track such as tomes, cymbals or sub-kicks. It is important to note that the examples are indicative. Indeed, Hockman, Davies, and Fujinaga (2012) mentioned that the breakbeat patterns are not universal.

- *dance*: Mainly characterized by synthesized riffs, male or female vocals, rap passages, sampling and a strong bass-drum. Its rhythmic structure is of the four-on-the-floor type: four percussion beats on four measures (4/4), each sequence of 16 or 32 beats delimited by a strong moment. The tempo ranges between 110 and 150 BPM. A rhythm pattern example is illustrated in Figure 3.19.

- *drum-n-bass*: Intensive use of "breakbeat chopping", i.e. the repetition of drum strokes from recording samples. The tempo is often above 160 BPM, with characteristic bass lines perceptible at half the tempo. A rhythm pattern example is illustrated in Figure 3.16.

- *dubstep*: breakbeat rhythmic structure, influenced by jungle, drum'n'bass, dub but also techno with a more instrumental approach, more introspective and guided by the physical energy of the low frequencies. The rhythm is characterized by the snare-drum specifically placed on the 3rd beat. The tempo is generally close to 140 BPM. A rhythm pattern example is illustrated in Figure 3.17.

- *electronica-downtempo*: Often associated with trip-hop, it can also encompass more or less experimental and/or minimalist sub-genres such as ambient. Its

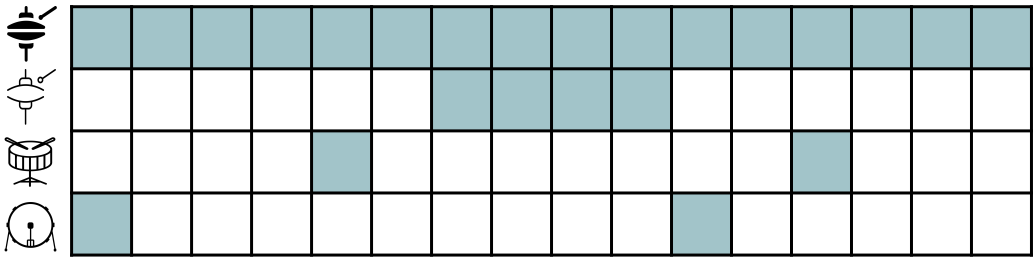

Figure 3.16: Drum-n-bass (110 - 150 BPM) rhythm pattern example.

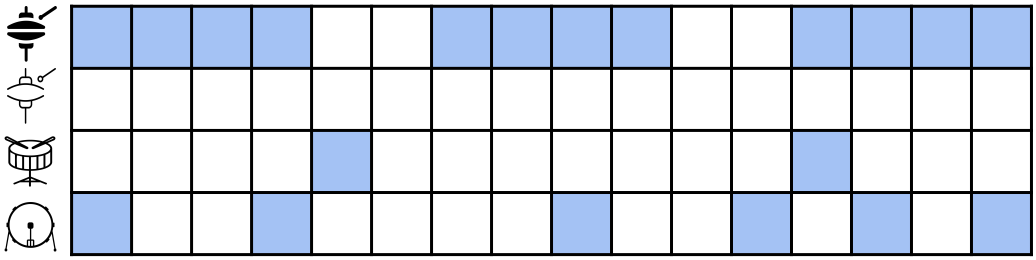

Figure 3.17: Dubstep (140 BPM) rhythm pattern example.

harmonic texture is often dense and abundant and rhythm is characterized by groove percussion. Even though it is often defined by breakbeat rhythms, this genre is mainly defined by its tempo which is very low, between 60 to 90 BPM.

- *funk-soul-disco*: Mixture of funk for its harmonic component (voice, jazzy brass-type instrumentation) and disco for the rhythm (four-on-the-floor). It is also influenced by a new scene, associated with the term "electro-funk". The pulsations are marked by the bass-drum on each beat of the 4/4 meter and the offbeats are underlined by a hi-hat often replaced by a hand-clap or other instruments. Tempo is generally around 120 BPM. A rhythm pattern example is illustrated in Figure 3.19.

- *hardcore-hard-techno*: It was historically the first breakbeat genre. Today, many sub-genres are affiliated with hardcore and their rhythm pattern can be breakbeat (breakcore) or four-on-the-floor (hard-techno). The bass-drum or kick is the most recognizable element, it is created by saturation, filter effects and superimposition of heavy synthetic percussive sounds. The presence of synthetic "snapping" sounds enhances the classical hi-hat and snare-drum palette. In a more global way, the accent is generally put on cuts or decreases

in intensity during the piece often followed by a high[4]. Many sub-genres result from this and although the tempo is generally very high: it can varies between 140 and 220 BPM. A rhythm pattern example is illustrated in Figure 3.19.

- *hip-hop-r-and-b*: It is a "meta-genre" in its own in the sense that its taxonomy is vastly developed. The rhythmic structure is that of breakbeat. It is represented in this dataset by sounds with rhythmic characteristics "boom bap" or "trap". "boum bap": Drums and vocals are loudest – samples are often in the background of the mix. Also, low pass filters are often used to prevent the mix from getting too bright and maintains lo-fi feel. Kick-drums are usually on the 1st and 3rd beat to provide the "boom" and the snare on the 2nd and 4th beat to provide the "bap". Beats are often sparse to allow rappers' lyrics to take focus in the music. Tempo is usually around 80 to 120 BPM. A rhythm pattern example is illustrated in Figure 3.18. "trap": Extensive use of the Roland TR-808 drum-machine kick (characterized by the greater presence of sub-bass), 16th notes, triplets and other faster time divisions with hi-hat tones, synthesizer pads and virtual string ensembles. Tempo is usually around 100 to 140 BPM.

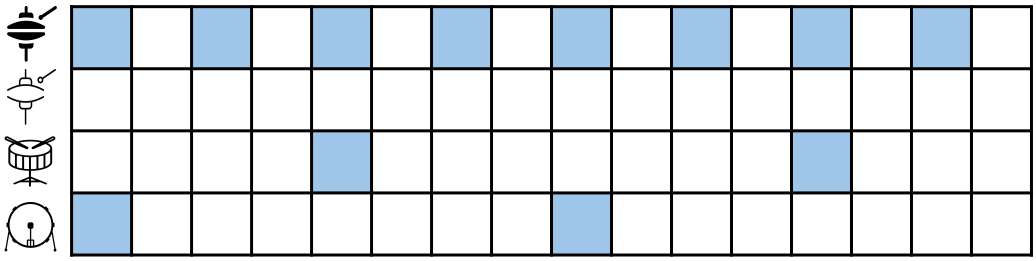

Figure 3.18: Hip-Hop (Boom bap, 80 - 120 BPM) rhythm pattern example.

- *house*: Typical four-on-the-floor rhythmic pattern consisting in a steady kick-drum on each downbeat in a 4/4 meter. Rhythmic distinguishing characteristics involve offbeat open hi-hat patterns and snare or claps on the two and four of every bar. Harmonic content, vocals and instrumentation are often borrowed from 'disco'. Tempo ranges from 115 to 135 BPM. A rhythm pattern example is illustrated in Figure 3.19.

- *reggae-dancehall-dub*: It is a combination of 3 sub-genres sharing the same rhythmic structures. Historically reggae is at the origin of dub (whose initial

4 In the jargon of electronic music a high is the climax of a rise in intensity.

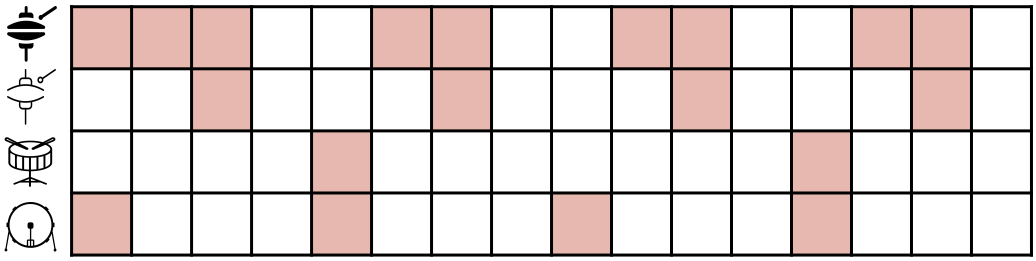

Figure 3.19: House (115 - 135 BPM) rhythm pattern example. It is a basic foor-on-the-floor pattern, it can also correspond to dance (110 - 150 BPM), disco (120 BPM) or techno (120 - 150 BPM)

term designated the principle of remix) and dancehall. Beyond the "riddims", rhythmic structure specific to these genres (determined by the bass line), they are characterized by: a four beat rhythm, with accentuation by the bass line which executes small riffs of one bar often in figure of eighth note and drums on the weak beats; the "skank" which designates the after-beat (in fact an accentuation of the second and fourth beats), generally marked by a flat chord played by the rhythm guitar or the keyboard; a snare-drum hit on the one drop (3rd beat). Tempo, even if it is generally perceived at double the tempo, ranges from 60 to 90 BPM. A rhythm pattern example is illustrated in Figure 3.20.

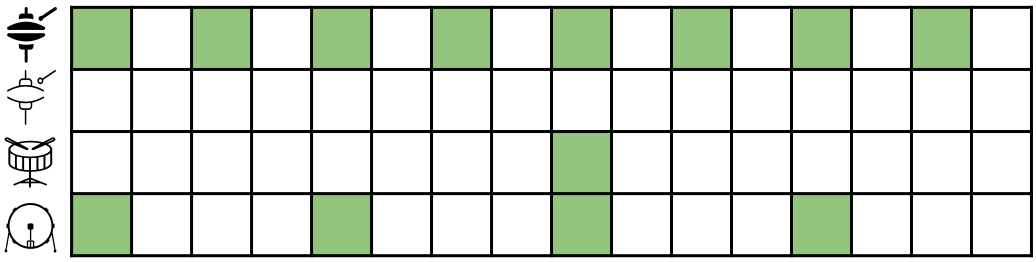

Figure 3.20: Reggae-dancehall-dub (stepper style, 60 - 90 BPM) rhythm pattern example.

- *techno*: The melody is reduced to the background, relying more on bass riffs and poly-rhythmic drums superimposed on a common bass-drum at the four-on-the-flour. The defining characteristics of the song is rhythmic, with grooves and percussive riffs taking precedence over the more traditional melodic and harmonic structure. Tempo typically ranges from 120 to 150 BPM. A rhythm pattern example is illustrate in Figure 3.19.

- *trance*: It is characterized by the use of thick and complex harmonic components. The rhythmic structure is less dominant and complex than in other types of EDM. From a global point of view, trance often uses arpeggios, drum rolls and long synthesizer crescendos. Tempo ranges from 125 to 150 BPM, mostly around 140 BPM. A rhythm pattern example is illustrated in Figure 3.21.

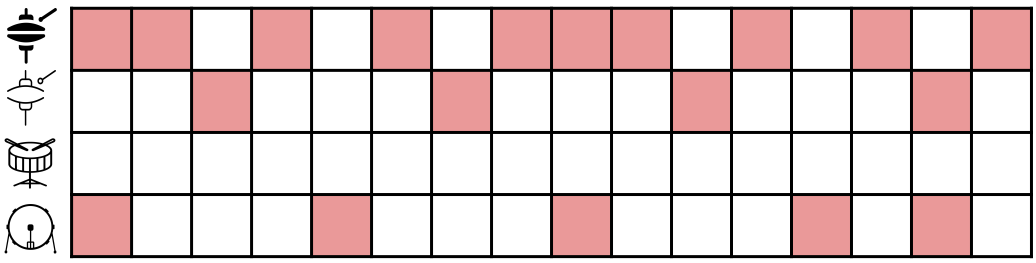

Figure 3.21: Trance (125 - 150 BPM) rhythm pattern example.

In (Knees et al., 2015), it is mentioned that the tempo annotations provided by Beatport are not reliable since Beatport does not provide information on the algorithm used for the estimation. To correct the ill-defined BPM values, Knees et al. (2015) download the user comments associated with each track that refers to the tempo values, they apply a restrictive filtering to obtain annotations made by human rather than an unknown algorithm. The corrected dataset is the tGS one. (Schreiber and Müller, 2018a) on their side proposed an experiment based on tapping to annotate the tMTG dataset.

We do not perform such a correction here. Indeed, our work with these datasets focuses mainly on analyzing musical genre. We still use these datasets for tempo estimation in some of our experiments with the annotations provided by beatport. To have an idea of the tempo range of these datasets, we display the distribution of the tempo values annotated by Beatport in figures 3.22 and 3.23 for the lBP and the sBP, respectively. We can observe on these graphs that the majority of the tracks are annotated with tempi between 120 and 130 BPM.

3.4 DISCUSSION

In this chapter, we have presented the datasets used in our experiments: on one side for global tempo estimation; on the other side for genre or rhythmic genre estimation. We have at our disposal two published EDM datasets. The tGS and the gMTG are reliably annotated in tempo but their musical genre distribution

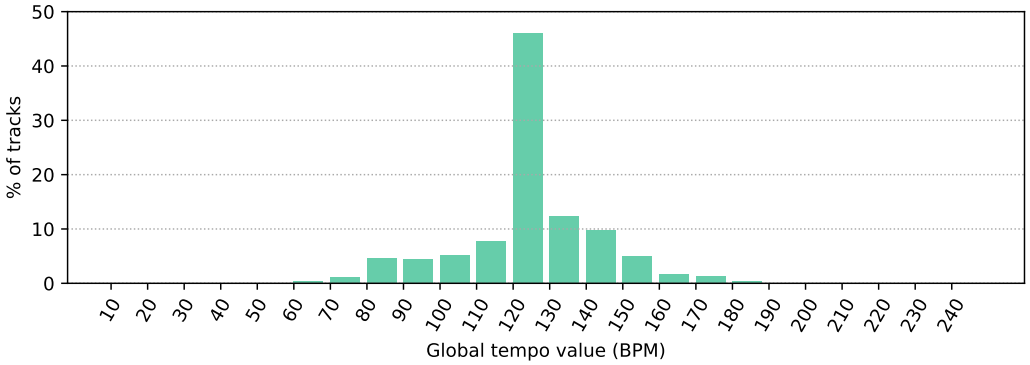

Figure 3.22: lBP dataset, tempo distribution.

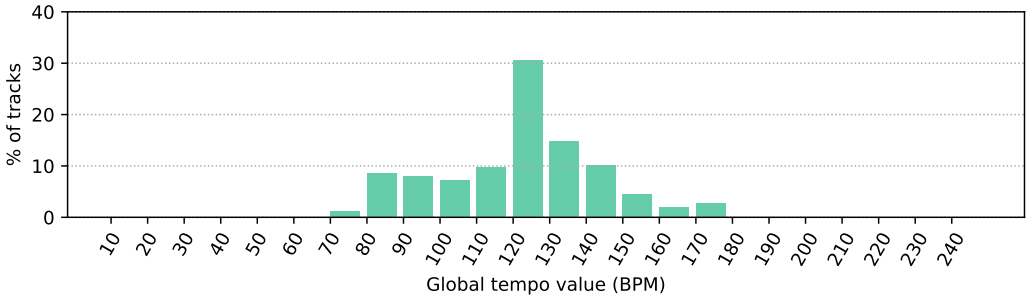

Figure 3.23: sBP dataset, tempo distribution.

is unbalanced. This has led us to create two genre-balanced EDM datasets: lBP and sBP. We have described in details EDM genres within the sBP dataset. Further details on the genres one can encounter in the different datasets can be found in the corresponding literature.

The remaining datasets give us the possibility to draw conclusions on a larger scale. The Ballroom datasets BR and EBR are quite interesting to study since Ballroom dances share some distinctive features with EDM such as rhythmic patterns, tempo and repetitions. The Gr dataset is also annotated in rhythmic genres. Thus, we consider the results of our work on these datasets as an important baseline. Plus, the GTzan dataset is balanced in popular musical genres and allows us to analyze the generalization of our classification methods.

It is important to note that the use of annotated datasets for tempo estimation has been debated for several years. Indeed, some criteria of these datasets (their size, the quality of their annotations, the examples chosen for their simplicity, etc.) constitute a ceiling to the evaluation of tempo estimation (Holzapfel et al., 2012;

Schreiber, Urbano, and Müller, 2020). We are aware of these limitations and for this reason we have chosen a wide spectrum of different dataset types (in relation to their musical genres or the features they highlight). Moreover, the objective of our work is not to outperform the previous scores to the nearest comma but rather to come up with original methods and analyze their impact on the tasks we wish to explore.

All information related to the different datasets are stored in an optimized format as a .json file. We created a .json file by dataset. In the following, we provide the nomenclature of the file relative to the gEBR dataset.


```
{
    "genres": {
        "Chacha": 8,
        "Foxtrot": 0,
        "Jive": 2,
        "Quickstep": 7,
        "Rumba": 4,
        "Samba": 3,
        "Tango": 5,
        "Viennesewaltz": 6,
        "Waltz": 1
    },
    "track_infos": {
        "100601": {
            "album": "Ballroom Swing",
            "artist": "Herberman",
            "dataset": "Extended_Ballroom",
            "filepath": "~/ExtendedBallroom/Waltz/100601.mp3",
            "genre": "Waltz",
            "hash": "273d858e7eaa59e7970883dfa4d0be97",
            "id": "Herberman",
            "tempo": 87.0,
            "title": "Morning Song"
        },
        ...
    }
}
```

4

DEEP RHYTHM

4.1 INTRODUCTION

Our objective is to study the rhythmic structure and more specifically its impact on the characterization of a musical genre. Motivated by this goal, we define two tasks:

- A tempo estimation task – as we have seen in the Section 3.3, we emphasize the fact that tempo ranges are typical of EDM genres.
- A genre classification task based on the assumption that the rhythmic structure is enough to retrieve the genre classes.

In this chapter, we propose a method to perform these two tasks. In Chapter 2, we have seen the two main strategies to deal with such objectives: knowledge-driven (handcrafted) and data-driven systems. Our method is at the crossroads of these two strategies, we name it Deep Rhythm (DR).

First, we want to obtain a representation of the periodic characteristics that make up the different levels of the rhythmic structure contained in a track. As we have described, the rhythm (and by extension the tempo) has a purely perceptive dimension. This is a substantial part of the development of a harmonic representation of rhythm.

Then, thanks to the annotated datasets described in Chapter 3, we want to learn automatically (using supervised training) the information that allows us to perform both tasks. This incites us to rely on deep learning schemes. The link between these two steps is essential: the automatic learning from a periodic representation of the rhythm annotated data is the essence of our method.

We begin this chapter with a description of the works that has motivated the development of our DR method (Section 4.2). We then present the Harmonic Constant-Q Modulation (HCQM) representation of the rhythm we propose, its computation and its different parameters (Section 4.3). This representation is used as an input to a deep Convolutional Neural Network (CNN) We describe its architecture and its training procedure (Section 4.4). We show that the architecture of this model with the HCQM as input allows us not only to perform the task of tempo estimation but also that of rhythm-oriented genre classification.

Within the considered datasets, the tempo inside a track can vary over time (actually some of the segments contain silence and the tempo is not even defined for those); however, the ground-truth annotations only provide a single tempo annotation value. To match our local tempo estimation to the global tempo annotation, we compute the mean of the softmax vectors over frame to estimate this single tempo. Interestingly, we show that if we would only consider the best softmax vector (oracle method) among those, the performances would actually be much higher. We therefore add an Attention Mechanism (AM) on top of DR to infer the best single tempo estimation from the sequence of predictions. This is describe in Section 4.5. Finally we evaluate the results of the DR method on the one hand for the estimation of global tempo and on the other hand for the rhythm-oriented genre classification (Section 4.6).

4.2 MOTIVATIONS

The method we propose here belongs to the data-driven systems in the sense that we learn from the data. It also considers both the tempo and rhythm pattern in interaction by adequately modeling the audio content through a handcrafted feature representation. The tempo of a track can of course vary along time, but in this work we focus on the estimation of global tempo and rhythm-oriented genre.

4.2.1 *Harmonic representation of rhythm components*

From Fourier series, it is known that any periodic signal $x(t)$ with period T_0 (or fundamental frequency $f_0 = 1/T_0$) can be represented as a weighted sum of sinusoidal components whose frequencies are the harmonics of f_0:

$$\hat{x}_{f_0,\underline{a}}(t) = \sum_{h=1}^{H} a_h \sin(2\pi h f_0 t + \phi_h) \tag{4–13}$$

For the voiced part of speech or pitched musical instrument, this leads to the so-called "harmonic sinusoidal model" (McAuley and Quatieri, 1986; Serra and Smith, 1990) that can be a starting point for audio coding or transformation. This model can also be used to estimate the pitch of a signal (Maher and Beauchamp, 1994): estimating the f_0 such that $\hat{x}_{f_0,\underline{a}}(t) \simeq x(t)$. The values a_h can be estimated by sampling the magnitude of the DFT at the corresponding frequencies $a_{h,f_0} = |X(hf_0)|$. The vector $\underline{a}_{f_0} = \{a_{1,f_0} \cdots a_{H,f_0}\}$ represents the spectral envelope of the signal and is closely related to the timbre of the audio signal, hence the instrument playing. For this reason, these values are often used for instrument classification (Peeters, 2004).

For audio musical rhythm, Peeters (Peeters, 2006; Peeters, 2010; Peeters, 2011) proposes to apply such a harmonic analysis to an OEF. The period T_0 is then defined as the duration of a beat (i.e. the time between two successive beats). In this harmonic analysis:

- a_{1,f_0} then represents the DFT magnitude at the 4^{th}-note level;
- a_{2,f_0} at the 8^{th}-note level;
- a_{3,f_0} at the 8^{th}-note-triplet level ...

while:

- $a_{\frac{1}{2},f_0}$ represent the binary grouping of the beats
- $a_{\frac{1}{3},f_0}$ the ternary one

Peeters considers that the vector $\underline{a}$ is representative of the specific rhythm and that therefore $\underline{a}_{f_0}$ represents a specific rhythm played at a specific tempo f_0 (in this context, tempo is assimilated to the fundamental frequency). He proposes the following harmonic series: $h \in \{\frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, 1, 1.25, 1.33, \ldots, 8\}$.

With the above-mentioned considerations, he shows:

- in (Peeters, 2011) that given the tempo f_0, the vector $\underline{a}_{f_0}$ can be used to classify different rhythm pattern;
- in (Peeters, 2006), that given manually-fixed prototype vectors $\underline{a}$, it is possible to estimate the tempo f_0 (looking for the f such that $\underline{a}_f \simeq \underline{a}$);
- in (Peeters, 2010) that the prototype vectors $\underline{a}$ can be learned (using simple machine learning) to achieve the best tempo estimation f_0.

An example of this harmonic representation of rhythm is described in Figure 4.1.

The method we propose is in the continuation of this last work: learning the values $\underline{a}$ to estimate the tempo or the rhythm-oriented genre class. We want to adapt $\underline{a}$ to the deep learning formalism proposed by Bittner et al. (2017).

4.2.2 *Adaptation to a deep learning formalism*

In (Bittner et al., 2017), a task of fundamental frequency estimation in polyphonic music is achieved. To this aim, the depth of the input to a convolutional network is used to represent the harmonic series $\underline{a}_f$ and f_0 denotes the fundamental frequency. Bittner et al. (2017) propose in a first step to compute the CQT of a harmonic signal.

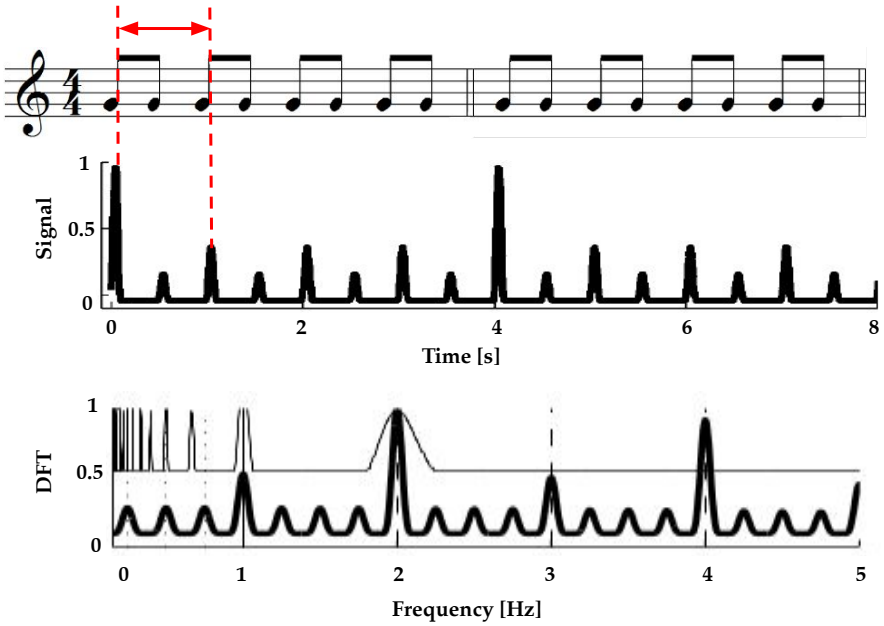

Figure 4.1: Example of a harmonic representation of rhythm components of an onset energy signal. Each beat is divided into 8^{th}-notes. The DFT of the OEF is represented at the bottom where vertical dashed lines represent a_{h,f_0} with $h = 1, 2, 3, 4$ and the vertical dotted lines represent a_{h,f_0} with $h = \frac{1}{4}, \frac{1}{2}, \frac{3}{4}$. Here, the signal have a tempo of 60BPM (1Hz). *Figure taken from (Peeters, 2011).*

CONSTANT-Q TRANSFORM. In musical audio, the frequencies are logarithmically spaced (according to the tempered scale): the frequencies of adjacent notes are closer together at low frequencies and further apart at high frequencies.

The STFT uses the same temporal length of the analysis window for all frequencies, therefore the same frequency resolution for all frequencies. This may be insufficient to distinguish between the frequencies of adjacent notes at low frequencies and too high at high frequencies. The use of the CQT (Brown and Puckette, 1992) solves this problem by adjusting the temporal range of the window with variable length.

The frequency length of the analysis window is calculated as a function of the f_k frequency considered.

Let the factor $Q = \frac{f_k}{f_{k+1} - f_k}$ be constant in frequency such that: $Q = \frac{f_k}{B\omega} = \frac{f_k . L_k}{C\omega}$, with $B\omega = \frac{C\omega}{L}$ the frequency resolution (or bandwidth at -3dB) and $C\omega$ the window characteristic factor.

The length of the window for each frequency can then be calculated as: $L_k = \frac{Q . C\omega}{f_k}$.

As the STFT extends the Fourier transform to time frame τ, the CQT can be computed over successive time frame τ. The results is then a matrix of size $(time\ \tau, log-frequency\ f)$

In (Bittner et al., 2017) the CQT is expanded to a third dimension which represents the harmonic series $\underline{a}_f$ of each f (with $h \in [\frac{1}{2}, 1, 2, 3, 4, 5]$). When $f = f_0$, $\underline{a}_f$ will represent the specific harmonic series of the musical instrument (plus an extra value at the $\frac{1}{2}f$ position used to avoid octave errors). When $f \neq f_0$, $\underline{a}_f$ will represent (almost) random values.

The goal is to estimate the parameters of a filter such that when multiplied with this third dimension $\underline{a}_f$ it will provide very different values when $f = f_0$ or when $f \neq f_0$. This filter will then be convolved over all log-frequencies f and time τ to estimate the f_0's. This filter is trained using annotated data. In the method, there are actually several of such filters; they constitute the first layer of a CNN. In practice, in (Bittner et al., 2017), the $a_{h,f}$ are not obtained as $|X(hf)|$; but by stacking in depth several CQTs each starting at different minimal frequencies i.e. by multiplying the lowest frequency in the range of the CQT f_{min} by the h coefficient: hf_{min}. A visual identification of f_0 is therefore possible when this transformation is applied on a simple harmonic signal by superimposing the CQT computed using the various values of h. This is illustrated in Figure 4.2. The representation is denoted by HCQT: $X_{hcqt}(f, \tau, h)$.

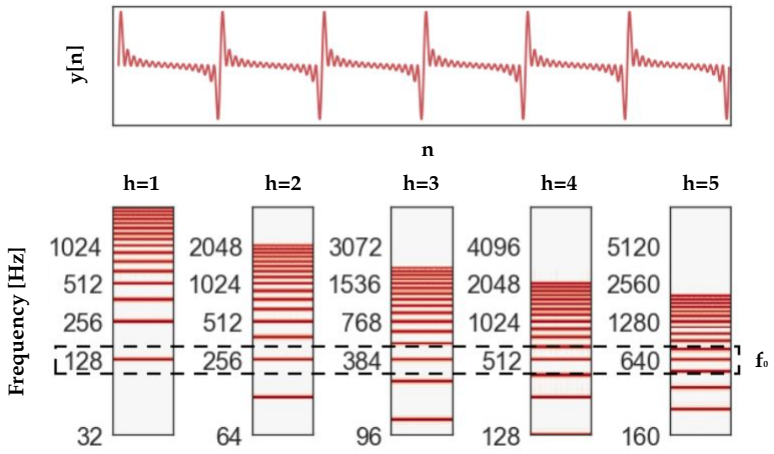

Figure 4.2: Computation of the CQTs of a harmonic signal according to the harmonic series h. The fundamental frequency is bordered by the black dotted rectangle. *Figure taken from Bittner's PhD Thesis.*

The Figure 4.3 illustrates the supervised learning process with the HCQT as input of a CNN and the pitch salience representations (i.e. the perceived energy of frequencies over time) as output.

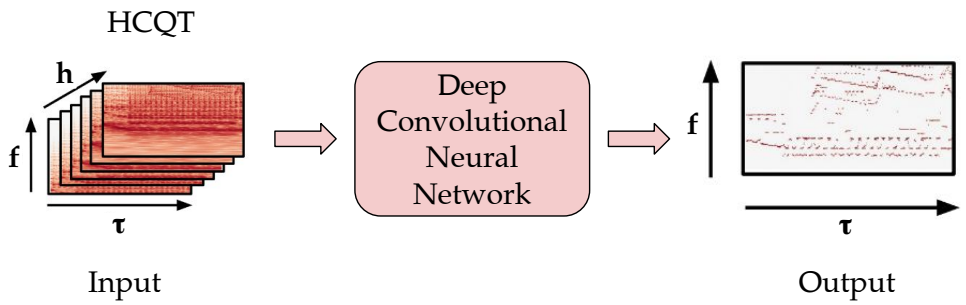

Figure 4.3: Schematic process of training for f0 estimation in polyphonic music using HCQT as input and deep salience as output. *Figure taken from (Bittner et al., 2017).*

4.3 HARMONIC CONSTANT-Q MODULATION

Our goal is to adapt the harmonic representation of the rhythm proposed in (Peeters, 2006; Peeters, 2010; Peeters, 2011) to the deep learning formalism proposed in (Bittner et al., 2017). For this, the HCQT proposed by (Bittner et al., 2017) is not applied to the audio signal, but to a set of OEFs which represent the rhythm content in several acoustic frequency bands. Each of those OEF is a low-pass signal whose temporal evolution is related to the tempo and the rhythm pattern, in this specific band.

We denote our representation by Harmonic Constant-Q Modulation (HCQM). As the Modulation Spectrum (MS) (Atlas and Shamma, 2003), which is a time/modulation-frequency representation, it represents the energy evolution (low-pass signal) within each acoustic frequency band b of a first time/acoustic-frequency (τ/f).

However, while the MS uses two interleaved STFT for this, we use a CQT for the second time/frequency representation in order to obtain a better spectral resolution. Finally, as proposed by Bittner et al. (2017), we add one extra dimension h to represent the content at the harmonics of each modulation frequency ϕ.

We denote it by $X_{hcqm}(\phi, \tau', b, h)$ where τ' denotes the times of the CQT frames, ϕ the modulation frequencies (which correspond to the tempo frequencies), b the acoustic frequency bands and h the harmonic series.

4.3.1 *Computation*

In Figure 4.4, we indicate the computation flowchart of the HCQM. We describe in detail the calculation steps below. For the implementation of the various signal processing steps, we use *librosa* (McFee et al., 2015)[1].

1 https://librosa.org/

1) STFT. Given an audio signal $x(t)$, we first compute its STFT, denoted by $X(f,\tau)$. We keep only its modulus.

2) SUM OVER ACOUSTIC FREQUENCY BANDS. The acoustic frequencies f of the STFT are grouped into logarithmic-spaced acoustic-frequency-bands $b \in [1, B]$. We denote the result by $X(b,\tau)$. The goal of this is to reduce the dimensionality while preserving the information of the spectral location of the rhythm events (kick patterns tend to be in low frequencies while hit-hat patterns in high frequencies).

3) ONSET ENERGY FUNCTION. For each band b, we then compute an OEF over time τ, denoted by $X_o(b,\tau)$. The goal is to keep only the most relevant onsets in each frequency band.

4) CQT. For a specific b, we now consider the signal $s_b(\tau) = X_o(b,\tau)$ and perform the analysis of its periodicities over time τ. One possibility would be to compute a time-frequency representation $S_b(\phi,\tau')$ over tempo frequencies ϕ and time frame τ' and then sample $S_b(\phi,\tau')$ at the positions $h\phi$ with $h \in \{\frac{1}{2}, 1, 2, 3, 4, 5\}$ to obtain $S_b(\phi,\tau',h)$. This is the idea used in (Peeters, 2011). However, in the present work, we use the idea proposed by (Bittner et al., 2017): we compute a set of CQTs, each one with a different starting frequency $h\phi_{min}$. We set ϕ_{min}=32.7Hz. Each of these CQTs gives us $S_{b,h}(\phi,\tau')$ for one value of h. $\tau' \in [0, T]$ is the temporal frame of the CQT applied to a window centered on τ. Stacking them over h therefore provides us with $S_b(\phi,\tau',h)$. The idea proposed by (Bittner et al., 2017) therefore allows to mimic the sampling at the $h\phi$ but provides the correct window length to achieve a correct spectral resolution. We finally stack the $S_b(\phi,\tau',h)$ over b to obtain the 4D-tensors $X_{hcqm}(\phi,\tau',b,h)$.

PARAMETERS. The computation parameters of the HCQM are set such that the CQT modulation frequency ϕ coincides with the tempo value in BPM. We set the STFT parameters as follows: sample rate: $sr = 22,500$Hz; hop size: $N_{hop} = 256$; Hanning window length: $N_{win} = 2,048$; FFT size: $N_{fft} = 16,384$. For the CQT we set the following parameters: $N_{bin} = 240$; N_{bin} per octave $= 60$.

Moreover, we set the CQT window in order to have temporal frames τ' that represent the harmonic content within $\tau = 8s$. This is in order to be able to encapsulate enough information about the rhythmic patterns.

The ranges of the acoustic frequency bands are chosen according to a logarithmic scale, $B = 8$ with:
$b \in [0, 64, 128, 256, 512, 1024, 2048, 4096, 8192]$

Regarding the harmonic series h, we tested two configurations.

The first one is inspired by (Bittner et al., 2017) where $h \in \{\frac{1}{2}, 1, 2, 3, 4, 5\}$ and describes the harmonic content of the audio signal including the sub-harmonics ($\frac{1}{2}$) and the fourth upper harmonics (4). The sub-harmonic is used to avoid octave errors. When $h = \frac{1}{2}$, the HCQT is supposed to represent the content of the audio signal an octave below its frequency, therefore with a low energy. In our case, it will represent the energy of the rhythm content an octave below its tempo, this frequency represents the periodicity at the half-note level. We denote this configuration as H6.

The second one is inspired by (Peeters, 2011) where $h \in \{\frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, 1, 1.25, 1.33, \ldots, 8\}$ and corresponds to the harmonic content of the OEF. In this case, the values of amplitudes at the harmonics do not only depend on the tempo but also depends strongly on the rhythm pattern (starting from the meter periodicity for $\frac{4}{4}$). We denote this configuration as H48.

We discuss the impact of the choice of the number of acoustic frequency bands B and the harmonic series h (H6 or H48) in the evaluation Section 4.6.

4.3.2 *Visual identification of tempo*

For easiness of visualization (it is difficult to visualize a 4D-tensor), we illustrate the HCQM $X_{hcqm}(\phi, \tau', b, h)$ for a given τ' (it is then a 3D-tensor). Figure 4.5 represents $X_{hcqm}(\phi, b, h)$ on a real audio signal with a tempo of 120 bpm. Each sub-figure represent $X_{hcqm}(\phi, b, h)$ for a different value of $h \in \{\frac{1}{2}, 1, 2, 3, 4, 5\}$, (H6). Also for the sake of clarity, we represent the $X_{hcqm}(\phi, b, h)$ side by side (in practice they are stacked together with h as depth). The y-axis and x-axis are the tempo frequency ϕ and the acoustic frequency band b. The dashed rectangle super-imposed to the sub-figures indicates the values $X_{hcqm}(\phi = 120\text{BPM}, b, h)$ which corresponds to the ground-truth tempo.

It is this specific pattern over b and h that we want to learn using the filters W of the first layer of our convolutional network.

4.4 DEEP CONVOLUTIONAL NEURAL NETWORK

4.4.1 *Architecture of the Convolutional Neural Network*

The architecture of the DR network is both inspired by the one from (Bittner et al., 2017) (since we perform convolutions over an input spectral representation and use its depth) and the one from (Schreiber and Müller, 2018b) (since we perform a classification task). However, it differs in the definition of the input and output.

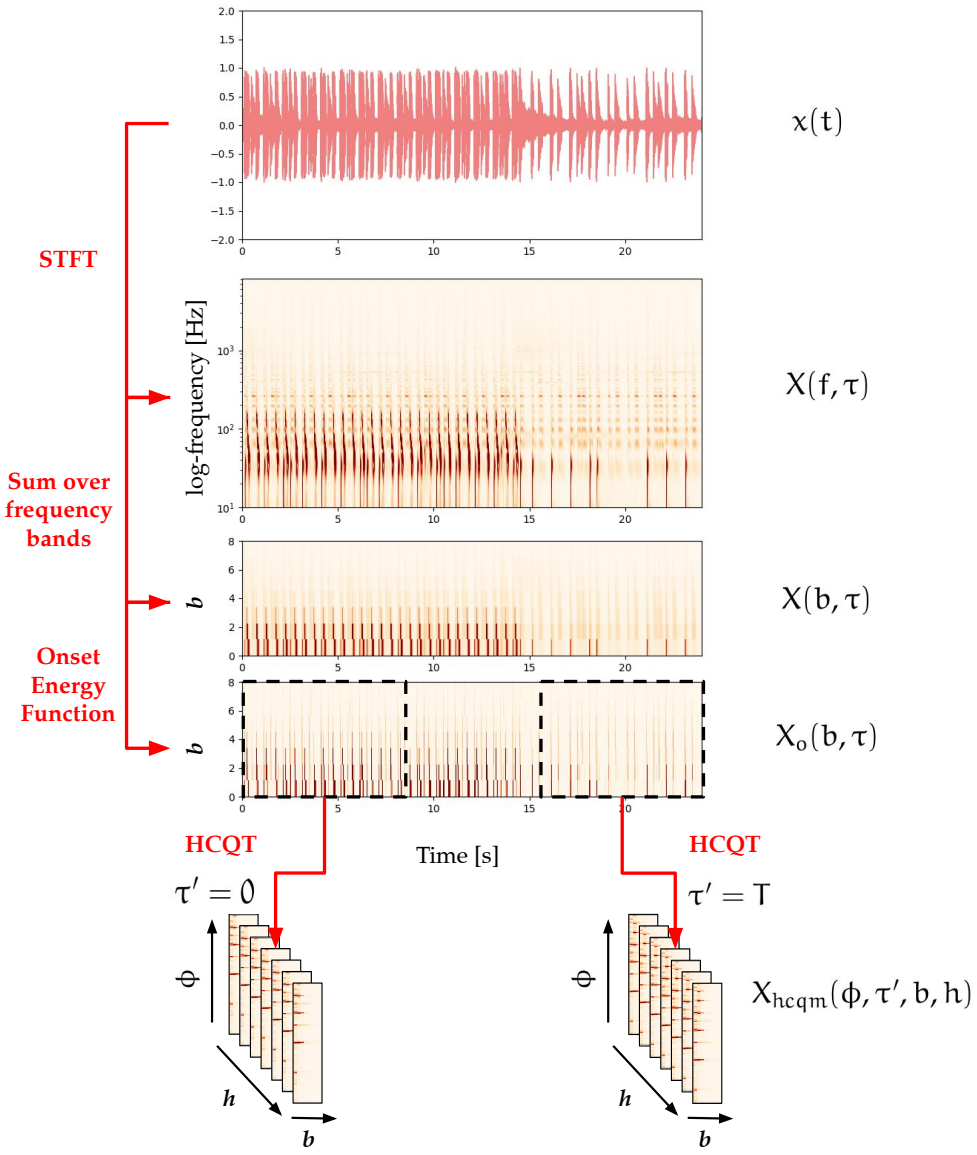

Figure 4.4: Flowchart of the HCQM computation steps for a given techno track excerpt of 24s with a tempo at 120BPM. Here $B = 8$ with H6 configuration and $T = 3$.

INPUTS. In (Bittner et al., 2017), the input is the 3D-tensor $X_{hcqt}(f, \tau, h)$ and the convolution is done over f and τ (with filters of depth H). In our case, the input could be the 4D-tensors $X_{hcqm}(\phi, \tau', b, h)$ and the convolution could be done over ϕ, τ' and b (with filters of depth H). However, to simplify the computation (in term of memory and computation time[2]), we reduce $X_{hcqm}(\phi, \tau', b, h)$ to a sequence over τ' of 3D-tensors $X_{hcqm}(\phi, b, h)$. We denote these inputs as τ'-HCQM. The convolution is then done over ϕ and b with filters of depth H.

2 The memory of the GPU servers is limited and 4D-convolutions are too costly in memory.

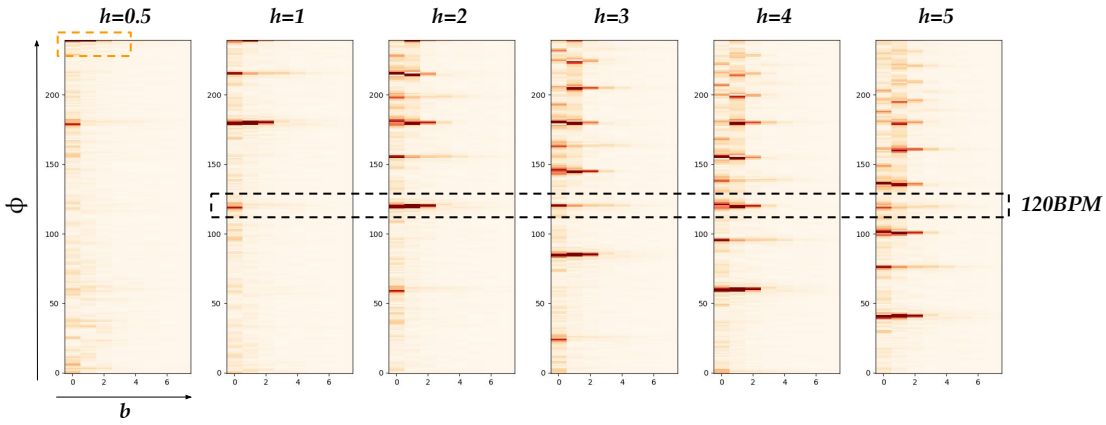

Figure 4.5: HCQM example for a given temporal frame τ' of a techno track excerpt. ϕ denotes the modulation frequency (associated with a candidate tempo), b the acoustic frequency band and h the harmonic coefficient. The tempo is visually identifiable at 120BPM by superimposing its rhythmic harmonic components through h. For $h = 0.5$, the tempo is detected at its sub-harmonic (orange dotted lines) that corresponds to the value 240BPM.

Our goal is to learn filters W narrow in ϕ and large in b which represent the specific shape of the harmonic content of a rhythm pattern. Convolution are pursued over b because the same rhythm pattern can be played with instrument transposed in acoustic frequencies (lower or higher tuning).

OUTPUT. The output of the network proposed by (Bittner et al., 2017) is a 2D representation which represents a saliency map of the harmonic content over time. As in (Schreiber and Müller, 2018b), we consider the tempo estimation problem as a classification problem (instead of a regression one) into 256 tempo classes ranging from 30 to 286BPM. In our case, the outputs are either the $C = 256$ classes of tempo or the $C =$ genre classes number for rhythm-oriented genre classification. For instance, $C = 9$ genre classes for gEBR, $C = 6$ for Gr or $C = 11$ for sBP. To do so, we added at the end of the network proposed by (Bittner et al., 2017) two dense layers, the last one with C units and a softmax activation function.

ARCHITECTURE. In Figure 4.6, we indicate the architecture of our DR network. The input is a τ'-HCQM. The first layer is a set of 128 convolutional filters of shape $(\phi = 4, b = 6)$ (with depth H). As previously mentioned, the convolution is done over ϕ and b. The shape of these filters has been chosen such that they are narrow in tempo frequency ϕ (to precisely estimate the tempo) but cover multiple frequency acoustic bands b (because the information relative to the tempo/ rhythm

is found in several bands)[3]. As illustrated in Figure 4.6 [Left] the goal of the filters is to identify the pattern over b and h specific to ϕ = tempo frequency.

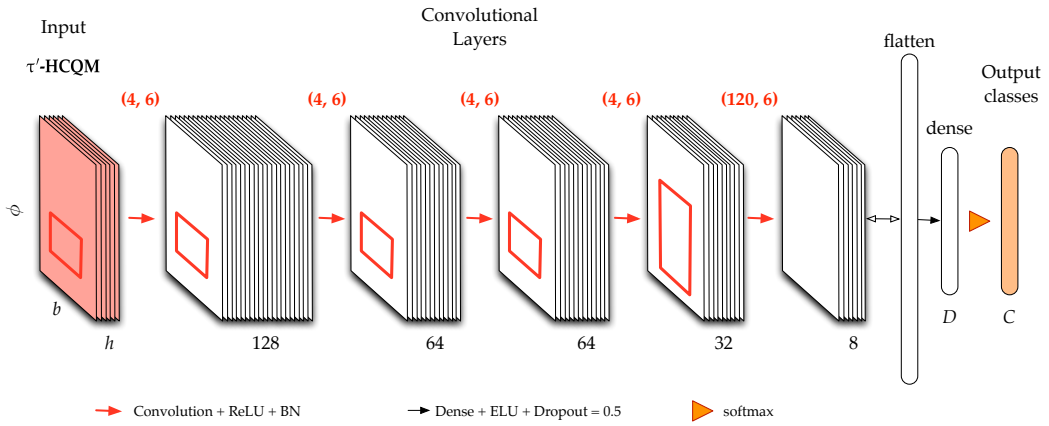

Figure 4.6: DR model Architecture with τ'-HCQM as input (of size (ϕ, b, h)), the size of convolutional filters is indicated in red $(\texttt{filter\_height}, \texttt{filter\_width})$, the number of kernel is indicated in black under each convolutional layer, D denotes the number of units in the dense layer while C is the number of units in the output layer (i.e. the classes logits).

The first layer is followed by two convolutional layers of 64 filters of shape $(4, 6)$, one layer of 32 filters of shape $(4, 6)$ and finally one layer of 8 filters of shape $(120, 6)$ (this allows to track down the relationships between the modulation frequencies ϕ). The output of the last convolution layer is then flattened and followed by a dropout with $p = 0.5$ (to avoid over-fitting (Srivastava et al., 2014)), a fully-connected layer of 256 units, and a last fully-connected layer of C units.

All layers are preceded by a batch-normalization layer (Ioffe and Szegedy, 2015). We used Rectified Linear Units (ReLU) (Nair and Hinton, 2010) for all convolutional layers, and Exponential Linear Units (ELU) (Clevert, Unterthiner, and Hochreiter, 2016) for the first fully-connected layer.

4.4.2 *Training*

The inputs of our network are the 3D-tensors τ'-HCQM of the music track. The datasets we will use for our experiments only provide **global** tempo[4] or **global**

3 Using MIR domain knowledge to choose filter shapes have been explored in details by Pons (Pons and Serra, 2017). They show that the use of a rectangular shape for filters is suitable for tasks such as temporal (hence rhythm) or spectral detection. We therefore choose a directional filter size according to the HCQM representation of the tempo.

4 It should be noted that this does not always correspond to the reality of the track content since some of them have a tempo varying over time (tempo drift), have a silent introduction or a break in the middle as we have disscused in Chapter 3.

rhythm classes as ground-truths. We therefore have several τ'-HCQM for a given track which are all associated with the same ground-truth (multiple instance learning).

LOSSES. We consider the tempo estimation and the rhythm-oriented genre classification tasks as single-label classification problems. In the case of tempo estimation, the classes are defined as $c \in [1, 256]$, for genre classification they are defined as $c \in [1, C]$ with C the number of genre classes in the given dataset.

For the output dense layer, a softmax activation function is generally used for single label classification and is often associated with a categorical cross-entropy as loss-function between the predicted class $\hat{y}_c$ and the ground-truth y_c:

$$\mathcal{L} = -\sum_{c=1}^{C} y_c \log(\hat{y_c}) - (1 - y_c) \log(1 - \hat{y_c}) \tag{4–14}$$

where $c \in [1, C]$ refers either to the 256 tempo classes or to the rhythm-oriented genre classes.

HYPER-PARAMETERS. We used the ADAM (Kingma and Ba, 2014) optimizer to find the parameters of the network with a constant learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e-8$. We used mini-batches of 256 τ'-HCQM with shuffle and a maximum of 20 epochs using early-stopping. The number of parameters θ is around 1,600,000. In terms of computing speed, we note that the model reaches 90% train accuracy between 2 and 5 epochs for tempo estimation and genre classification.

4.5 AGGREGATING DECISIONS OVER TIME

It is important to consider here that the DR network processes each temporal frame τ' independently. We denote by $x_{\tau'}$ the segment of the audio signal centered on time τ and of 8 s duration. During training, a Multiple Instance Learning paradigm is considered: each $x_{\tau'}$ is seen as an instance of the single global ground-truth tempo T_{BPM} and the network trained accordingly. For testing, the output of the network (the softmax output) provides for each $x_{\tau'}$ a tempo likelihood vector $p(T_{BPM}|x_{\tau'})$ which represents the likelihood of each tempo T_{BPM}. The average over frame τ' of this vector is computed, $p(T_{BPM}) = \int_{\tau'} p(T_{BPM}|x_{\tau'}) d\tau'$, and used to estimate the global tempo: $\hat{T}_{BPM} = \arg\max_{T_{BPM}} p(T_{BPM})$. During our experiments, we found that calculating the median rather than the mean of the softmax output vectors slightly improves the results. This also echoes previous

works on tempo estimation (Peeters, 2006; Percival and Tzanetakis, 2014; Böck, Krebs, and Widmer, 2015).

4.5.1 *Oracle Frame Prediction*

Can we find a better way to infer T_{BPM} from the sequence of tempo likelihood vectors $p(T_{BPM}|x_{\tau'})$?. We do this by proposing an attention mechanism below. Before doing so, we would like to know what would be the upper bound achievable by DR to predict T_{BPM} from the succession of $p(T_{BPM}|x_{\tau'})$. To do so we define an Oracle Frame Predictor. This oracle knows which is the best frame τ' to be used to predict T_{BPM}. We denote this best frame by τ'^*. The oracle defines the best frame as $\tau'^* = \arg\min_{\tau'}(T_{BPM} - \arg\max_{T_{BPM}} p(T_{BPM}|x_{\tau'})^2$. It is important to note that the final prediction of the oracle still uses the tempo likelihood vector to estimate the tempo (but only using the best frame): $\hat{T}^*_{BPM} = \arg\max_{T_{BPM}} p(T_{BPM}|x_{\tau'^*})$.

Typically, if the track only contains a single frame corresponding to T_{BPM} and if the network is performing well, the Oracle should be able to find τ'^* and the corresponding $\hat{T}^*_{BPM}$ would be a good estimation. In the contrary, the average value $p(T_{BPM})$ will be blurred and $\hat{T}_{BPM} = \arg\max_{T_{BPM}} p(T_{BPM})$ would provide a wrong prediction. Hence $\hat{T}^*_{BPM}$ is an upper bound.

4.5.2 *Attention Mecanism*

PRINCIPLE. Given the tempo likelihood vectors $p(T_{BPM}|x_{\tau'})$ at each frame τ', our goal is to train an attention mechanism to estimate the single global tempo T_{BPM}.

An *attention mechanism* takes as input a sequence $\{y_1, \dots y_i \dots y_N\}$ and a context c and returns a weighted sum of the y_i: $y = \sum_{i=1}^{N} \alpha_i(c) y_i$. The attention weights $\alpha_i(c)$ defines the importance of each y_i in the context c. They are trained as the other parameters of a network.

Attention mechanisms have been previously used in MIR especially in the case of training with weakly-labeled data (for which the ground-truth only indicates the presence of the label in a document/media without providing its location in it, which is our case). For example, Kong et al. (2019) study the use of different types of attention mechanisms. Those are applied to the output features of a VGG-like network trained for tagging the AudioSet (a weakly labeled dataset (Gemmeke et al., 2017)). Their output features represent information at the frame level and are concatenated along time to be used as an input to the attention network. They then propose 3 different architectures among which a basic one called "decision level single attention". This basic architecture first applies a set of embedding layers followed by the attention mechanism itself. The latter is made of two branches.

The *attention branch* has a softmax activation which is then normalized. It provides the temporal attention weights $\alpha_i(c)$ (here c is the output of the embedding layers). The *prediction branch* has a sigmoid activation and provides the y_i. Both are then used to compute the timeless output prediction vector (the number of dimension is the number of classes): $y = \sum_{i=1}^{N} \alpha_i(c) y_i$.

The idea of this method is to automatically learn the labels present at the frame level from a global annotation of an audio track.

ADAPTATION TO DEEP RHYTHM. The flowchart of our attention mechanism is illustrated in the bottom right part of Figure 4.7. The input of the attention mechanism is the sequence of tempo likelihood vectors $p(T|x_{\tau'})$ each of dimension 256 (the number of tempo classes). Considering that the duration of music track can vary, zero-padding is applied to store them in a matrix of size (45×256) (45 being the maximum number of frames). These features are then used as input to train an attention mechanism network. We tested all the different architectures presented in Kong et al., 2019 and different activation functions (sigmoid/softmax) either applied at the prediction or attention layer.

The chosen AM architecture corresponds to the multi-attention architecture proposed by Kong et al., 2019. Input features are the predictions taken from the penultimate layer of the DR. It is followed by 3 embedding layers. An attention/prediction layer pair (with a softmax/sigmoid, respectively, as activation function) is applied at the output of the second and the third embedding layers. Finally, these two pairs of layers are concatenated and connected to a dense layer with the size of the number of tempo classes and a softmax as output function.

In order to adapt as well as possible to our problem, we carried out a series of experiments by modifying certain parameters. Since we have vectors of softmax as features and our method is a single-class classification, we replaced the sigmoid activations of the prediction layer by a softmax. We also tried to use the penultimate layer of the DR model as a feature while keeping the prediction activation functions the same (e.g. sigmoid). The results presented in Section 4.6.1.4 are those obtained with this last parameterization as they are the highest we have obtained during our experiments.

4.6 EVALUATION

We evaluate the proposed DR method including the HCQM representation combined with the CNN for the task of global tempo estimation on one side and for the task of rhythm-oriented genre classification on the other side. We used the same system (same input representation and same network architecture) for both tasks. However, considering that the class definitions are different from a dataset

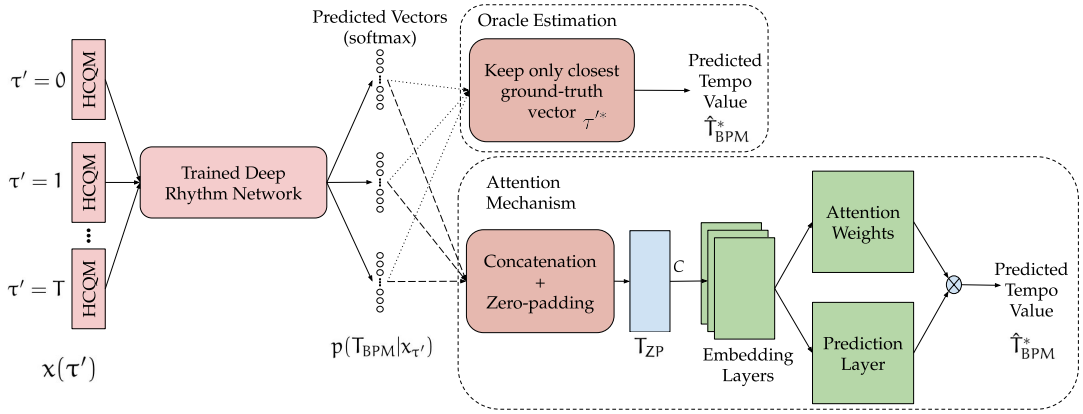

Figure 4.7: Time-varying tempo strategies flowcharts. [top] Oracle Frame Prediction, [bottom] Attention Mechanism with T_{ZP} the number of temporal frame after zero-padding application.

to another one, we performed two independent trainings of the DR model. Thus, the evaluation protocol diverges for the two tasks and the results are presented independently.

4.6.1 *Tempo Estimation*

4.6.1.1 *Evaluation protocol*

To be able to compare our results with previous works, we use the same paradigm (cross-dataset validation[5]) and the same datasets as (Schreiber and Müller, 2018b) which we consider here as the state of the art.

TRAINING SET. As indicated in Chapter 3, the training set is the combination of the LMD, the tMTG and the EBR datasets.

The total size of the training set is 8596 tracks. It covers multiple musical genres to favor generalization.

TESTING SET. For the testing set, we also define the same datasets as in (Schreiber and Müller, 2018b): ACM, BR, tGS, GTzan, Hains., ISMIR04, SMC and finally Combined which denotes the union of all test-sets. Again those state-of-the-art datasets cover various musical genres to show the relative effectiveness of the results.

METRICS. Among the metrics used to evaluate the performance of the global tempo evaluation, we choose the following ones:

5 Cross-dataset validation uses separate datasets for training and testing; not only splitting a single dataset into a train and a test part.

- *Class-Accuracy (Acc):* it measures the ability of our system to predict the correct tempo class (in our system we have 256 tempo classes ranging from 30 to 285BPM);
- *Accuracy1 (Acc1):* it measures if our estimated tempo is within ±4% of the ground-truth tempo;
- *Accuracy2 (Acc2):* is the same as Accuracy1 but considering octave errors as correct (global tempo estimated at two or three times the value of its ground truth or at half or a third of this value within ±4% window).

These metrics allows us to compare the results of our work with those of our peers.

4.6.1.2 *Results and comparison to the state of the art*

We compare our results to the state of the art represented by the 3 following systems[6]:

- *sch1* denotes the results published in (Schreiber and Müller, 2017);
- *sch2* in (Schreiber and Müller, 2018b);
- *böck* in (Böck, Krebs, and Widmer, 2015).

Results are provided in Figure 4.8.

DISCUSSION. For ACM, Hains. and GTzan, results are quite similar compared to the other methods in terms of the three accuracies. We can interpret those results by the good generalization of our method when it is applied to a popular music dataset.

For the Combined dataset, DR results are always closer to the best state of the art method (*sch2*) in terms of *Acc* and *Acc1*. In terms of *Acc2* the *böck* method outperforms the others. One reason for this is that our network (as the ones of sch1 and sch2) is trained to discriminate BPM classes, therefore it doesn't know anything about octave equivalences which are taken into account in *böck* method and in *Acc2*. Except for the SMC dataset (which contains rhythm patterns very different from the ones represented in our training sets), the DR results are quite similar to the state of the art methods, especially the *sch2* one.

We can observe that in terms of *Acc*, the DR method have the best score for the BR dataset (and similar to *sch2* for *Acc1* and *Acc2*). In terms of *Acc1* and *Acc2*, the DR method performs better for the tGS EDM dataset. **These results partly confirm our hypothesis: for rhythm-oriented genres such as ballroom music or EDM, our system is particularly efficient in terms of tempo estimation.** The harmonic

6 Those systems are described in Section 2.4

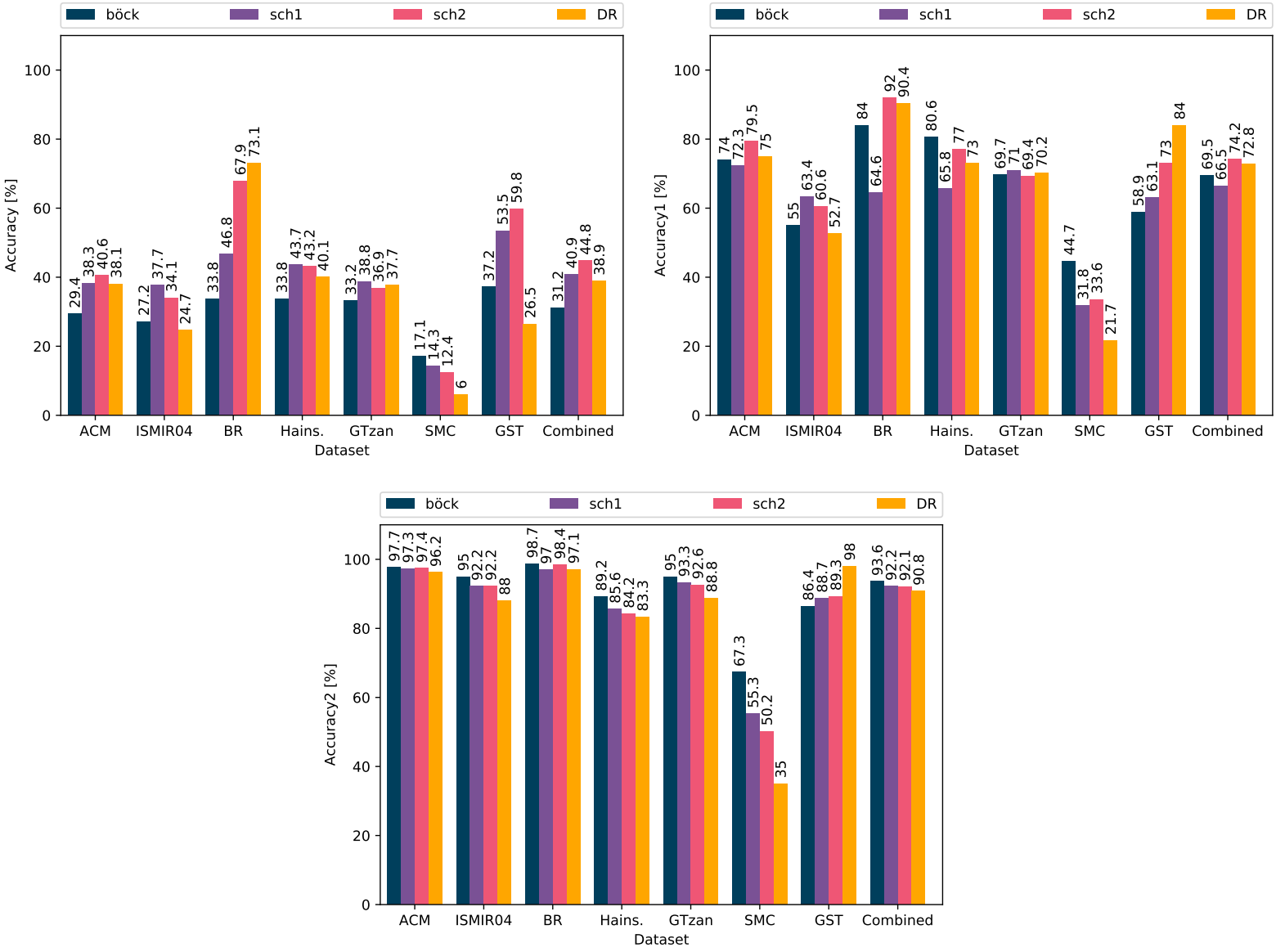

Figure 4.8: DR results scores compared to state-of-the-art methods. [Top left] in terms of *Acc*, [Top Right] in terms of *Acc1*, [Bottom] in terms of *Acc2*.

characteristics of the rhythm included in the HCQM are suitable to obtain relevant estimation of the tempo. For these results, we used a H48 type h harmonic series.

4.6.1.3 *Validation of the HCQM parameters*

Through experiments, we have validated the use of certain parameters, for network training or for the calculation of the HCQM. Among the most important ones: the harmonic series h and the number of acoustic frequency bands B.

In Figure 4.9, we compare the following settings in terms of *Acc*, *Acc1* and *Acc2* only for the Combined dataset:

- H48: same parameters of the DR method as used for Figure 4.8, $h \in \{\frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, 1, 1.25, 1.33, \ldots, 8\}$ and $B = 8$;
- H6: $h \in \{\frac{1}{2}, 1, 2, 3, 4, 5\}$ and $B = 8$;
- H1: no depth i.e. $h = 1$ and $B = 8$;
- B1: sum over all acoustic frequency bands, H6 and $B = 1$.

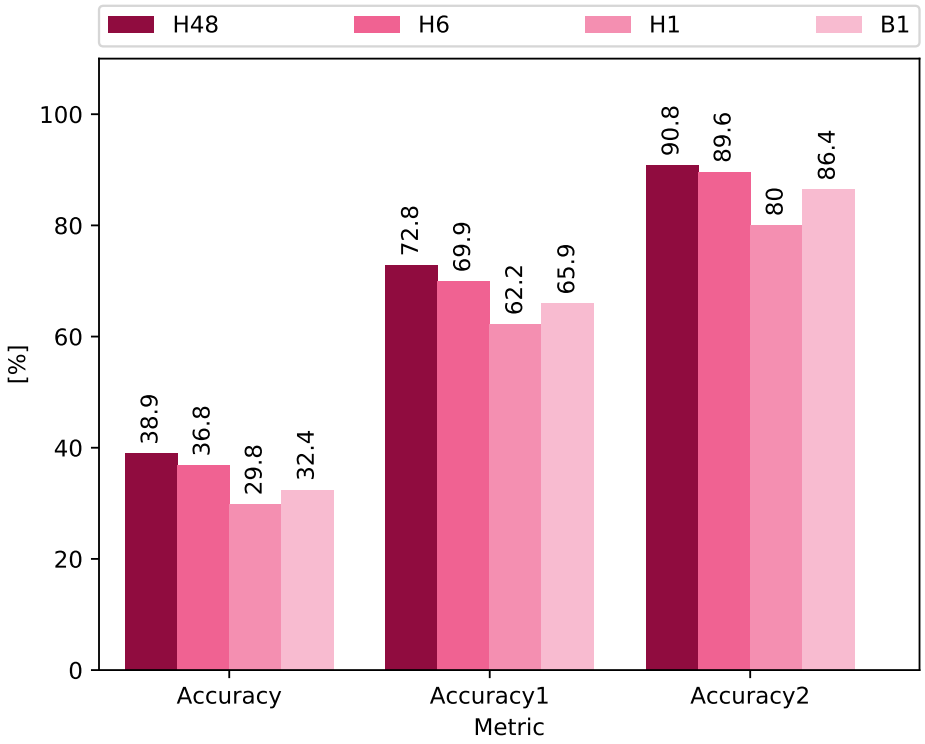

Figure 4.9: Validation of H and B parameters, results of DR for Combined dataset.

DISCUSSION. Exploiting this comparison, we first observe that the H48 strategy works better than the H6 one for tempo estimation. It can be explain by the fact that using such a harmonic series leads to deeper representation of the components of rhythm (including the harmonics of the meter frequency) and the network learns more precise information that allows it to better estimate the tempo. Then, we can notice the importance of the depth h when comparing the H48 and H6 strategies with H1. Finally, we show the significance of the decomposition into acoustic frequency bands since H48 and H6 largely outperform B1.

4.6.1.4 *Time varying tempo*

We now evaluate our temporal frame selection methods described in Section 4.5.

To do so, we compare in Figure 4.10 the results obtained with:

- Oracle-DR, that denotes the Oracle Frame Prediction. As a reminder, in this method we chose the best prediction softmax vector among all the temporal prediction vectors.
- AM-DR, that denotes the Attention Mechanism applied on top of the DR method. We trained an AM on the trained prediction vectors in order to learn to automatically select the best prediction softmax vector.
- DR, that denotes our baseline (H48).
- *sch2*, that denotes the results published in (Schreiber and Müller, 2018b), we keep this state of the art method among the others because it is the most recent and it has the best results in terms of Accuracy1 for the Combined dataset.

DISCUSSION. The results obtained with Oracle-DR method are given in Figure 4.10. It provides us with an upper-bound to reach. High scores in terms of *Acc2* directly reflect the presence of a good estimate in the prediction set obtained from the τ'-HCQM of a given track (except for the SMC dataset). The Oracle-DR method outperforms the DR one (for which the various predictions for a given track are averaged before the maximum selection) and the *sch2* method in terms of the three metrics *Acc*, *Acc1* and *Acc2*.

For the AM method, we observe results similar to the DR baseline for the three accuracies (although slightly higher for 5/7 datasets in terms of *acc*). These results do not allow us to conclude on a significant gain compared to the baseline. We explain these results by the fact that the model fails to generalize. Indeed, the softmax vectors are used as attention input independently of the DR network. We want the model to learn how to generalize the selection of the right time frame corresponding to the right tempo candidate from these vectors. Given the multitude of dataset and the difference between the audio examples composing them (size, presence of silence or not, audio extracted at various moments of the track), this does not allow the model to clearly learn the tempo candidate specific positions. We believe that the automation of the selection of this candidate is currently limited by the size of the memory and the computing time of the machines at our disposal. Indeed, we suppose that the use of 4D-convolution in a network with the attention mechanism as an output (in an end-to-end way) would make this automation possible.

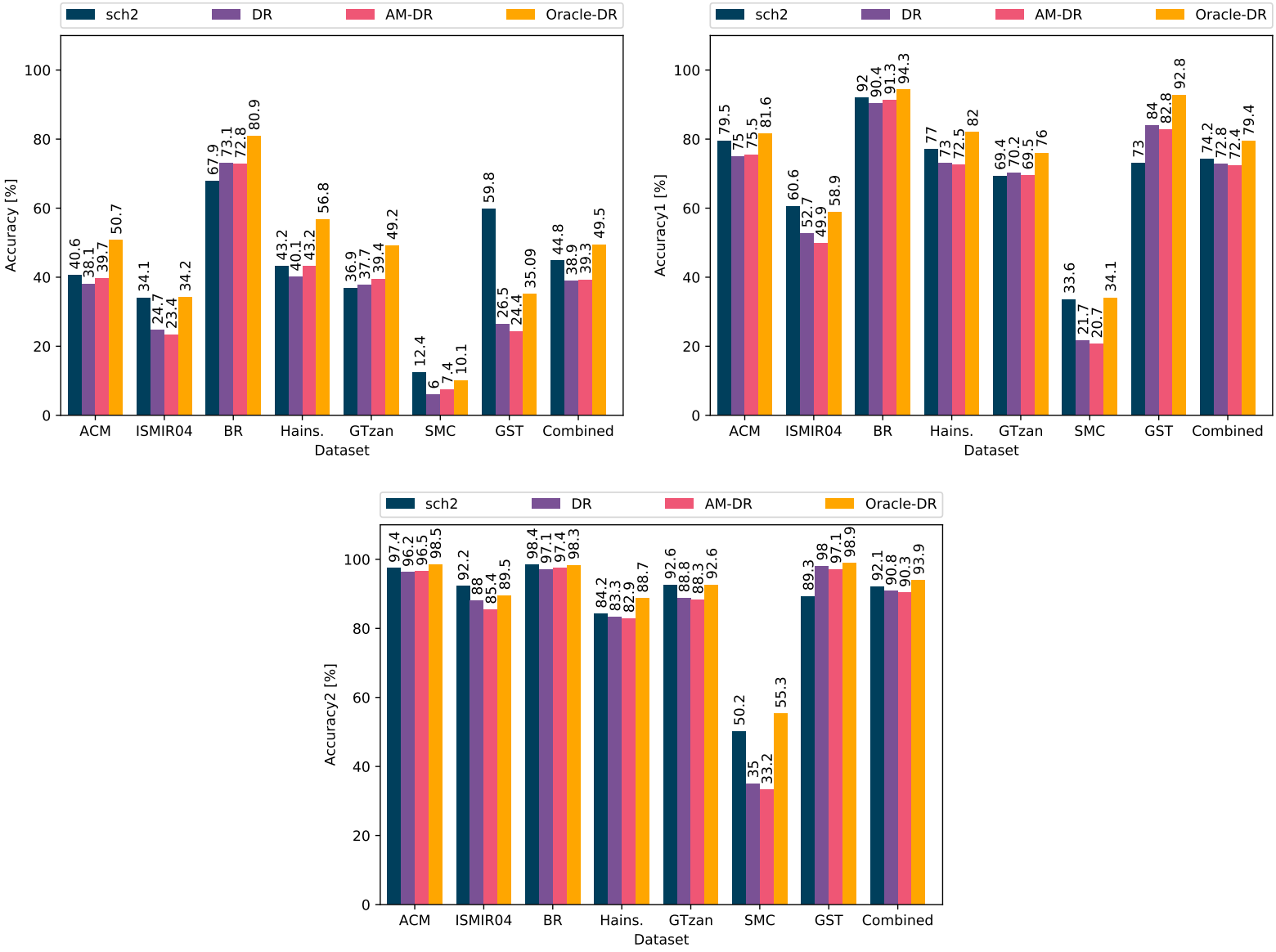

Figure 4.10: Results of time aggregation strategies, Oracle-DR and AM-DR, compared to our baseline DR and a state-of-the-art method *sch2*. [Top left] in terms of *Acc*, [Top Right] in terms of *Acc1*, [Bottom] in terms of *Acc2*.

4.6.1.5 *Ground-truth Smoothing*

PRINCIPLE. Ground-truth smoothing of the tempo annotations have been proposed by Böck, Davies, and Knees (2019). The original tempo classification is based on reducing the error between a prediction softmax vector (probability vector) and a one-hot vector with the ground-truth value at 1 and the others at 0. The idea of smoothing is to extend the range of the global tempo annotation values to ±2BPM

by weighing the neighboring BPM by 0.5 and 0.25 respectively. Normalizing those weighting values allows to form a probability distribution suitable for the application of a softmax activation function (Figure 4.11).

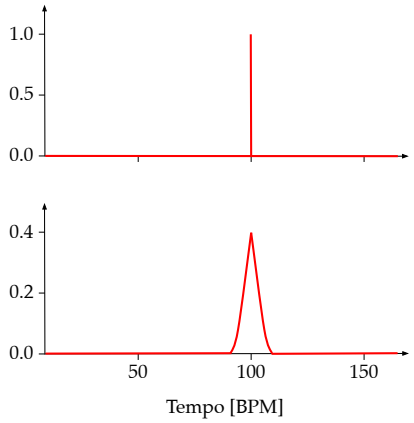

Figure 4.11: Ground-truth smoothing principle. [Top] original ground-truth annotation. [Bottom] Smoothed ground-truth.

As described in Chapter 3, to obtain tempo annotations manually, a panel of listeners is often asked to "tap" the tempo of an audio piece. A margin of error is then tolerated knowing that the perceived tempo may not be accurately equal to the exact BPM (or its integer value). Such a margin also exists for algorithms since they rarely detected a precise BPM value as shown by the use of the metric *Acc1*. This smoothing applied when learning the model allows it to learn how to extend the tempo estimation to its close neighborhood.

In the same vein, we therefore propose an adaptive smoothing of the ground truths. This time, the tempo range for a given value is modified by multiplying the ground-truth by a factor of $\pm 4\%$ in accordance with *Acc1*.

We present the results in Figure 4.12, the methods compared are the Smooth-DR (fixed ground-truth smoothing) and the Ada-smooth-DR (adaptive ground-truth smoothing)

DISCUSSION. First, in terms of *Acc*, we can observe slightly lower results for the Ada-smooth strategies compared to DR (except for Gtzan and SMC) while the Smooth-DR performs better for 5/7 of the datasets and for the Combined one.

In terms of *Acc1* and *Acc2*, the Ada-smooth-DR shows the best results for the Combined dataset, higher than *sch2*. However, by individually comparing the results on the datasets, the two smoothing strategies are quite equivalent.

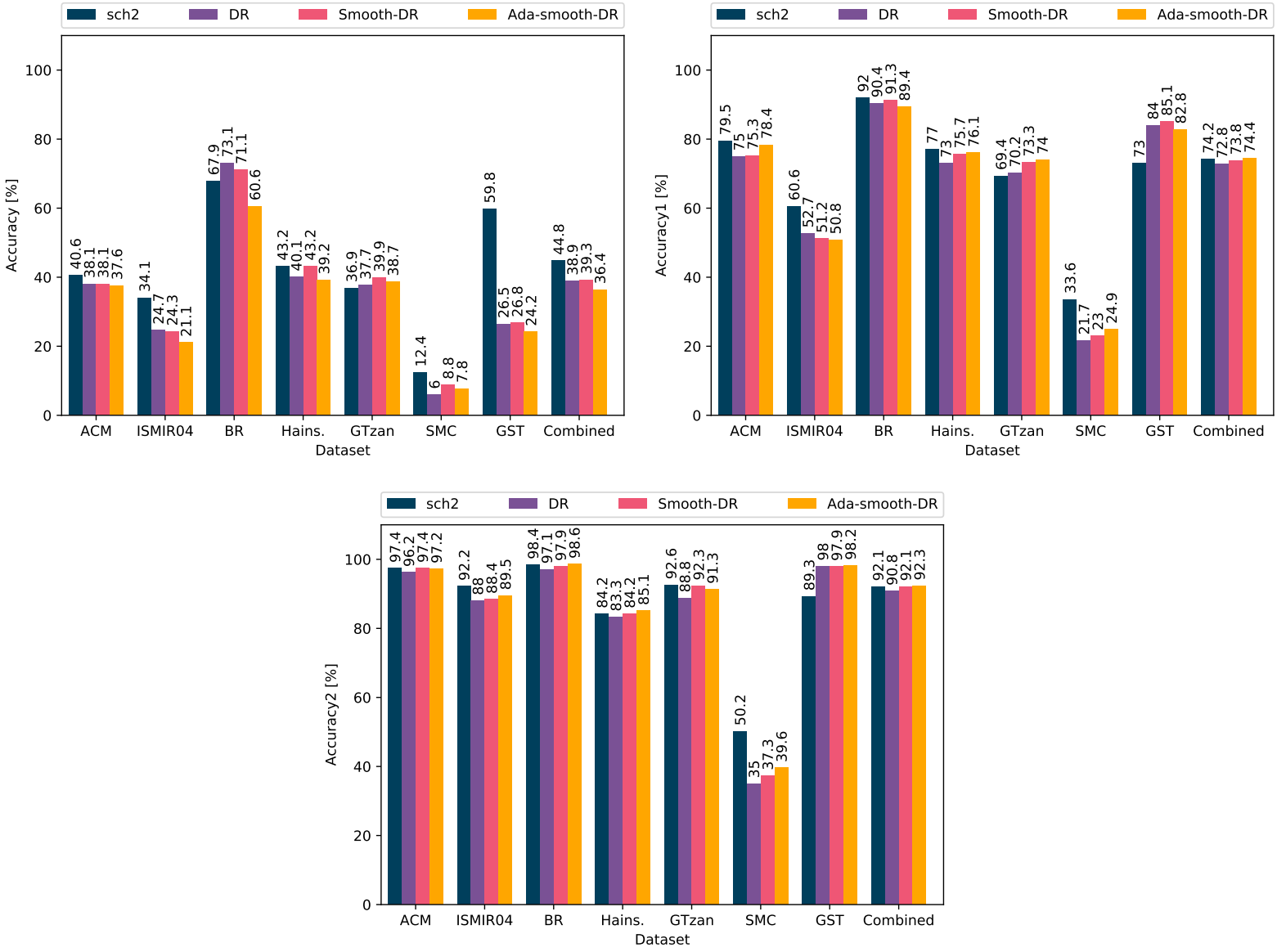

Figure 4.12: Results of the adaptive smoothing of the tempo ground-truth values compared to our baseline DR and a state-of-the-art method *sch2*. [Top left] in terms of *Acc*, [Top Right] in terms of *Acc1*, [Bottom] in terms of *Acc2*.

We can explain the lack of precision in *Acc*. Since ground truths are smoothed, the estimated softmax probabilities are unavoidably impacted and this results in a loss of accuracy in terms of exactness.

Regarding these results, the Smooth-DR becomes our new baseline method.

4.6.2 *Rhythm-oriented genre classification*

4.6.2.1 *Evaluation protocol*

TRAINING/TESTING SETS. For the rhythm-oriented genre classification, it is not possible to perform cross-dataset validation (as we did for tempo estimation) because the definition of the genre classes is specific to each dataset. Therefore we perform a k-fold cross-validation on the various datasets independently.

Given a dataset, we separate all the data into k-folds. Each fold is successively used as testing data set while the remaining $k-1$-folds are used to train the model. As in the work of (Marchand and Peeters, 2016a), we choose $k = 10$ to perform a 10-fold cross-validation. More precisely, we use a stratified k-fold where the folds are made by preserving the percentage of samples for each class[7].

We choose to evaluate 3 datasets – BR, gEBR and Gr – in order to compare our results to the state-of-the-art method of Marchand and Peeters (2016a) which performs a rhythm-pattern recognition. It means that the rhythm-oriented genres within those 3 datasets are also defined by the rhythmic patterns that are played in different tempo ranges.

Next, we study the results on EDM datasets gMTG, lBP and sBP. We do not compare the results with any previous work because none of them uses the same data as we do for automatic genre classification.

METRICS. We measure the performances using the average-mean-recall $\hat{R}$[8]. The recall is computed as: $Recall = \frac{\text{True Positive}}{\text{True Positive+False Negative}}$. For a given fold f, the mean-recall $\hat{R}_f$ is the mean over the classes c of the class-recall $R_{f,c}$. The average mean-recall $\hat{R}$ is then the average over f of the mean-recall $\hat{R}_f$. We also present the confusion matrices of each dataset evaluated to better understand the statistical results.

4.6.2.2 *Results and confusion matrices*

STATISTICAL RESULTS. The results are indicated in Table 4–2. We compare our results[9] to the ones of Marchand and Peeters (2016a), considered here representative of the current state of the art for rhythm-pattern recognition.

DISCUSSION. Our results are largely lower the ones of Marchand and Peeters (2016a) for Gr. For BR they are slightly below while for gEBR slightly above.

7 It is important to mention that in all methods of this manuscript we use the same 10-fold split for each dataset. This allows us to accurately compare the different methods we have implemented.

8 The mean-recall is not sensitive to the distribution of the classes. Its random value is always $1/C$ for a problem with C classes.

9 The results presented are the ones obtained using the H48 strategy

Table 4–2: Results of rhythm-pattern recognition/rhythm-oriented genre classification in term of average-mean-recall $\hat{R}$.

Dataset	Marchand (2016)	Proposed DR
Ballroom (BR)	**96%**	93.0%
Extended Ballroom (gEBR)	94.9%	**95.4%**
Greek Dance (Gr)	**77.2%**	68.9%
genre MTG (gMTG)	-	37.6%
Beatport large (lBP)	-	40.7%
Beatport small (sBP)	-	52.8%

It should be noted that the "Scale and shift invariant time/frequency" representation proposed in (Marchand and Peeters, 2016a) takes into account the interrelationships between the frequency bands of the rhythmic events while our HCQM does not. Furthermore, our DR method has a data-driven part since we train a CNN, such models have to trained on a sufficient amount of data to increase their efficiency (generalization). The few data contained in the BR and especially Gr datasets may explain these results.

For EDM datasets, we can notice that the results are better for the genre-balanced ones than for the gMTG. As we have seen, the genres in the sBP are specifically selected for their rhythmic characteristics unlike the lBP where all Beatport genres are represented (this includes some rhythmic ambiguities). The score has therefore improved significantly (+12.8%).

CONFUSION MATRICES. The confusion matrices with respect to each evaluated dataset highlight the predicted genre distribution compared to the ground-truth labels. We do not show the gMTG and lBP confusion matrices here since, given their size, they are not subject to a concrete, relevant interpretation. The results are normalized over classes and represent the class prediction averaged over folds f.

First and foremost we can observe that for the datasets BR and gEBR the prediction score is very high regardless of the genre (Figure 4.13).

As for the Gr, despite good scores for 4/6 Greek dance genres represented, the results are more disparate (Figure 4.14). Referring to the paper of Holzapfel and Stylianou (2011), at the origin of the creation of the dataset, we can see that the tempo ranges of *pent* and *sous* overlap which could explain the confusion between these two genres. Moreover, the dataset is initially used as a proof of concept for a so-called "handcrafted" method. Rhythmic similarities are evaluated by comparing the scale transforms of the tracks using a simple distance metric. In our case, even

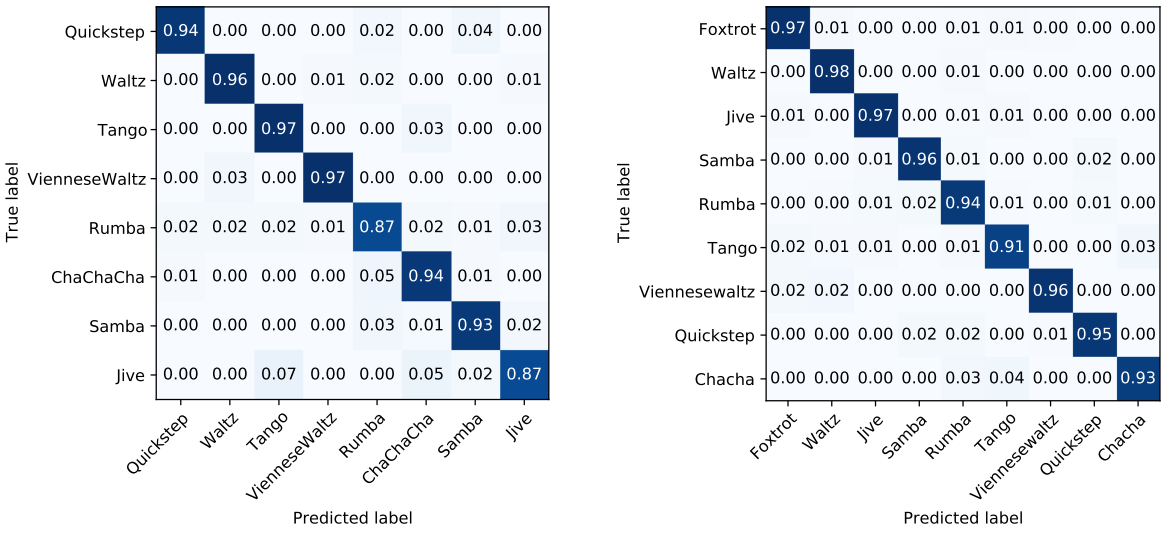

Figure 4.13: Confusion matrices of the evaluation of the BR [Left] and the gEBR[Right] datasets using DR method.

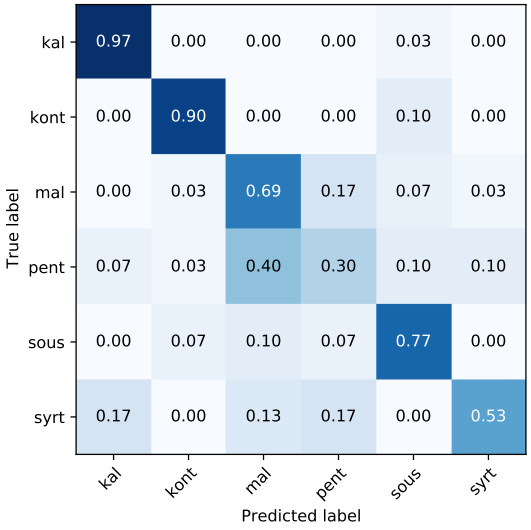

Figure 4.14: Confusion matrix of the evaluation of the Gr dataset using DR method.

though the rhythmic components are represented via the HCQM, the CNN we are training is not fully capable of learning from such a small amount of data.

From sBP's confusion matrix (Figure 4.15), we can see that for all genres the predictions are correct for the most part (i.e. the score of a given class is higher than the false negative score for the same class).

We can observe that the genres of breakbeat rhythmic style are the best classified (*drum-and-bass*, *dubstep*), it is also the case of trance which, as we mentioned in

Figure 4.15: Confusion matrix of the evaluation of the sBP dataset using DR method.

Chapter 3, has a complex rhythmic structure. Looking at the genre classification, we can also point out that the confusions make sense.

For *dance*, the results are the most disparate, the tracks annotated with this label are predicted as *funk-soul-disco* at 15% but also as *house* at 14%. This explains why the *dance* genre is influenced by both these genres and shares some characteristics such as the tempo range and the fact that they all have a four-on-the-floor rhythmic structure.

funk-soul-disco is categorized as *hip-hop* at 21%, which is also due the funk and soul influence in the tracks that are annotated *hip-hop-r-and-b*.

reggae-dancehall-dub is categorized at 17% as "dance", knowing that the barrier between *dancehall* and *dance* is relatively thin. *dancehall*, *reggae* and *dub* are three genres that share a four-on-the-flour rhythmic structure called "stepper". It is also 11% categorized as *dubstep*, a genre with which it shares very similar rhythmic components but also a tempo range perceived often around the same value for *dub* music.

For *techno*, once again, the results make sense since it is categorized at 17% in *hardcore-hard-techno* where the rhythmic structure is the same despite a BPM

often higher for hard-techno. It is also categorized as *house*, knowing that beyond the instrumentation, both are four-on-the-floor style and share a common origin (*disco*, reminding that techno is categorized at 11% in this genre). This can also be noticed by looking at the classification distribution of *house*: 17% in *techno*.

We can conclude by confirming our assumptions about the link between rhythm and genres in electronic music. Indeed, the confusion per genres shows that the classification strongly depends on their rhythmic structure since the combination of the HCQM and the CNN allows to represent and learn it, respectively.

4.7 CONCLUSION

In this chapter, we have proposed the Harmonic-Constant-Q-Modulation (HCQM) representation of rhythm inspired by previous works on handcrafted and data-driven systems. It is defined as a 4D-tensor which represents the harmonic series at candidate tempo frequencies of a multi-band OEF. The 3D τ'-HCQM, which describe temporal slices of HCQM of a given track are then used as input to a deep convolutional network. The filters of the first layer of this network are expected to learn the specific characteristic of the various rhythm patterns contained in the HCQM. We denote this method Deep Rhythm (DR) since the depth of the HCQM rhythmic representation is directly the input depth of a deep network. We have evaluated our approach for two tasks: global tempo estimation and rhythm-oriented genre classification.

For the datasets BR and tGS, the results are higher for the tempo estimation task than the ones obtained with the state-of-the-art methods in terms of exact accuracy (*Acc*) and exact accuracy within a $\pm 4\%$ tolerance window (*Acc1*). The genre of these datasets are strongly defined by the rhythm patterns played at specific tempo ranges. We also showed the relative importance of the HCQM parameters: the number of separation bands B and the harmonic series coefficient h (the depth through which the network filters are convoluted). Within the considered datasets, the tempo inside a track can vary over time. However the ground-truth annotations only provide a single global tempo annotation value. While the mean of the softmax vectors over frame is used as the baseline to estimate this single tempo, we showed that if we consider the best softmax vector (oracle frame prediction method), the performance are actually much higher. We therefore added an Attention Mechanism on top of DR to infer the best single tempo estimation from the sequence of predictions but the results did not live up to our expectations. Still for tempo estimation, we propose the use of smoothed ground-truths tempo annota-

tions. The results slightly increased with this process and outperform the state of the art in terms of *Acc1* and *Acc2*.

For rhythm-oriented genre classification, our method works better for the gEBR dataset compared to the state-of-the-art method of (Marchand and Peeters, 2016a) but doesn't work as well for the BR and the Gr. However, the confusion matrices indicate that our recognition is above 90% for the majority of the classes of the BR and the gEBR. As far as the identification of the genres in EDM is concerned, we have showed the efficiency of a rhythmic approach thanks to the careful observation of the confusion matrix obtained by evaluating the dataset sBP using our DR proposal.

DR method and its evaluation have been published at the ISMIR conference in 2019 (Foroughmand and Peeters, 2019). It should be noted that the results for tempo estimation presented in the paper are not the same since an error was made when retrieving annotations from the ISMIR04 dataset. In addition, for the estimation of the genre, the results took into account the dataset EBR in its entirety and not the gEBR as stated in the state of the art. These results have been corrected in this manuscript.

In the continuation of our work, we have considered three extended DR methods using not only signal processing principles but also more complex deep learning principles. We describe that in the next chapter.

5

DEEP RHYTHM EXTENSIONS

5.1 INTRODUCTION

In the previous chapter, we combined two types of systems in our DR method, namely the handcrafted system with the HCQM representation of rhythm as a knowledge-driven feature and a deep learning network as the data-driven system. In this chapter, we present some extensions of our DR method. The development of each of these representations has as a starting point a musical intuition. For each of these extensions, we present our motivations, the resulting method as well as the analysis of its evaluation.

COMPLEX NETWORK. In the original Deep Rhythm network, the input HCQM and the network do not allow to represent the inter-relationship between the various frequency bands b. This is due to the fact that each OEF is modeled by the modulus of the CQT, and the modulus does not preserve the information of temporal location. Therefore, the network cannot consider the inter-relationship between the various acoustic frequency bands b. To face this, we propose to replace the use of the modulus of the HCQM by a complex-valued HCQM and turn it into an input of a complex CNN This is described in Section 5.2.

MULTITASK LEARNING. In the original DR method, independent systems are trained for the task of tempo estimation and rhythm-oriented genre classification. We propose here a multitask approach where a single system is trained to solve both tasks simultaneously. This is done by defining two losses for the optimization of the system. We believe that using a single network jointly trained for the two tasks would allow sharing information in the network. This is described in Section 5.3.

MULTI-INPUT NETWORK. The DR network was designed to represent the rhythm content of an audio track. As shown previously, the tempo range and possible rhythm patterns are strongly correlated to the music genre of the track (especially in the case of EDM). We therefore study an extension of the Deep-Rhythm by associating it with a second input branch. This second branch is a network dedicated to the representation of timbre with a mel-spectrogram as input. It is based on

a commonly-used network for audio tagging presented in (Choi, Fazekas, and Sandler, 2016). This is described in Section 5.4.

5.2 COMPLEX DEEP RHYTHM

5.2.1 *Why complex representation/convolution?*

A rhythmic pattern can be affiliated to simpler rhythm pattern (quarter note, eighth note, ...) played by different instruments at different frequency bands. Two different rhythm pattern examples are represented in Figure 5.1.

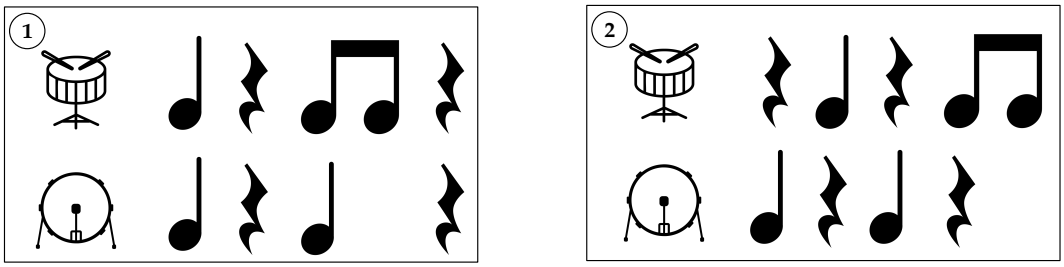

Figure 5.1: Two examples of rhythm patterns.

(1) The bass drum (or kick in the case of an EDM track) and the snare drum are played simultaneously.

(2) The bass drum and the snare drum are played alternately.

As mentioned in Section 4.3, in the HCQM representation, the modulations frequencies ϕ are modeled independently by calculating the HCQT of the OEF in each acoustic frequency bands b. Since only the modulus of the HCQT is computed, the temporal information relative to the rhythmic components is not taken into account. The representation is then passed through a deep CNN. The network does not consider the inter-relationship between the various frequency bands b when it learns the different rhythmic structure to perform the tempo (or the rhythm-oriented genre) estimation. For example, the snare drum and the bass drum being located on two different frequency bands, we can assume that our DR method is not able to differentiate between the two rhythmic patterns illustrated in Figure 5.1. Moreover, ignoring these inter-band relationships impacts the learning of rhythmic components related to tempo within the HCQM.

Marchand et al. deal with the same limitation with their modulation scale spectrum representation (Marchand and Peeters, 2014). They therefore show in (Marchand and Peeters, 2016c) that modeling the inter-relationships between acoustic

frequency bands using inter-band correlation coefficients allows to better estimate the rhythm pattern.

In our case, due to the data-driven aspect of our DR method, we would like to find a way to keep the temporal information of the rhythmic content present in each acoustic band b of the HCQM and furthermore to be able to train a network that includes this information.

In a temporal representation of the frequency evolution such as the STFT or the CQT, the positional information of the windowed signals are contained in the phase of the complex-values.

Recently, complex neural networks have appeared in the MIR field and allow model training on inputs with complex values. These models have proven their efficiency for various tasks such as automatic transcription (Trabelsi et al., 2017) of music or speech enhancement with a deep U-net (Choi et al., 2019). However, they have never been used for classification purposes. Trabelsi et al., 2017, adapted the different elements of a CNN to their complex version.

We propose here to use a complex HCQM as input of a complex layer convolutional network in order to take into account the inter-relationships between the acoustic bands.

5.2.2 *Complex HCQM.*

In order to obtain a complex representation of the HCQM, we modify one of its calculation steps presented in Section 4.3. When the HCQT is computed on the onset strength function of the STFT summed in acoustic frequency bands we keep the complex-values (in addition to the modulus). We therefore keep its real and imaginary parts instead of keeping only its absolute value. To be used as input of a convolutional neural network, the two parts are then superimposed on top of each other (Trabelsi et al., 2017), resulting in Cplx-τ'-HCQM of a size $(\Phi \times b \times 2h)$.

5.2.3 *Complex Convolution*

The cplx-HCQM input to the layers is denoted by $H = H_{\mathfrak{R}} + iH_{\mathfrak{I}}$ (with $H_{\mathfrak{R}}$ and $H_{\mathfrak{I}}$ its real and imaginary parts, respectively). The complex kernel matrix of the layer (which is the trainable parameter) is denoted by $K = K_{\mathfrak{R}} + iK_{\mathfrak{I}}$ (with $K_{\mathfrak{R}}$ and $K_{\mathfrak{I}}$ its real and imaginary parts, respectively). The complex convolution is then expressed as:

$$K * H = (K_{\mathfrak{R}} * H_{\mathfrak{R}} - K_{\mathfrak{I}} * H_{\mathfrak{I}}) + i(K_{\mathfrak{I}} * H_{\mathfrak{R}} + K_{\mathfrak{R}} * H_{\mathfrak{I}}) \tag{5–15}$$

or expressed in matrix form as:

$$\begin{bmatrix} \Re(K * H) \\ \Im(K * H) \end{bmatrix} = \begin{bmatrix} K_{\Re} & -K_{\Im} \\ K_{\Im} & K_{\Re} \end{bmatrix} * \begin{bmatrix} H_{\Re} \\ H_{\Im} \end{bmatrix} \tag{5–16}$$

The output of each complex convolution layer is itself complex and is then used as input to the next complex convolution layer. All convolution layers of the original Deep Rhythm network are therefore replaced by complex convolution layers. Also, each complex convolution layers is followed by a complex batch normalization (as described in (Trabelsi et al., 2017)). After the last complex convolution, the resulting feature maps are flattened, hence by concatenating the real and imaginary outputs.

We illustrate this in Figure 5.2 on which we only detail the complex convolution for the first convolution layer (the one applied to the input complex HCQM H). On this figure, H_{Re} and H_{Im} correspond respectively to $H_{\Re}$ and $H_{\Im}$ in Equation 5–15 and 5–16 (this is also holds true for K). Instead of Batch-Normalization (BN), which is applied only to real-values, we use complex-BN preceding each complex-convolution layer in order to ensure equal variance in both real and imaginary components. The whole complex-BN computation process is detailed in (Trabelsi et al., 2017).

On this figure, the number of feature maps indicated under the layers are doubled compared to the original network since the number of complex-kernel are considered. For the implementation of the 2D complex convolution layers, we rely on the package *complexnn*[1] provided in (Trabelsi et al., 2017). We denote this method as Complex Deep Rhythm (Cplx-DR).

5.2.4 *Evaluation*

We use the same evaluation protocol as for the DR method.

For the global tempo estimation task, we perform a large scale analysis by training the network with three combined datasets (LMD, tMTG and EBR) and testing it with seven datasets and the combined one (ACM, BR, tGS, GTzan, Hains., ISMIR04, SMC and Combined).

For the rhythm-oriented genre classification task, we perform again a 10-fold cross validation on the genre-annotated datasets (BR, gEBR and Gr; gMTG, lBP and sBP).

1 https://github.com/ChihebTrabelsi/deep_complex_networks

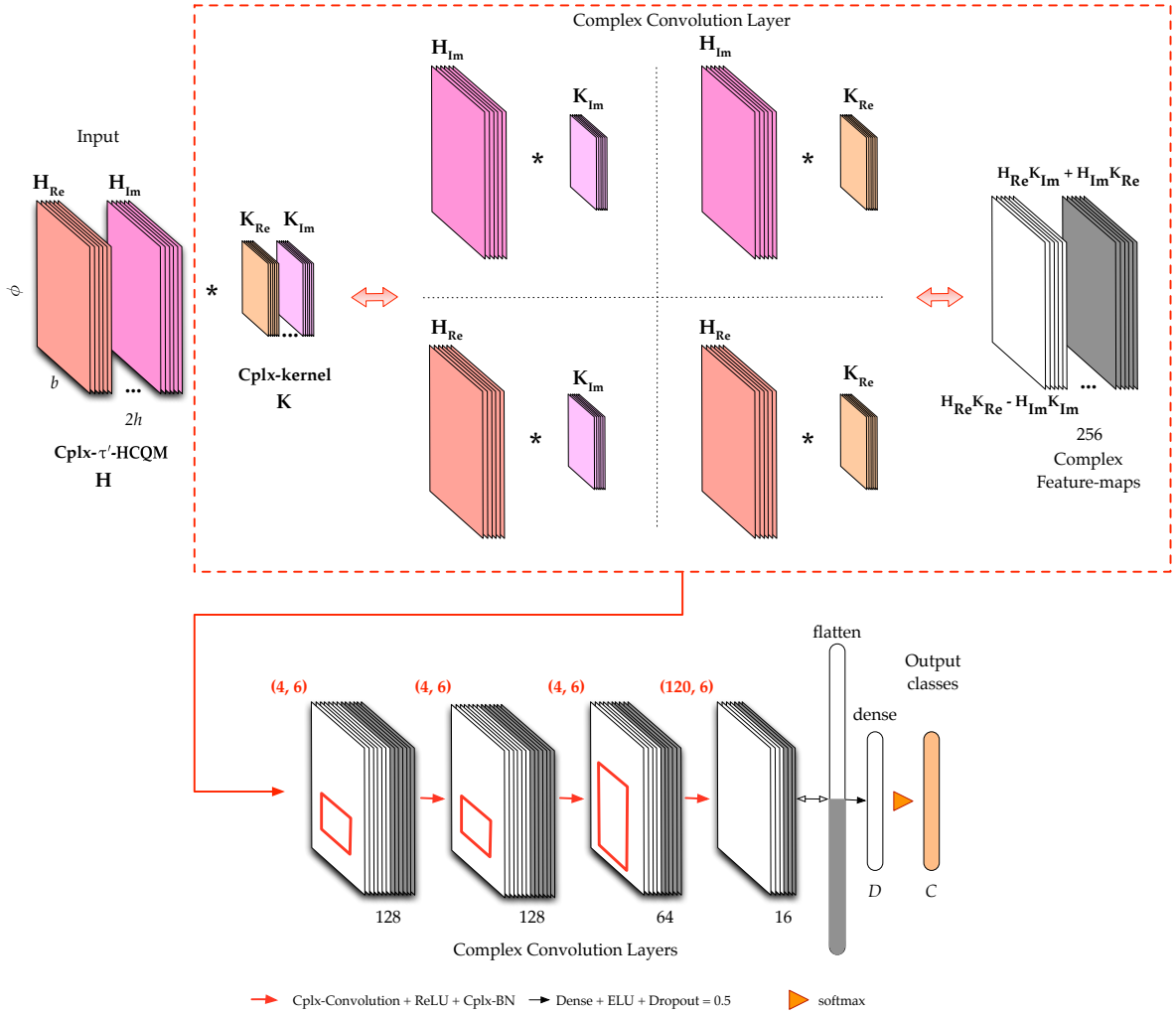

Figure 5.2: Cplx-convolution applied to Cplx-τ′-HCQM as input and Cplx-DR model Architecture. D is the number of units in the dense layer, C is the number of units in the output layer (i.e. the classes logits).

5.2.4.1 *Tempo estimation*

We evaluate the method in terms of Accuracy (*Acc*), Accuracy1 (*Acc1*) and Accuracy2 (*Acc2*) (like in Section 4.6.1). First, we analyze the results obtained with the Cplx-DR method. We also compare the Oracle Frame Prediction (described in Section 4.5.1) results to the Oracle-DR ones, we denote it by Oracle-Cplx-DR. In a second step, we further investigate the results of the method including the ground-truth smoothing described in Section 4.6.1.5. We refer to them as Smooth-Cplx-DR for the fix smoothing and Ada-smooth-Cplx-DR for the adaptive smoothing. We compare these results to the *sch2* state-of-the-art method from (Schreiber and Müller, 2018b) as well.

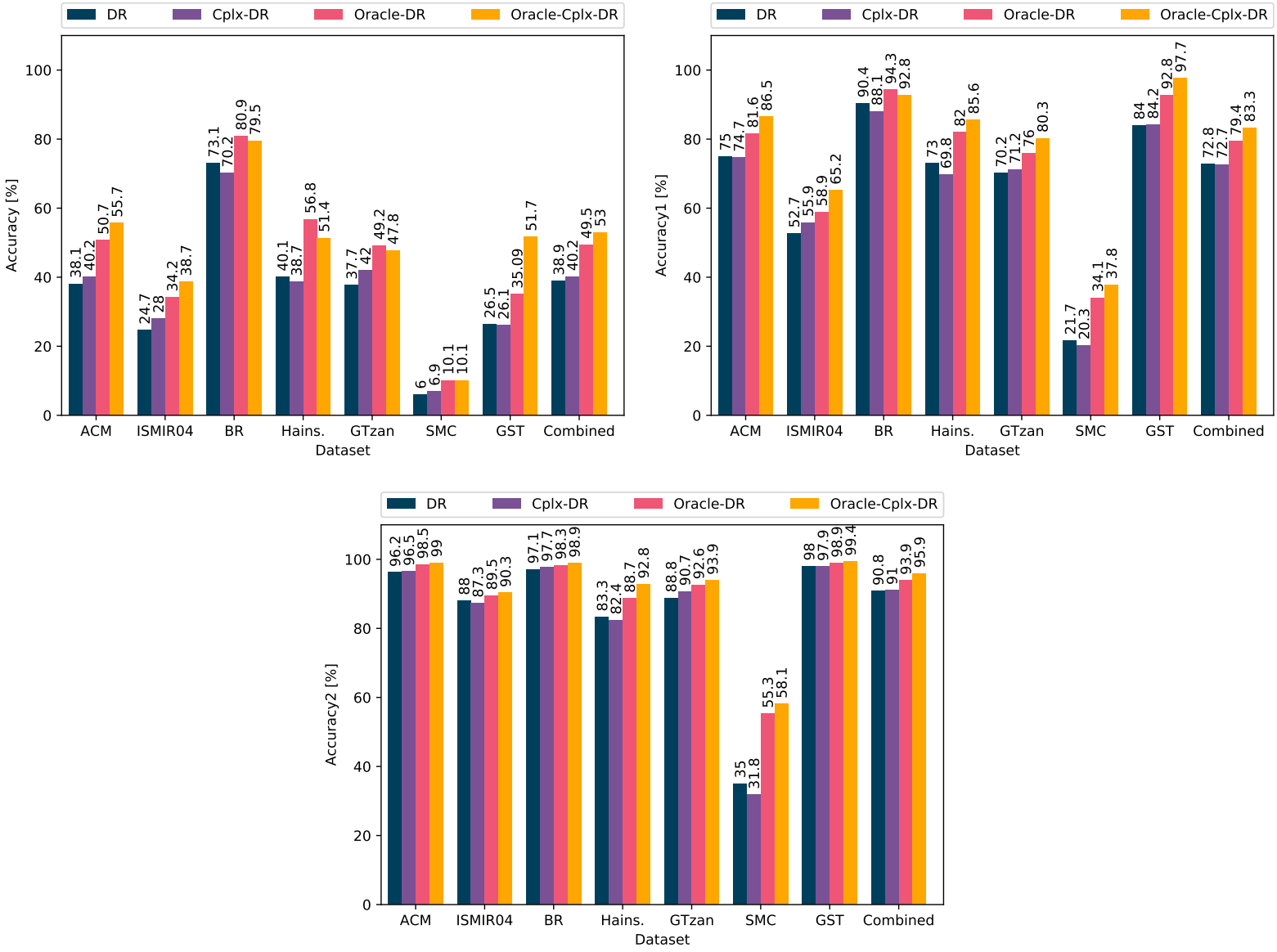

Figure 5.3: Results of Cplx-DR and Oracle-Cplx-DR methods compared to the DR and Oracle-DR methods. [Top left] in terms of *Acc*, [Top Right] in terms of *Acc1*, [Bottom] in terms of *Acc2*.

ORACLE FRAME PREDICTION. The first remark on can made is that the results of Oracle-Cplx-DR are better than those of Oracle-DR in terms of *Acc1* and *Acc2* for all independently evaluated datasets as well as for the combined one. This clearly demonstrates that the complex version allows a significant improvement in tempo estimation. It is necessary to mention here that we have verified in practice that these results were not simply due to the twice as large size of the network convolutional layers. For this purpose, we have trained a non-complex network

from the real-valued HCQM with the same kernel sizes as the complex network. If we look more closely at the results in terms of *Acc* for these same methods, we can also see that on the one hand the results are better for 3/7 datasets and on the other hand that the performance gain for the tGS dataset is drastic (+16.6%). Hence, the use of Cplx-DR definitely allows a better estimation of global tempo when applied to EDM music.

Another observation can be made at this stage by comparing the results obtained in terms of *Acc2*: we can see that DR and Cplx-DR, although slightly below, are almost at the same level as their Oracle version (except for SMC). This allows us to state that for a given track, the average of its predictions from its different τ'-HCQM (respectively Cplx-τ'-HCQM) reflects the presence of octave error in the estimated candidate when using the basic methods.

GROUND-TRUTH SMOOTHING. We now compare the results obtained with the methods using ground-truth smoothing and with the state-of-the-art method *sch2* (Schreiber and Müller, 2018b). From the previous DR method, we keep only the adaptive smoothing method Ada-Smooth-DR for comparison. Smooth-Cplx-DR denotes the fixed smoothing applied to Cplx-DR while Ada-smooth-DR refers to the adaptive one. Compared to *sch2*, the results for the complex methods are similar to the previous analysis provided in Section 4.6.1.5. Indeed, in terms of *Acc1*, the results are better for the Combined dataset while compared to the Ada-smooth-DR, the results are slightly better for the Smooth-Cplx-DR (75%). From a global point of view, we also can see that the two complex methods approximately perform equally.

5.2.4.2 *Rhythm-oriented genre classification*

We provide the results in term of average-mean-recall for the DR and the Cplx-DR methods in Table 5–3.

By exploiting the results, we find that the complex version of the DR does not allow a better genre classification for the BR, gEBR, gMTG and Gr datasets. This can potentially be explained by the fact that including the phase in the learning of the model does not lead to an optimal generalization for the genres represented in these datasets. However, for the gender-balanced EDM datasets, we observe an improvement in the average-mean-recall results of +1.4% for lBP and +1.2% for sBP.

Looking at the confusion matrix Figure 5.5 of the sBP genre classification, one can see a gain in some classes compared to the one obtained with the DR method. Among those, *dance* (+6%), *electronica-downtempo* (+7%), *hip-hop-r-and-b* (+2%), *house* (+2%) present improvements that demonstrate that the inter-relationships between frequency bands are an important factor in the classification of EDM genres. There

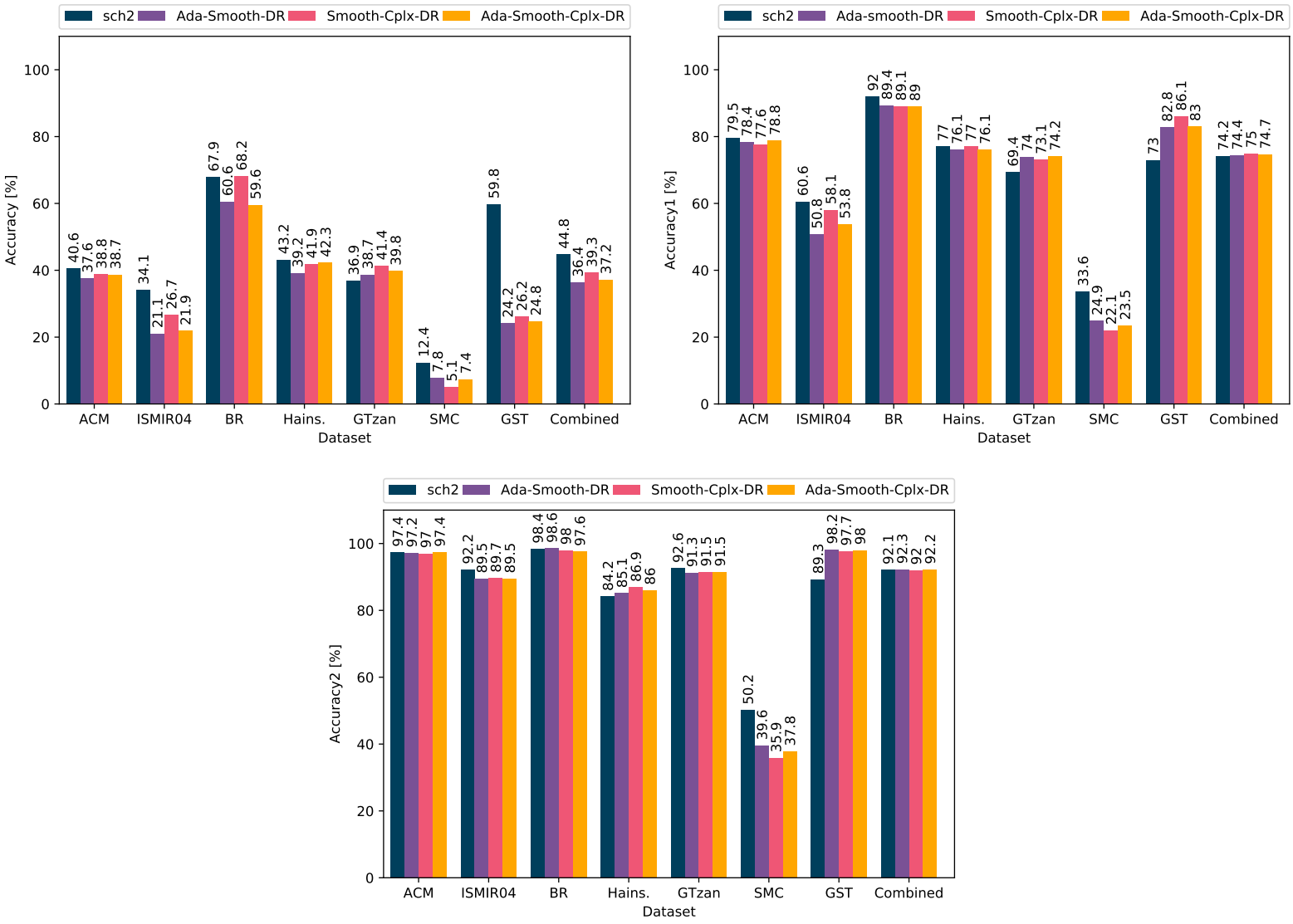

Figure 5.4: Results of Smooth-Cplx-DR and Ada-Smooth-Cplx-DR methods compared to the Ada-Smooth-DR and *sch2* methods. [Top left] in terms of *Acc*, [Top Right] in terms of *Acc1*, [Bottom] in terms of *Acc2*.

again, *dance* is classified in similar genres as *funk-soul-disco* or *house*. Even if there is a slight improvement in genre classification, the confusion matrix is quite similar to the one presented in Section 4.6.2.2. We thus reach the same conclusions on genre classification based on rhythmic characteristics.

Table 5–3: Results of rhythm-pattern recognition/rhythm-oriented genre classification using Cplx-DR method compared to DR method in term of average-mean-recall $\hat{R}$.

Dataset	DR	Cplx-DR
Ballroom (BR)	**93.0%**	86.5%
Extended Ballroom (gEBR)	**95.4%**	92.1%
Greek Dance (GR)	**68.9%**	40.0%
genre MTG (gMTG)	**37.6%**	36.4%
Beatport large (lBP)	40.7%	**42.1%**
Beatport small (sBP)	52.8%	**54.0%**

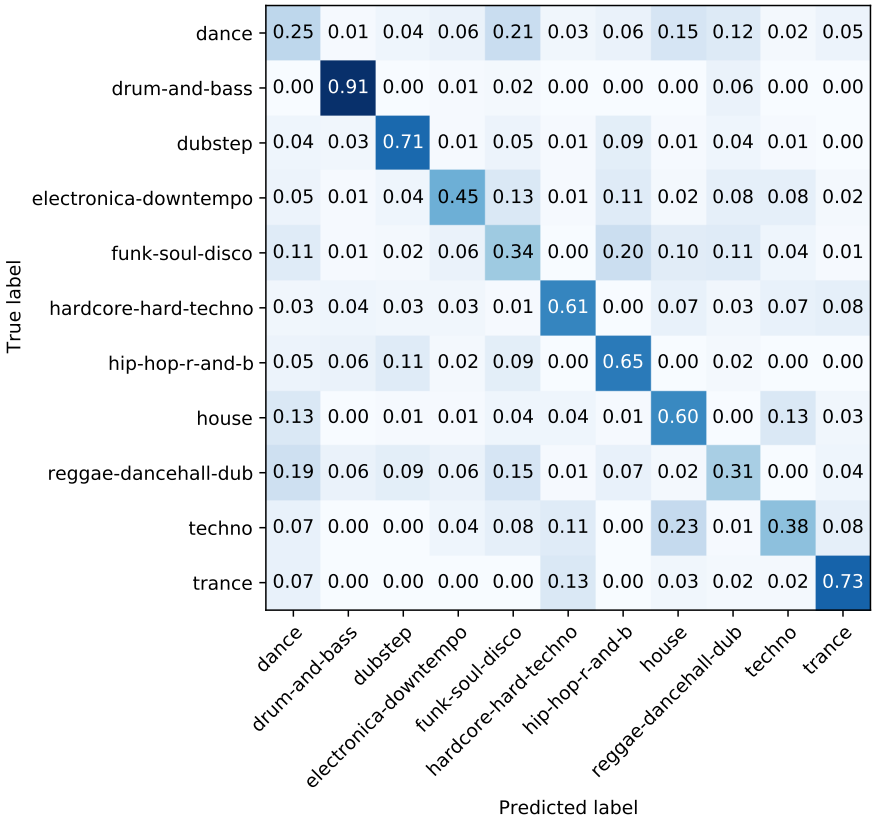

Figure 5.5: Confusion matrix of the evaluation of the sBP dataset using Cplx-DR method.

5.3 MULTITASK LEARNING

5.3.1 *Why multitask learning?*

The goal of MultiTask Learning (MTL) is to share information between two related tasks in order to enable a model to generalize better on both of these tasks. The origin of MTL is biological. For example, babies learn to recognize faces of their parents before apply this knowledge to recognize other faces. From a machine learning point of view, MTL can be seen as an inductive transfer that can help learning a model by introducing an inductive bias. In this case, the inductive bias is provided by the auxiliary task which lead the model to favor hypotheses that are beneficial to all several tasks at once.

Taking the example of EDM and the genres (whose rhythmic characteristics we have described in Section 3.3), we can see that the tempo ranges are characteristic of certain genres. This observation can be extended more generally, in the case of ballroom music for instance.

With the DR method, we showed that the same network architecture, but with two different trainings and hence set of parameters, can be used to achieve two different tasks: tempo estimation and rhythm-oriented genre classification. We want here to exploit this multitasking aspect through the implementation of a single network which jointly estimates the tempo and the rhythm pattern class.

Two main strategies stand out when it comes to MTL in the field of deep learning. These reside in the type of parameter sharing carried out between the hidden layers of the network.

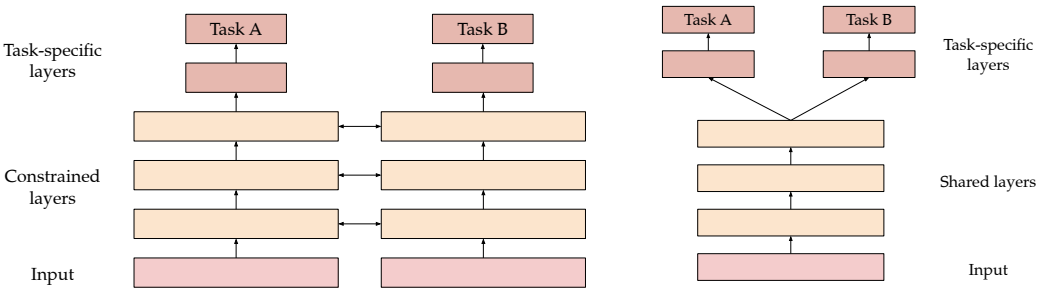

Figure 5.6: MTL strategies. [Left] Soft parameter sharing. [Right] Hard parameter sharing.

The two strategies are shown in Figure 5.6. For the soft-sharing [Left], each task has its own network. In order to learn similar parameters, the distance between them is regularized for each layer of the networks. This method have been proposed in (Yang and Hospedales, 2016) using the trace norm for regularization

while in (Duong et al., 2015) the l_2-norm is used for a language processing task. For the hard-sharing [Right], the hidden layers are shared between all tasks while conserving specific layers for each task. This is the most common strategy. Hard parameter sharing has been proven to limit the risk of overfitting (Baxter, 1997).

Other benefits of using MTL in deep learning are to be considered. A model that jointly learn two tasks is able to learn a more general representation: it has an implicit data augmentation effect. It can be difficult for a model to learn the differences between relevant and irrelevant features, MTL acts as an attention focusing on the various features since it combines the relevance of features from two linked tasks.

MTL methods have already shown their efficiency in various domains like computer vision (Girshick, 2015), natural language processing (Collobert and Weston, 2008) or speech recognition (Deng, Hinton, and Kingsbury, 2013). In MIR field, they have been used for the estimation of the fundamental frequency (Bittner, McFee, and Bello, 2018). In the case of rhythm description, the work of Böck, Davies, and Knees (2019) have showed good results by learning tempo estimation in parallel with the beat tracking.

5.3.2 *Multitask Deep Rhythm*

ARCHITECTURE. For our MTL network we choose a hard parameter sharing strategy. We illustrate the architecture of the network in Figure 5.7. The convolutional part is the same as the original DR network. The extension starts at the flatten layer that follows the last convolutional layer. The associated vector then acts as input for two independent branches, each with two fully-connected layers ending with a softmax activation function. One branch is dedicated to genre classification, the other to tempo estimation. We choose $D = 64$ based on parameter validation experiments. The output of the first (resp. of the second) branch has the same size as the number of genre to be detected (resp. as the number of tempo classes).

It is important to note that using the MTL network architecture to deal with both tasks simultaneously allows to halve the number of trainable parameters. Indeed, the original DR network is trained to perform the two tasks separately and therefore requires two independent training.

LOSSES. To train the system, we simultaneously minimize two categorical cross-entropy losses: one for the rhythm pattern classes $\mathcal{L}_{genre}$ and one for the tempo classes $\mathcal{L}_{tempo}$. Both are applied to the output of the sub-networks ended by a

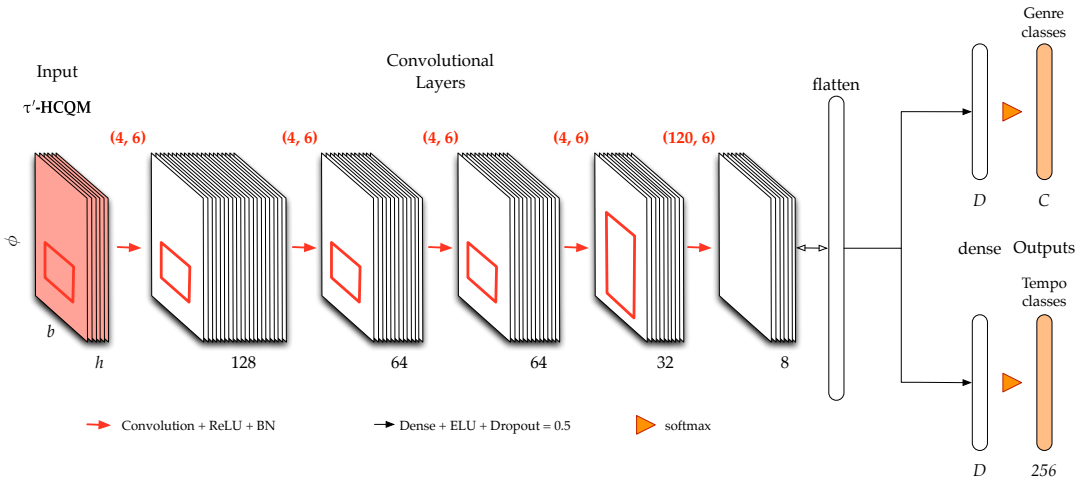

Figure 5.7: MTL model Architecture with τ'-HCQM as input (of size $(\phi \times b \times h)$), D is the number of units in the dense layer, C is the number of genre classes in an output layer and 256 tempo classes in the other.

softmax activation function. We choose the use of an additive loss with equal weights between the genre loss and the tempo loss.

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{genre} + \mathcal{L}_{tempo} \\ &= -\sum_{c_g=1}^{C} y_{c_g} \log(\hat{y}_{c_g}) - (1 - y_{c_g}) \log(1 - \hat{y}_{c_g}) \\ &- \sum_{c_t=1}^{256} y_{c_t} \log(\hat{y}_{c_t}) - (1 - y_{c_t}) \log(1 - \hat{y}_{c_t}) \end{aligned} \tag{5–17}$$

with the predicted genre class $\hat{y}_{c_g}$ and the genre ground-truth y_{c_g} (c_t for the tempo classes, respectively). We then minimize $\mathcal{L}$ (both losses $\mathcal{L}_{genre}$ and $\mathcal{L}_{tempo}$ are equally weighted) with the same ADAM optimizer as original DR method.

5.3.3 *Evaluation*

EVALUATION PROTOCOL. Once again we evaluate the results for both estimation tasks. However, unlike our previous methods, learning is joint between the two tasks (i.e. the same model is trained/evaluated to estimate both tempo and genre). The experimental protocol is thus limited to datasets annotated in both genre and tempo: BR, gEBR, gMTG and GTZAN. We do not present the results for Gr since data are not annotated into tempo value and for lBP and sBP since the tempo annotations are considered ambiguous (cf. Section 3.3). Similarly to the DR method, we choose to perform a 10-fold cross validation.

We present the results in two tables. The Table 5–4 is dedicated to genre classification and results are presented in terms of average mean recall. The Table 5–5 concerns tempo estimation and results are presented in terms of Accuracy1 (within a ±4% tolerance window applied to the ground-truth tempo value). In these two tables, the results corresponding to the MTL methods are the results obtained by jointly using the same model. We compare them to the DR and Cplx-DR methods. For the latter two, the results are not obtained jointly but independently (i.e. for tempo estimation we have re-trained the networks on the independently evaluated datasets). It should be noted that for all three types of methods, a fixed smoothing was applied to the ground truths of the tempo value. Finally, we tested the complex version of the MTL network for comparison, denoted as Cplx-MTL.

Table 5–4: Separate and joint estimation results of rhythm-oriented genre classification in term of average-mean-recall $\hat{\mathrm{R}}$. MTL and Cplx-MTL results are obtained jointly with the tempo results of Table 5–5

Dataset	DR	Cplx-DR	MTL	Cplx-MTL
Ballroom (BR)	**93.0%**	86.5%	92.1%	86.1%
Extended Ballroom (gEBR)	**95.4%**	92.1%	94.8%	92.4%
genre MTG (gMTG)	37.6%	36.4%	37.1%	**39.8%**
GTzan tempo (GTzan)	**59.1%**	43.5%	57.1%	44.0%

Table 5–5: Separate and joint estimation results of global tempo estimation in term of Accuracy1. MTL and Cplx-MTL results are obtained jointly with the genre results of Table 5–4

Dataset	DR	Cplx-DR	MTL	Cplx-MTL
Ballroom (BR)	**93.3%**	90.0%	93.2%	91.4%
Extended Ballroom (gEBR)	96.0%	95.7%	**96.4%**	95.6%
MTG tempo (gMTG)	**92.0%**	91.8%	91.2%	**92.0%**
GTzan tempo (GTzan)	**76.3%**	72.4%	74.8%	70.8%

DISCUSSION. The results of the MTL and Cplx-MTL methods presented in the two tables for tempo estimation and genre classification show that joint learning of the both tasks is justified. Indeed, only one model was used to obtain these results despite the fact that they perform slightly below the DR methods for BR and GTzan. For the gEBR the best results are achieved with the MTL method. We also note that for EDM dataset gMTG, the results are better with Cplx-MTL for the genre classification (+2.2%) and equal for the tempo estimation (92%) compared to

those of the DR method. This confirms once again the fact that the genres of EDM are strongly defined by their tempo ranges.

5.4 MULTI-INPUT NETWORK

5.4.1 *Why multi-input network?*

The Deep Rhythm network was designed to represent the rhythm content of an audio track. As shown in (Gouyon et al., 2006) and demonstrated in our previous methods, the tempo range and possible rhythm patterns are strongly correlated to the music genre of the track. The DR network, however, focuses exclusively on the aspects related to rhythm, not on other features like instrumentation or timbre.

Since we want to perform a genre classification, even rhythm-oriented, our method could benefit from representations describing other musical elements. For instance in EDM, we have seen that many genres were also defined, beyond their rhythmic structure, by more instrument-oriented features. In the analysis dedicated to the basic DR method, we can have seen that genres with instrumentation marked by the presence of vocals and acoustic instruments (*disco, hip-hop, dance, house*) were more likely to be confused during classification.

This intuition led us to the creation of a new network in order to introduce information distinct from rhythm when training the network for genre classification. To do so, we keep the convolutional part of the DR dedicated to rhythm with the HCQM as input and we add a convolutional branch dedicated to the representation of timbre and instrumentation with a log-mel magnitude spectrogram as input. We obtain a multi-input, multi-branch network, and to denote this network we use the term multi-input network (and by extension Multi-Input (MI) method).

5.4.2 *Multi-input Network*

We first start by describing the branch dedicated to the analysis of timbre and instrumentation. Our main inspiration is the work of Choi, Fazekas, and Sandler (2016) and Choi et al. (2017) who uses this type of convolutional layers with a mel-spectrogram as input to perform an automatic tagging task from large annotated databases. Pons et al. (2017b) also recommend the use of log-mel magnitude spectrogram[2] as input of a CNN to analyze timbre patterns through the convolutional feature maps of the network.

5.4.2.1 *Mel-spectrogram*

Studies on human frequency perception have shown that we are better at detecting differences in lower frequencies than in higher frequencies. It was then shown by Stevens, Volkmann, and Newman (1937), in the field of psychoacoustics, that a

2 For ease of reading we refer to it as mel-spectrogram

unit of pitch called mel scale was more adapted to represent frequencies for the human auditory system. Thanks to the mel scale, an equal distance in pitch sounds equally distant to the listener.

To compute the mel-spectrogram (represented in Figure 5.8), the STFT of a raw signal is computed, the frequency axis is converted to log scale and the amplitude are converted to decibels to obtain the log-spectrogram. Finally, the frequency are mapped onto the mel scale.

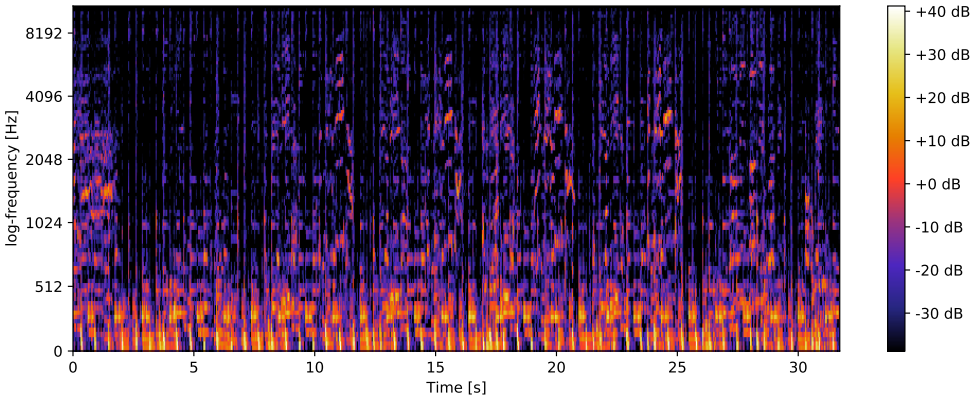

Figure 5.8: Log-mel Magnitude Spectrogram example

Timbre is defined as the character or quality of a musical sound. Also, music timbre is often associated with the identification of the instrument characteristics. It is related to the spectral envelope shape and the time variation of the spectral content (Peeters et al., 2011). Thus, it can be assumed that the mel-spectrogram is an adequate representation of timbre since it is a time/frequency expression well-suited to the human auditory system and so to the music perception.

Therefore, the mel-spectrogram contains timbre patterns that are considered as more pitch invariant than the STFT ones since they are based in a different perceptual scale (Pons et al., 2017b). For all these characteristics and also for the gain in performance it allows, this representation is one of the widespread features used in deep learning for various MIR tasks. It has notably been used for boundary detection (Ullrich, Schlüter, and Grill, 2014; Cohen-Hadria and Peeters, 2017), onset detection (Schlüter and Böck, 2014), latent feature learning (Oord, Dieleman, and Schrauwen, 2013) or tagging (Pons and Serra, 2017; Choi, Fazekas, and Sandler, 2016; Choi et al., 2017; Dieleman and Schrauwen, 2014).

For tagging task, the mel-spectrogram is often used as input representation of a CNN trained with large-scale datasets. We choose to use a common architecture dedicated to the timbre branch of our MI network.

5.4.2.2 *Network Architecture*

The first branch of the network is the one dedicated to rhythm with the HCQM as input followed by the DR convolutional layers. The second branch is the one dedicated to timbre characteristics followed by a network commonly-used for audio tagging with the mel-spectrogram as input. This latter branch is inspired from the well-known VGG network (Simonyan and Zisserman, 2015). Its architecture is adapted to large-scale analysis in the sense that its efficiency has been shown on large annotated databases. It is composed of a series of convolution layers associated with max-pooling layers.

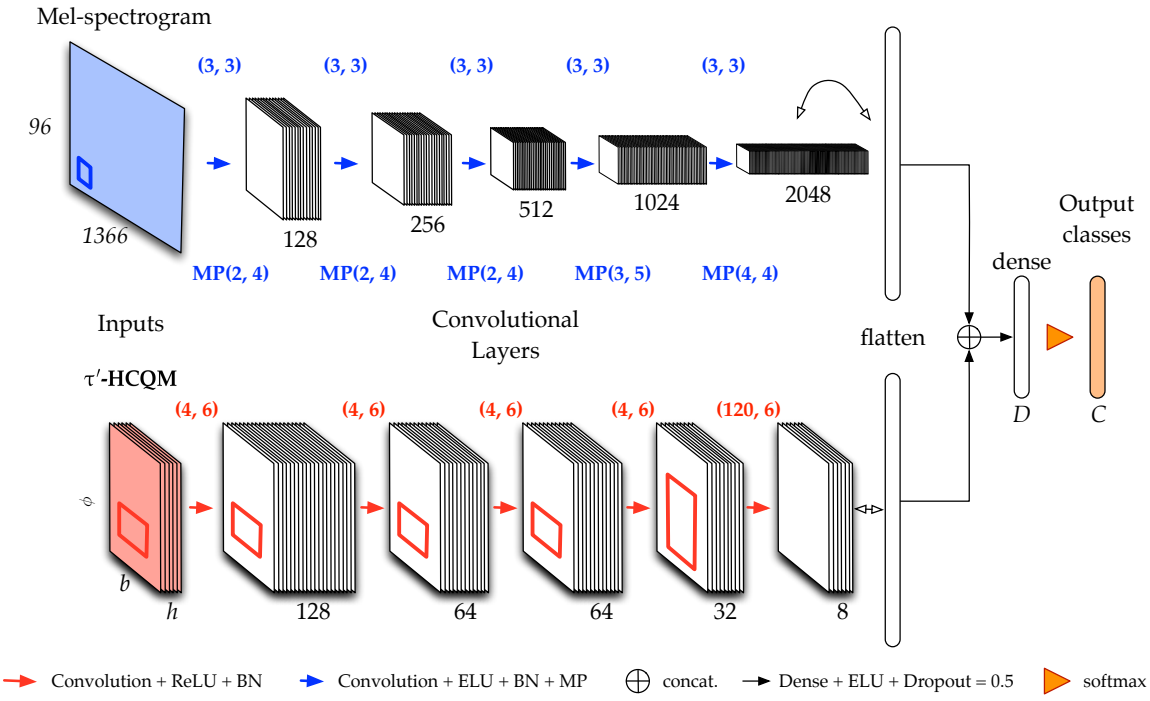

Figure 5.9: MI model architecture. [Top] Branch dedicated to "timbre", VGG-like convolutional layers with a mel-spectrogram as input. [Bottom] Branch dedicated to rhythm, DR convolutional layers with an τ'-HCQM as input.

The complete architecture of the MI network is described in figure Figure 5.9. We take the same network parameters described in (Choi, Fazekas, and Sandler, 2016) and in (Choi et al., 2017). This network uses mel-spectrograms as input followed by five convolutional layers of (3×3) kernels[3] each connected to a max pooling layer $(2 \times 4), (2 \times 4), (2 \times 4), (3 \times 5), (4 \times 4)$ in order to reduce the size without losing

3 In (Pons et al., 2017b), it has been shown that the use of domain-knowledge inspired kernel size (e.g. by taking the whole frequency axis) leads to better performance in the case of large-scale training. However, because the difference in results is not significant on smaller scale analyses, so we choose to use square filters.

information during training. In the original network, the last layer then predicts the tags. We skip it here. The output of the timbre branch does not need to be flatten since using max pooling already shapes the last layer as a (1×2048) vector. The flatten layer of the DR branch is concatenated with the last layer of the timbre branch and used as input of a dense layer of size 256. The output layer of the MI network is, as for the previous methods, the softmax activation function of C classes (C as the genre classes in the case of genre classification task or 256 tempo classes in the case of global tempo estimation). Again, the network is trained by minimizing a categorical cross-entropy.

5.4.3 *Evaluation*

EVALUATION PROTOCOL. In (Choi, Fazekas, and Sandler, 2016), the model is trained with a large amount of data from the Million Song dataset (Bertin-Mahieux et al., 2011). We have not this quantity of tempo labeled data in our datasets (we have seen that we cannot merge them because the genres covered are too disparate from one dataset to another). In order to evaluate our results, we compare all the previous methods:

- DR, the original one presented in Chapter 4;
- Cplx-DR, presented in Section 5.2;
- MTL and Cplx-MTL presented in Section 5.3;
- MI and its complex version Cplx-MI. For Cplx-MI, only the DR branch is converted to its complex version.

To these, we add the MI-MTL method (and its complex version Cplx-MI-MTL). This architecture combines the two input branches of the MI and their concatenation followed by an MTL output with on one side a dense layer followed by the softmax dedicated to genre classification and on the other side the same layers dedicated to tempo classes. The loss minimized is the additive categorical cross-entropy loss presented in Equation 5–17

We also want to compare the efficiency of our method for genre classification to a baseline from the state of the art. As in the DR method we choose to train/test all networks from different methods with a 10-folds cross-validation on our different datasets including the independent tagging network described in (Choi, Fazekas, and Sandler, 2016), we denote it by *choi* in our results analysis.

As for the MTL evaluation in Section 5.3.3, we display the results in the form of two tables. Table 5–6 presents the results for genre classification in terms of average mean recall. Table 5–7 presents the results for global tempo estimation

in terms of Accuracy1. In the two tables, the results of the *-MTL methods are obtained jointly using the same model for both tasks. The results of the others methods (DR, Cplx-DR, MI, Cplx-MI) are obtained using a 10-fold cross validation on tempo estimation task.

Table 5–6: Comparative and joint estimation results of rhythm-oriented genre classification in term of average-mean-recall $\hat{R}$ for all methods.

Dataset	choi	DR	Cplx-DR	MTL	Cplx-MTL	MI	Cplx-MI	MI-MTL	Cplx-MI-MTL
BR	60.1%	93.0%	86.5%	92.1%	86.1%	**94.2%**	92.3%	93.0%	91.9%
gEBR	72.1%	95.2%	92.1%	94.8%	92.4%	**96.5%**	93.9%	96.2%	94.6%
Gr	38.1%	68.9%	40.0%	-	-	**69.4%**	47.2%	-	-
gMTG	21.7%	37.6%	36.4%	37.1%	39.8%	37.3%	**40.6%**	39.6%	40.3%
GTzan	74.2%	59.1%	43.5%	57.1%	44.0%	**74.3%**	74.1%	67.2%	66.0%
lBP	21.3%	40.7%	**42.1%**	-	-	31.3%	32.0%	-	-
sBP	46.4%	52.8%	54.0%	-	-	54.8%	**55.8%**	-	-

Table 5–7: Comparative and joint estimation results of global tempo estimation in term of Accuracy1 for all methods.

Dataset	choi	DR	Cplx-DR	MTL	Cplx-MTL	MI	Cplx-MI	MI-MTL	Cplx-MI-MTL
BR	-	**93.3%**	90.0%	93.2%	91.4%	91.3%	92.7%	92.2%	92.4%
gEBR	-	96.0%	95.7%	96.4%	95.6%	**96.1%**	94.6%	96.0%	95.7%
gMTG	-	**92.0%**	91.8%	91.2%	**92.0%**	91.6%	90.1%	91.3%	91.6%
GTzan	-	**76.3%**	72.4%	74.8%	70.8%	73.3%	69.5%	71.5%	68.5%

DISCUSSION. First of all, we can observe that the results obtained with our DR method are much better than the *choi* baseline for all datasets except for GTzan. We can thus conclude that a small-scale rhythm-oriented classification method is much more efficient than a method more generally applied to large-scale dataset tagging. The results on GTzan can be explained by the fact that the *choi* method has shown good results in previous work on dataset balanced in various popular music genres. As learning takes place on this various data, it allows a VGG-like model to generalize better.

We can observe that the MI method is the most efficient for genre classification of BR, gEBR, Gr, GTzan. Moreover, for the MTG and the sBP dataset it is the Cplx-MI version that performs best. Although learning is joint, the MI-MTL methods fail to match the previous ones in terms of statistical results. We can still validate the use of a MI model for genre classification. The timbre branch brings a significant added value.

Once again, we observe the confusion matrix of the genre classification this time using Cplx-MI method in Figure 6.10. The addition of the MI branch dedicated to

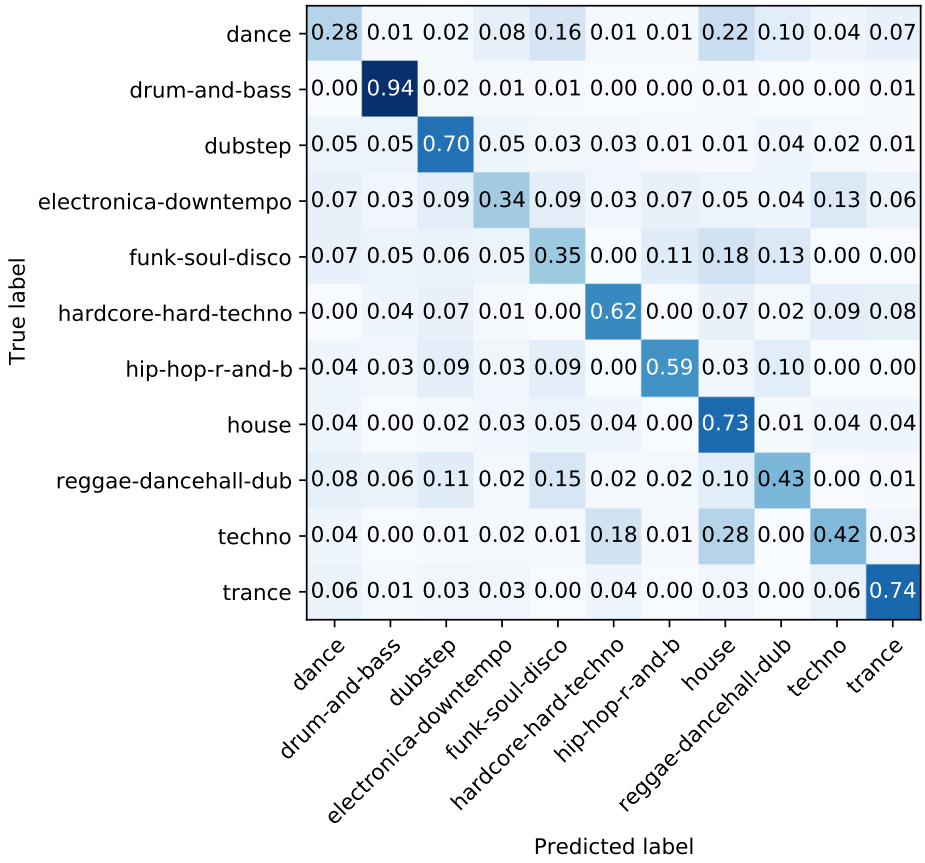

Figure 5.10: Confusion matrix of the evaluation of the sBP dataset using Cplx-MI method.

timbre allows to improve the classification performances compared to the Cplx-DR method for *dance* (+3%), *house* (+13%) and *reggae-dancehall-dub* (+12%). All these genres are specifically defined by the presence of vocals or acoustic samples (brass, guitar riffs, etc.) in their composition. The addition of a timbre branch therefore justifies this improvement.

Regarding the use of MI for tempo estimation the results remain very close to the DR method and are even slightly higher for gEBR. Using a 10-fold cross validation protocol, DR still remains the most performant method for the majority of the datasets.

5.5 CONCLUSION

In this chapter, we presented three main extensions of the Deep Rhythm method with the aim of exploiting the specific characteristics of this method to perform tempo estimation and rhythm-oriented genre classification.

First, we wanted to take into account inter-band acoustic relationships in order to improve our estimation. To do this, we proposed to integrate the temporal relations between the bands through the combined learning of the module and the phase of our HCQM representation by a neural network. We kept the complex-values when calculating the Cplx-HCQM used as input of a complex network using complex convolution layers. The effectiveness of this complex version of the DR, the Cplx-DR, is shown through its evaluation for the tempo estimation task. Indeed this method allows a clear improvement of the results when analyzing the results obtained via the Oracle Frame Prediction in terms of *Acc1* and *Acc2*. Regarding the genre classification, we have noticed that except for an EDM dataset, the results are inferior to those obtained with the DR method.

Second, in order to better take into account the interdependence between tempo and genre we proposed a multitask network where the two tasks of tempo and genre estimation are jointly solved. For this, we proposed a hard parameter sharing network architecture with shared hidden layers and with two independent output layers dedicated to each task. This network was trained to minimize the additive categorical cross-entropy losses of the two outputs. We showed that MTL led to an improvement for both tasks on the only evaluated EDM dataset gMTG. In addition, even if the results are not better than the ones of the previous methods, it is important to emphasize that they are obtained using a single network trained to perform both tasks.

Third, we wanted to take into account an other descriptor in addition to the rhythm one represented by the HCQM and the DR network. We put forward a multi-branch/multi-input network where VGG-like convolutional layers with mel-spectrogram input are added to represent timbre information. We showed that this MI architecture allowed a much better genre classification for almost all datasets evaluated especially for EDM.

To conclude, we showed that following these three intuitions, we were able to extend the DR method effectively and particularly in the context of EDM. We published these three methods at the MuMe conference 2020 (Foroughmand and Peeters, 2020). Note that the results for the methods estimating tempo differ from the article for those prior to the ground-truth smoothing method.

For our last chapter, we wish to exploit the benefits of a rhythmic representation from another angle. To do this, we want to highlight the quality of the HCQM representation to emphasize the rhythmic similarities between the tracks. We then describe the implementation of a metric learning paradigm in order to obtain a new data space that we evaluate on our two tasks.

6

METRIC LEARNING DEEP RHYTHM

6.1 INTRODUCTION

As we have seen in the case of music genre classification, many features have been proposed to highlight the similarity of two music tracks (rhythm, timbre, etc). Classification algorithms estimate a similarity rating between two music tracks by calculating the distance between the features extracted from them. In section Section 2.4, we have presented some of the machine learning methods used to perform such classification (k-NN, SVM, etc). We also discussed the use of deep learning in the previous chapters and more specifically the use of CNN to perform classification on large-scale dataset for tempo estimation and on independent cross-fold validation datasets for rhythm-oriented genre classification. In general, algorithms for genre classification are supervised, all items of a dataset are labeled. In the opposite, some algorithms like k-means, are unsupervised. In these, the similarity between items is defined by the metric used (Lloyd, 1982).

Such metric can be effective to estimate the differences between features of a given type but not for other types. The choice of an adapted metric is one of the specific problems when performing unsupervised learning. The metric learning paradigm provides a solution.

In the upcoming content we use the DR method (and by extension the HCQM representation) is suitable for the metric learning of rhythm similarity. For this, we develop a method based on a descriptor dedicated to rhythm allowing to create a space of rhythm similarity. We first introduce the metric learning principle in Section 6.2. Then in Section 6.3, we focus on one of the loss function used for metric learning, namely the triplet loss. In Section 6.4, we present the use of this paradigm to the DR. Finally, we discuss in Section 6.5 the evaluation protocol we have relied on and the results we have obtained.

6.2 METRIC LEARNING PRINCIPLES

The terminology of metric learning has been proposed by Lowe (1995) that defines it as a solution for bearing at the low generalization of algorithm such k-NN. The method aims to learn the parameterization of the metric during the train-

ing of a model on a dataset. The aim is also to improve the probability of a good classification. Xing et al. (2003) demonstrate that metrics can be learned from a training dataset and can be expressed as a Mahalanobis distance. Let $X_{train} = [x_1, x_2, ..., x_N] \in \mathbb{R}^{D \times N}$ be the training samples of a given dataset, D the dimension of the Euclidean space, N the total number of training samples, x_i the i training sample, the distance between two training samples x_i and x_j is expressed:

$$d_M(x_i, x_j) = \sqrt{(x_i - x_j)^T M (x_i - x_j)} \tag{6–18}$$

where M denotes a positive semi-definite matrix and can be decomposed as follow:

$$M = W^T W. \tag{6–19}$$

Should M be the identity matrix, the metric d_M is the Euclidean distance. It comes:

$$\begin{aligned} d_M(x_i, x_j) &= \sqrt{(x_i - x_j)^T M (x_i - x_j)} \\ &= \sqrt{(x_i - x_j)^T W^T W (x_i - x_j)} \\ &= \|Wx_i - Wx_j\|_2 \\ &= d_M(x_i, x_j) = d_M(Wx_i, Wx_j) \end{aligned} \tag{6–20}$$

Learning a metric through M is equivalent to learn a linear transformation and apply the Euclidean distance to the transformed data. The transformed data are then called the *embedding* of the original data (respectively the transformed space is called the *embedding space*).

It has been shown in the literature that the automatic learning of metrics is equivalent to learning the parametrization of M. In (Xing et al., 2003), the purpose of the training criteria is to minimize the distances between similar pairs of samples and maximize the ones between dissimilar pairs. Goldberger et al. (2005) introduced the neighborhood component analysis with M learned through a stochastic method for classification purposes. The probability of a point being correctly classified is expressed as a function of its distance from all other points in its class. M is then updated to maximize this probability for all points.

According to Chopra, Hadsell, and LeCun (2005), given a measure c between a pair of samples and a transformation function f, we can write $c(f(x_i), f(x_j))$. For a sample x_i, $f(x_i)$ denotes its embedding. Here, a constraint is applied in order to avoid that all the embedding points representations end up in the same spot. c

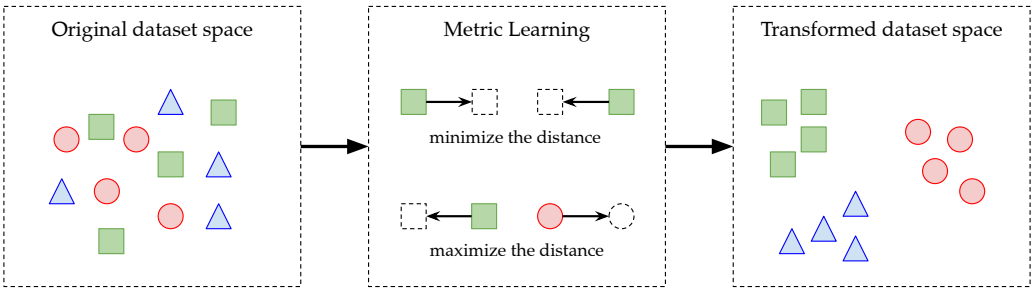

Figure 6.1: Metric learning principle.

is thus the measure of similarity between two training samples. Chopra uses the term *contrastive* to refer to the fact that we are not only minimizing c for similar sample pairs but also maximizing c for dissimilar sample pairs.

To summarize, the original training data are projected from their original space to a target (embedding) space. A function f is trained in order to perform the best possible transformation i.e. to allow the best classification. A similarity between two samples in the original space is equivalent to an estimation of the distance between their embedding in the target space. In a schematic way, this will allow to "bring closer" similar samples (into clusters) and to "move away" dissimilar samples in the embedding space as described in Figure 6.1. For example, in the case of a classification in musical genres we expect that the clusters formed in the embedding space group together all tracks belonging to the same genres while being distant from the other clusters.

In order to apply the metric learning (and by extension the corresponding transformation of the original data) the loss function must be adapted to the task at hand.

6.3 LOSSES

6.3.1 *Evolution of metric learning losses*

In the metric learning context, minimizing a loss function means minimizing the distance between two similar samples; on the contrary, maximizing the loss function means maximizing the distance between two dissimilar samples. Hadsell, Chopra, and LeCun (2006) extended *the contrastive loss* introduced in (Chopra,

Hadsell, and LeCun, 2005), as the sum of partial losses $\mathcal{L}_P(x_i, x_j)$ computed for each possible sample pairs (x_i, x_j) in the input space:

$$\begin{aligned}\mathcal{L}(P) &= \sum_{x_i,x_j} \mathcal{L}_P(x_i, x_j) \\ &= \sum_{x_i,x_j} \delta_{x_i,x_j} \frac{1}{2} d_P^2(x_i, x_j) \\ &+ (1 - \delta_{x_i,x_j}) \frac{1}{2} [max(0, \alpha - d_P(x_i, x_j))]\end{aligned} \tag{6–21}$$

where $d_P is the distance \in \mathbb{R}^+$, $\delta_{x_i,x_j} = 1$ if x_i and x_j are similar samples and zero if they are dissimilar and α is a margin parameter.

Calculating the distances between all pairs is laborious and the estimation of similarity or dissimilarity tends to be limited. Weinberger, Blitzer, and Saul (2006) propose to overcome this problem by considering a limited number k of positive matches for a given sample. The contrastive loss equation is modified as follows and is called the *large margin nearest neighbours* loss:

$$\begin{aligned}\mathcal{L}(P) &= \sum_{x_i,x_j} \mathcal{L}_P(x_i, x_j) \\ &= \sum_{x_i,x_j} \eta_{x_i,x_j} d_P^2(x_i, x_j) \\ &+ \sum_{x_i,x_j,x_z} \eta_{x_i,x_j} (1 - \delta_{x_i,x_j}) [max(0, \alpha + d_P(x_i, x_j)) - d_P(x_i, x_z))]\end{aligned} \tag{6–22}$$

where x_z is a negative sample, $\delta_{x_i,x_j} = 1$ if x_i and x_j are similar (i.e. a positive pair) and zero otherwise and $\eta_{x_i,x_j} = 1$ if x_j is in the k first similar samples of x_i and zero otherwise. According to the first term, each samples will be pulled near its k positive elements. The second one implies that a samples will be closer to its k positive elements than it is to any of its negatives ones ((x_i, x_z) is a negative pair) by a margin α. The complementary use of a dissimilar sample x_z to similar samples x_i and x_j is at the origin of the notion of *triplet*.

The two losses are schematized in Figure 6.2.

In the field of image analysis and more precisely for a classification task, Simo-Serra et al. (2015) observe that once the network has strongly learned from data, the triplets tend to be ineffective for the remaining of the training. In other words, for similar positive pairs, the distance is already close to zero and for negative pairs beyond α. We talk about *easy* triplets in opposition to *hard* triplets. The latter are representative of a high distance between positive pairs and, accordingly, a small distance between negative pairs.

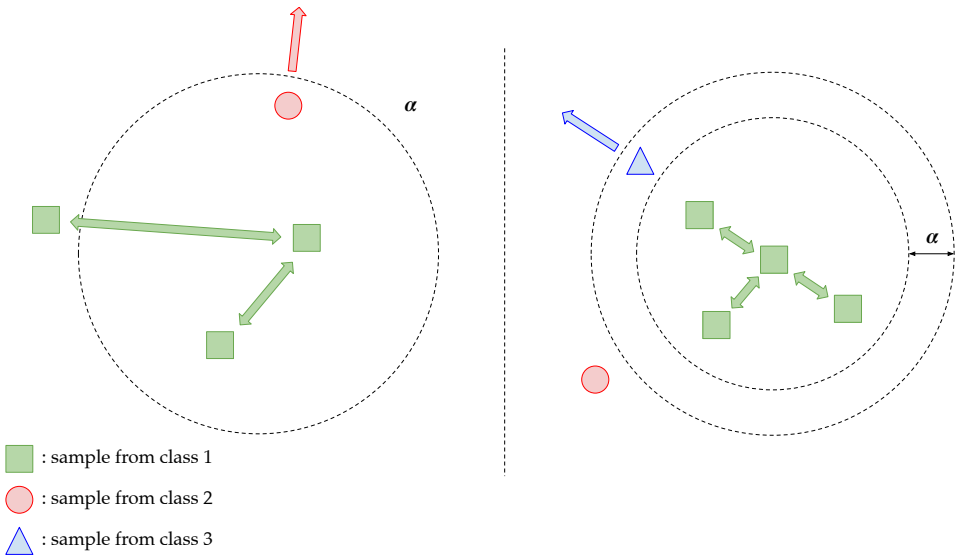

Figure 6.2: Schematic representation of the *contrastive loss* (Left) and the *large margin nearest neighbours loss* with $k = 3$ (Right). Samples movements during training are represented by large arrow. A double arrow indicates a *push* a single arrow indicates a *pull* between sample pairs.

In order to avoid the exclusive selection of easy triplets during training, Simo-Serra et al. (2015) propose to select only hard triplets in what they call a pair mining strategy and more precisely a *hard mining strategy*. Our strategy for music genre classification is based on the use of the *semi-hard triplet loss*.

6.3.2 *Triplet loss*

According to the literature but also for for the sake of readability, we redefine the syntax of a triplet as $\{a, p, n\}$ where a is an anchor, p one of its positive samples and n one of its negative samples. The triplet loss aims to minimize the distance between an anchor and a positive sample knowing that they have both the same class label and maximize the distance between the anchor and the negative knowing that they have a different class label. By pairing the samples into anchor/positive and anchor/negative triplets, the network learns the distribution of samples from each class with respect to all other classes.

The loss function is formulated as follow:

$$\mathcal{L}(a, p, n) = max(0, d(a, p) + \alpha - d(a, n)) \tag{6–23}$$

where $d(a, p)$ and $d(a, n)$ represent the Euclidean distances between the anchor and the positive or the negative respectively, and α is the margin parameter. The

anchor-positive should have a small distance between them whereas the anchor-negative should have a large distance.

To perform classification on large scale dataset, with a large amount of triplets, the hard-mining strategy shows some limitations. First, the selection of triplets during training can be ambiguous. Second, the training may be time consuming if the computation of all pairwise distance happened every n training step.

Schroff, Kalenichenko, and Philbin (2015) overcome these problems by proposing a *semi-hard* mining strategy. It consists in the selection of all embedding as anchor paired with all its positive in the successive batch. Each respective anchor/negatives pairs are selected according to three categories:

1. The easy negatives: $d(a, n) > d(a, p) + \alpha$
2. The semi-hard negatives: $d(a, p) < d(a, n) < d(a, p) + \alpha$
3. The hard negatives: $d(a, n) < d(a, p)$

The negatives are then selected at each training step according to few rules. If they are in the easy negative category, they can't be selected. If there is some negatives in the semi-hard category, the closest one to the positive is selected (Figure 6.3. Left). If there is only hard negatives, the closest one to the positive is also selected (Figure 6.3. Right).

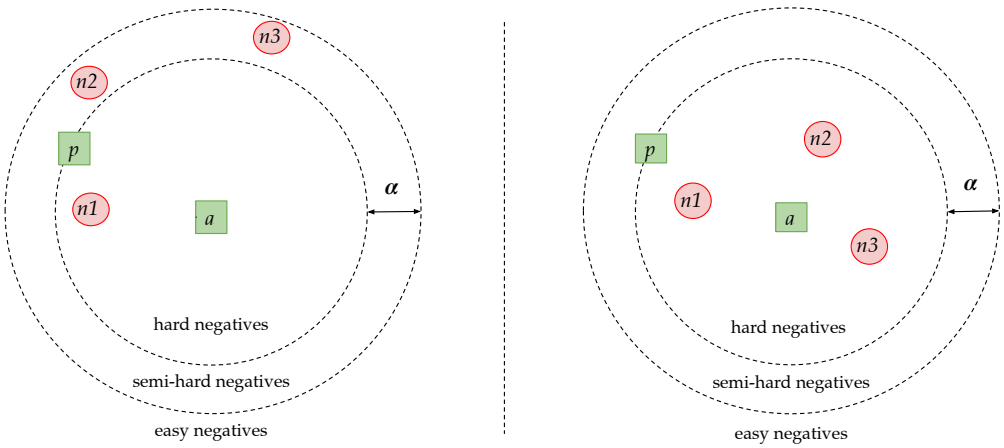

Figure 6.3: Schematic representation of the anchor/negative pairs (a, n) selection for a given anchor/positive (a, p) pair. (Left) Category **2.**, $n2$ is selected. (Right) Category **3.**, $n1$ is selected.

During training, the distance between anchor and semi-hard and hard negatives respectively keeps gradually increasing. This method has to respect some conditions in order to remain effective. To prevent the training from reaching a deadlock,

it is important to have a large training set with equally distributed classes. The use of large batches is also crucial to maintain a high amount of anchor/positive and anchor/negative pairs to learn the mining from.

Triplet loss has been used mainly for image recognition task (Oh Song et al., 2016; Sohn, 2016; He et al., 2016; He et al., 2018) and more recently in the MIR field for cover detection task (Doras and Peeters, 2020b; Doras and Peeters, 2020a).

In the next section, we apply the metric learning paradigm on the task of genre classification.

6.4 TRIPLET LOSS DEEP RHYTHM

As we have seen, metric learning has proven its effectiveness for image recognition tasks. A more recent study shows that it can be applied to music tagging tasks (Prétet, Richard, and Peeters, 2020) because it allows the study of similarities between samples of large-scale datasets. To use it in our context, we adapt our DR method to allow the application of the triplet loss. On the one hand, we want to apply it to different datasets annotated in genre in order to perform a classification. On the other hand, we want to study its impact on tempo estimation for the large-scale dataset. Given the subjective concept of similarity relative to the original space of our data, we want to learn a function f that will map this data onto a new space. In our case the term "sample" refers to a temporal slice of the HCQM of a track: τ'-HCQM.

In the context of genre classification, we hope that the model will learn the similarities between music tracks from the rhythmic representation of the HCQM.

In the following, we define the architecture of the model and the training process.

6.4.1 *Architecture*

We used the baseline model of Schroff, Kalenichenko, and Philbin (2015) for learning f using the triplet loss. We adapted it for the use of the HCQM as input of a DR convolutional network. We refer to this model as the TriDR.

The TriDR architecture is presented in Figure 6.4. The inputs of the network are the HCQM temporal slices of a given track τ-HCQM of size $(h \times b \times \Phi)$. It is followed by the DR model convolutional part with the same parameters described in details in Section 4.4. After flattening the last convolutional layer, a L2-normalization is applied on a dense layer which constitutes the output of the layer: the embedding vector of size E. An ELU activation function is applied to the last layer. We indicate that the model is trained according to the triplet loss to learn the transformation function f.

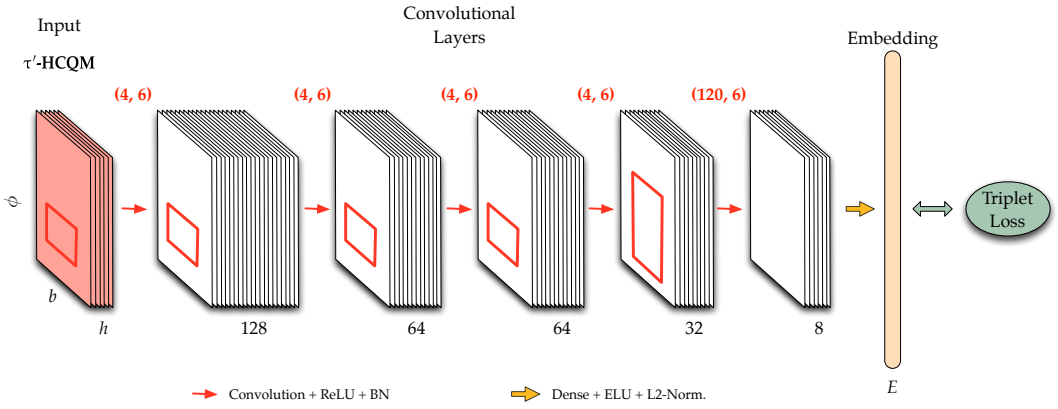

Figure 6.4: Architecture of the TriDR model.

6.4.2 *Training*

The semi-hard mining strategy, based on the triplet loss, aims to learn a transformation function f by selecting successively each sample in the batch as the anchor. As the anchor designates a temporal slice of a music track at a given moment of the training, this track will have as positive (resp. negative) the tracks belonging to the same class (resp. to another class). Through parameter optimization, we have set $\alpha = 1.5$ and $E = 128$. We perform a training using mini-batch gradient descent with batch of size B_s. It is important that an anchor have enough positive and negative sample in one batch. Thus, the dataset must be large enough and the classes evenly distributed. For this the batch-size must be set correctly, we set $B_s = 256$. The loss function is minimized thanks to ADAM optimizer and the number of epoch is automatically set using early stopping.

6.5 EVALUATION AND ANALYSIS

The embedding space resulting from the training allows to define a similarity between samples as the Euclidean distance between the embedding of our data. The observation of the embedding clusters in a 2-dimensional space already allows us to make early conclusions on the efficiency of the method. In order to obtain precise scores, a classifier trained on the embedding is necessary. Its role is to evaluate the probabilities of a test sample to belong to a class in the test samples based on the estimated distance between the embedding points. The distance is equivalent to the c function and can be the Euclidean distance.

Like the other methods we have developed, TriDR model is trained according to a 10-fold cross validation on each dataset for genre classification. For the sake of comparison, we also keep the same train/test/valid splits.

6.5.1 *Datasets*

It is mentioned in the literature (Schroff, Kalenichenko, and Philbin, 2015) that the larger the train dataset, the more efficient the triplet loss. The fact that the representations are separated in temporal slices for training already allows to increase the number of annotated samples. We therefore apply the TriDR method to the various genre annotated datasets presented in Chapter 3 namely BR, gEBR, GTzan, gMTG, lBP and sBP. We choose to not apply the TriDR method on the Gr dataset beacause of its small size.

6.5.2 *Embedding space*

6.5.2.1 *Protocol*

We apply a metric learning method based on the HCQM/DR combination. Datasets are labeled in genre, but the learning of f is done using a rhythm-oriented representation as input of the TriDR. Once f is learned, the next step is to evaluate its efficiency. To do so, we apply f to a set of test samples that are independent from the train samples. Since we perform 10-fold cross validation on the various datasets, the test set represent 1/10th of the whole dataset. First we project the test data in the embedding space through the transformation function f. As for other methods, a music track is divided in temporal slice (8 seconds long). We compute the median of all the temporal slice embedding for a given test sample to get only one embedding vector per test sample[1].

We represent the embedding space in 2-dimension in order to make a visual identification of the sample clusters. The embedding vectors obtained using the trained TriDR network are in E-dimension (E being the size of an embedding vector). Thus we want to reduce the E-dimension embedding vectors to 2-dimension. Several methods of dimensionality reduction for data visualization have been developed. Among the various dimentionality reduction algorithms, we have chosen the t-distributed Stochastic Neighbouring Entities (t-SNE) introduced by Maaten and Hinton (2008) for its flexibility. It allows the exploration of high-dimensional data. It is a non-linear algorithm and adapts to the underlying data which tries to preserve the local structure of data. The benefit of t-SNE is that it gives a guess about the number of close neighbors each sample has, also it can handle outliers. It is important to notice that the t-SNE algorithms help us for the visual observation of the data but don't give us a precise statistical score of good classification.

1 This allows to have more sparse visualization to identify the clusters

6.5.2.2 *Observations and discussion*

In this part, we observe the projection of each test sample contained in a specific fold in the embedding space in 2 dimensions. Since we use a 10-fold cross validation, we choose to represent the embedding space corresponding to the fold with the highest results in term of mean over class recall. We use here the same 10 folds as the previous methods. However, it appears that there are not enough examples per class in the case of the gMTG because it is strongly unbalanced (for example *dub* class has only one instance as discussed in Section 3.2). For the lBP, there are 30 different genre classes. It is difficult to visualize a space with so many classes in a figure. As a result, for these two datasets we choose not to represent their embedding spaces even if we applied the TriDR method to them.

For all the figures we display, one can see on the left the embedding space before training (i.e. the test data are projected to the space through a randomly initialized TriDR network) and on the right after training the TriDR network. Each class of the dataset is represented using a specific, unique color.

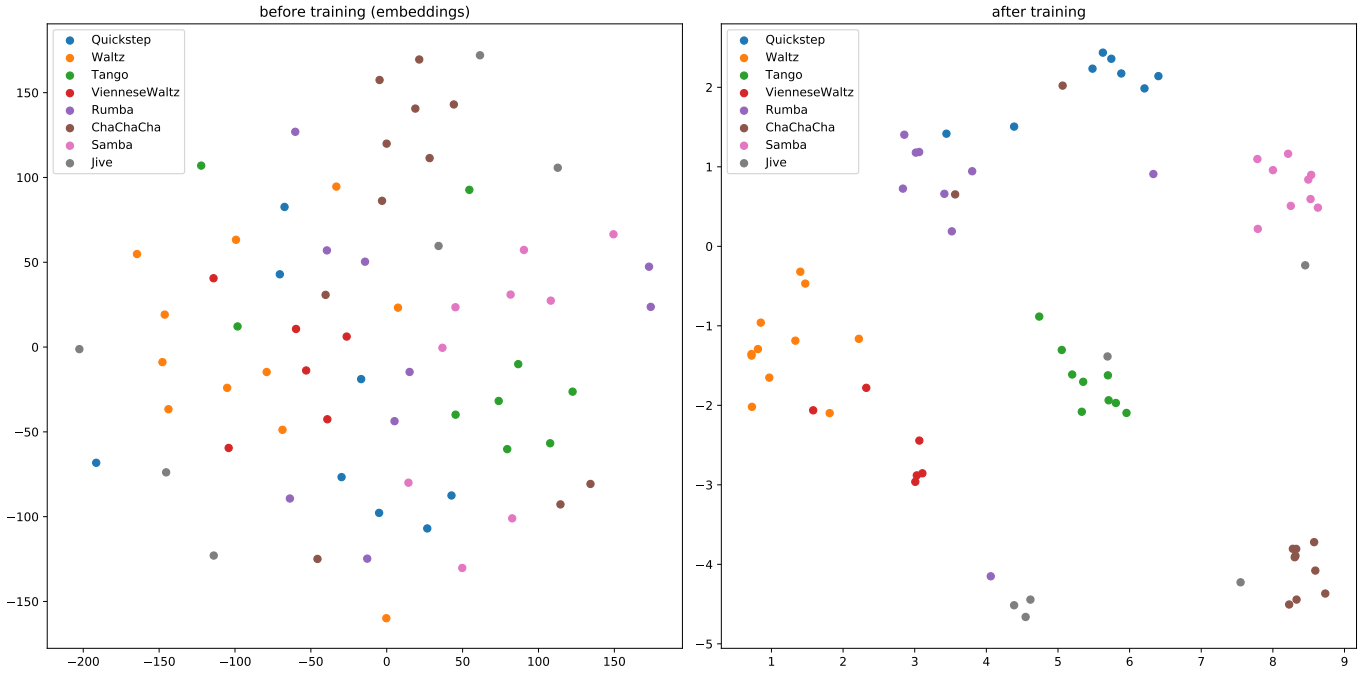

Figure 6.5: Embedding space for the BR dataset.

BALLROOM DATASETS. For the BR dataset (Figure 6.5), we can observe a well-defined cluster corresponding to genres. These results are even more pronounced for the gEBR dataset (Figure 6.6) knowing that there are many more examples per class. In addition, the clusters are more widely spaced, demonstrating that the

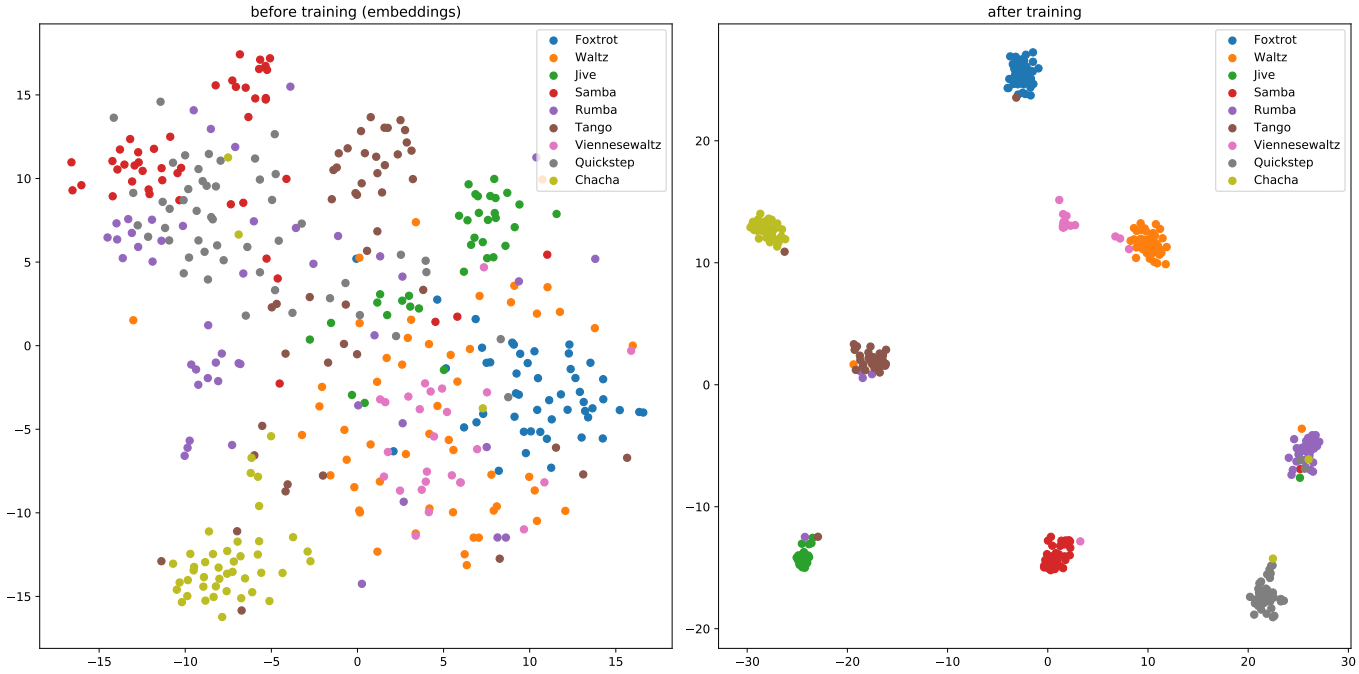

Figure 6.6: Embedding space for the gEBR dataset.

method works very well for these datasets. The efficiency of our methods based on the HCQM and thus on the harmonic components of the rhythm through analysis by convolutional neural networks is no longer to be proven on these two datasets. As a reminder from the Chapter 3, they were derived from the characteristics of ballroom music, music dedicated to dance and therefore inseparable from their rhythmic structure. Strictly based on the rhythmic aspects, we are able to classify these datasets. The similarities observed in the embedding spaces makes sense and explains the good results we have been able to obtain so far.

GTZAN DATASET. GTzan is a dataset whose labels are selected to correspond to the most popular genres of today popular music. It can thus already be seen that the projection of the test data does not allow the formation of genre-clusters as defined as for the previous datasets in Figure 6.7. However, it is interesting to note that it is still possible to identify scattered clusters. Among these, the genres *reggae*, *jazz*, *classical* or *hip-hop* are visually identifiable. Other genres are located "in the middle" of the space and even if groupings are observable, they are intermingled. We believe that for some genres (such as *reggae* or *hip-hop*), the rhythmic structure is essential to their description and prevails over other features, hence the clusters formation for these genres.

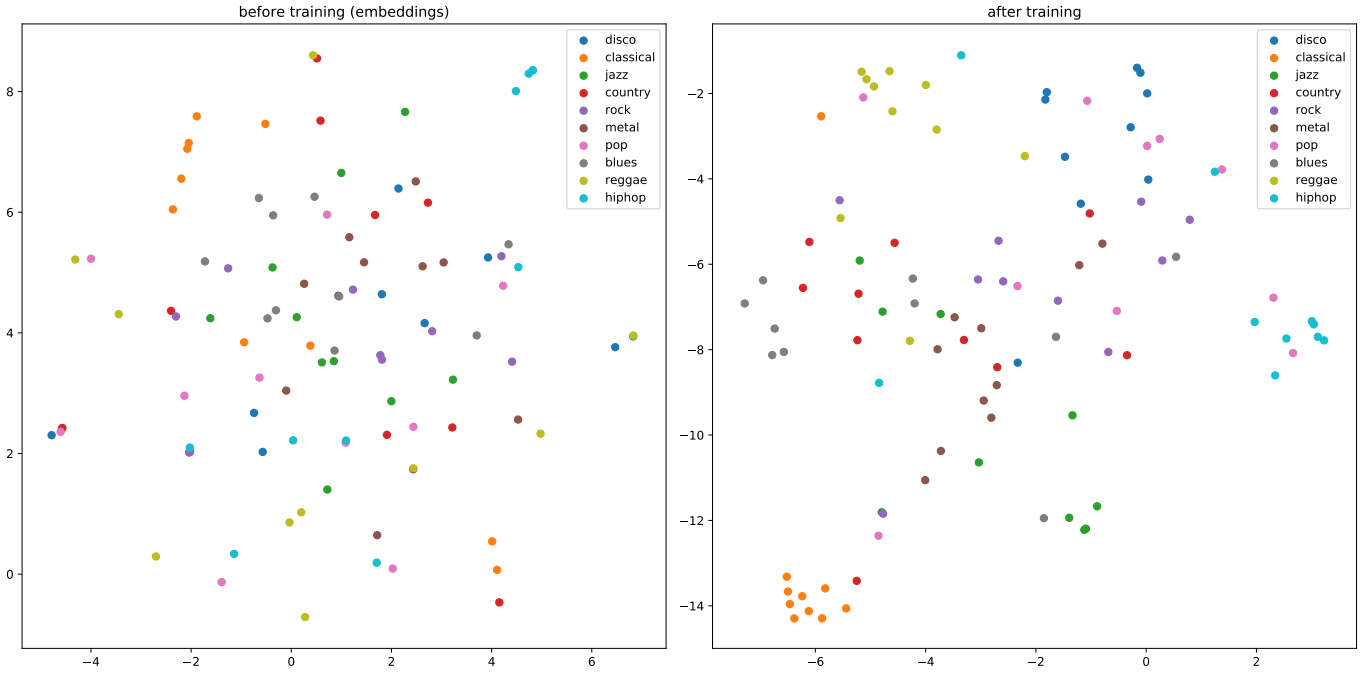

Figure 6.7: Embedding space for the GTzan dataset.

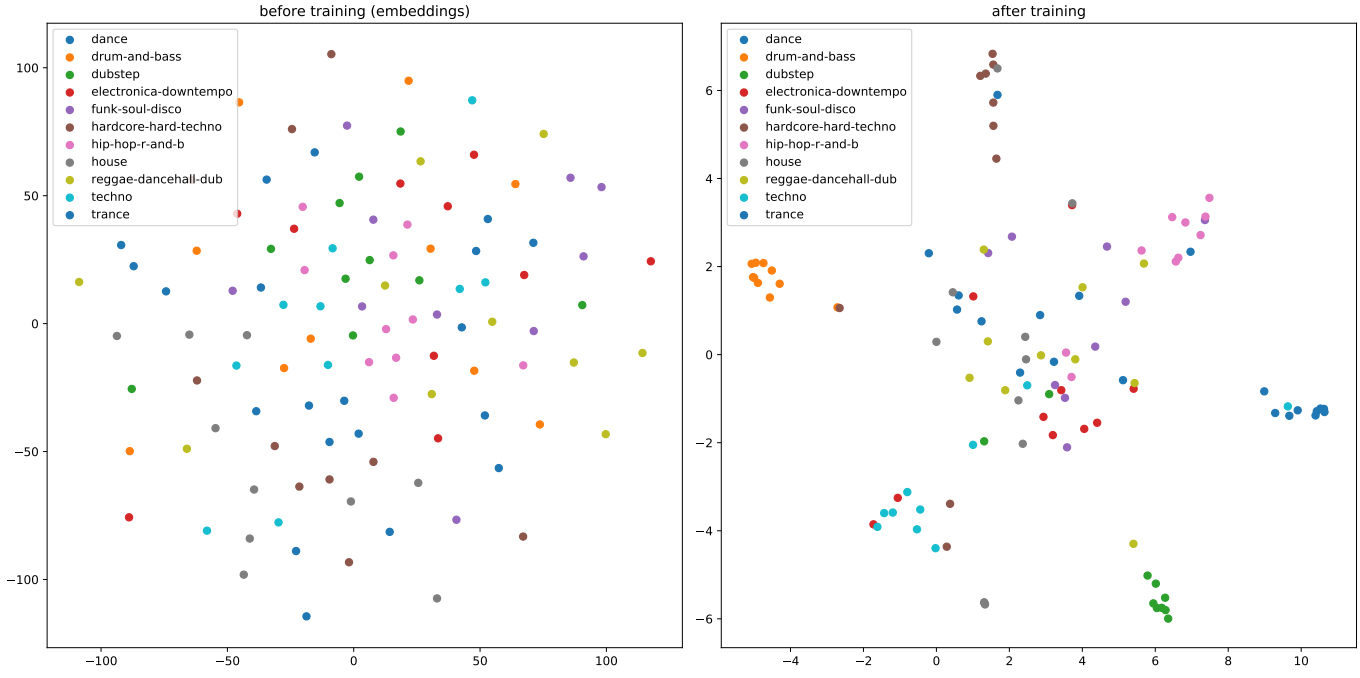

Figure 6.8: Embedding space for the sBP dataset.

BEATPORT SMALL DATASET. Finally, we analyze the use of metric learning on an EDM dataset, the sBP (Figure 6.8). By observing embedding space, we can

see the grouping in genre of 6 labels. On one side the breakbeat genres (such as *drum-and-bass* and *dubstep*) and on the other side the four-on-the-floor genres (such as *techno*, *trance*, *hard-techno*) are gathered in distinct clusters. The examples labeled *reggae-dancehall-dub*, *house*, *down-tempo* and *funk-soul-disco* are located in the "center" of the space and remain indistinctly mixed. These observations echo our previous analysis. Indeed, the genres with the most distinctive rhythmic structures such as the breakbeat pattern genres have well-defined clusters. The examples belonging to genres with more ambiguous rhythmic structures are intermingled such as *dance* with *disco* and *house*, the latter have similar rhythmic structures, so it is not surprising to find this mixture of clusters on the embedding space. One can also observe a certain logic when some examples of *hard-techno* are located within the cluster related to *techno*. It should be noted that some *hardcore* tracks may contain breakbeat patterns while some *hard-techno* tracks are structured with four-on-the-floor patterns (as *techno*). The mixing of both genres included in the label *hardcore-hard-techno* could explain the confusion with the *techno* example. We can conclude that a metric learning paradigm with the HCQM as an input to the DR network allows an estimation of similarity between the music tracks that makes sense.

6.5.3 *Classifiers*

To evaluate the ability of TriDR to learn projections (embedding) that are close to their similar siblings we need to analyze the distance between them. *Classifiers* are the algorithms that evaluate that distance in order to assign a class to each sample. To train a classifier we use the trained embedding vectors of size E as input. We then evaluate the classification performances on the test data. We tested two types of classifiers, the k-*NN* and the *Nearest Centroid* (*NC*).

k-NN. An unlabeled sample (query point) is classified by assigning the label which is most frequent among its k nearest training samples. The k value represents the margin that encompasses the k training observations around the query point. Within this margin, the label assigned to the query is the one of the most represented class.

NC. Each class is represented by its centroid in the embedding space (equivalent to the mean of the training samples), the test samples are classified by assigning to the query the label of the class of training samples whose centroid is the closest to the query.

6.5.4 *Classification Results*

Because the results obtained using the NC are higher than those using k-NN, we have chosen to only comment the former. We compare the efficiency of the TriDR for rhythm-oriented genre classification in terms of average recall to the original DR method and with the best of our methods for this task, the MI.

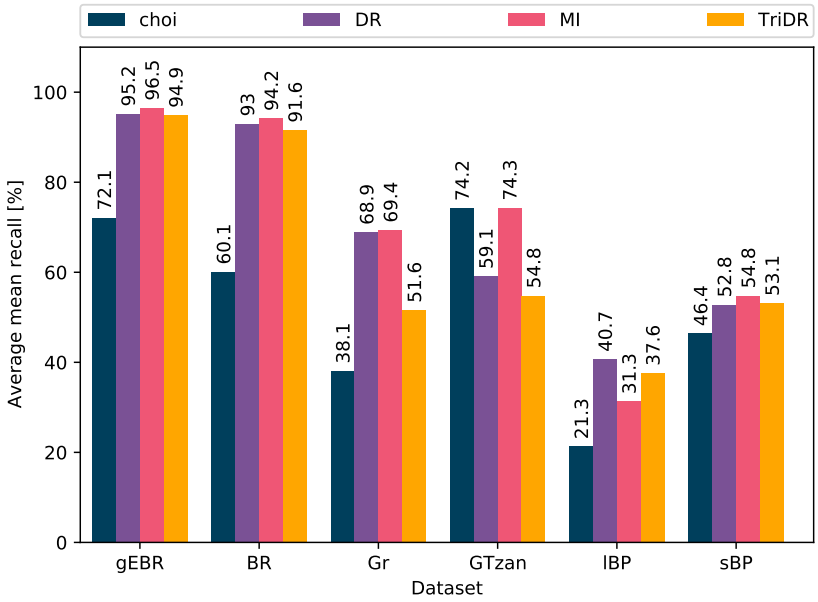

Figure 6.9: Results comparison between TriDR and previous methods in terms of average mean recall.

We compare the results in Figure 6.9: the baseline method *choi*, the original DR method, the most efficient method MI and the TriDR method. First of all, results obtained using the *choi* method appear to be largely below the TriDR ones for all datasets (except for the GTzan). TriDR performs almost as DR or as MI, for gEBR, BR, lBP while for the sBP EDM dataset results are slightly higher (+0.3%) than DR. Even if the results obtained with the TriDR method are not superior to those obtained with the MI, method we can see that treating the similarity between pieces using a metric learning paradigm allows a good classification into genres.

It is satisfying to note that for the balanced dataset of EDM sBP the results are superior to the original DR method. They are also better than MI method for lBP which contains more genres. The analysis of the similarities between the tracks allows us once again to conclude that the genres of EDM can be retrieved according to their rhythmic structure.

On the confusion matrix in Figure 6.10, we can observe a certain concordance with the embedding space of the sBP dataset of Figure 6.8. For example, we can

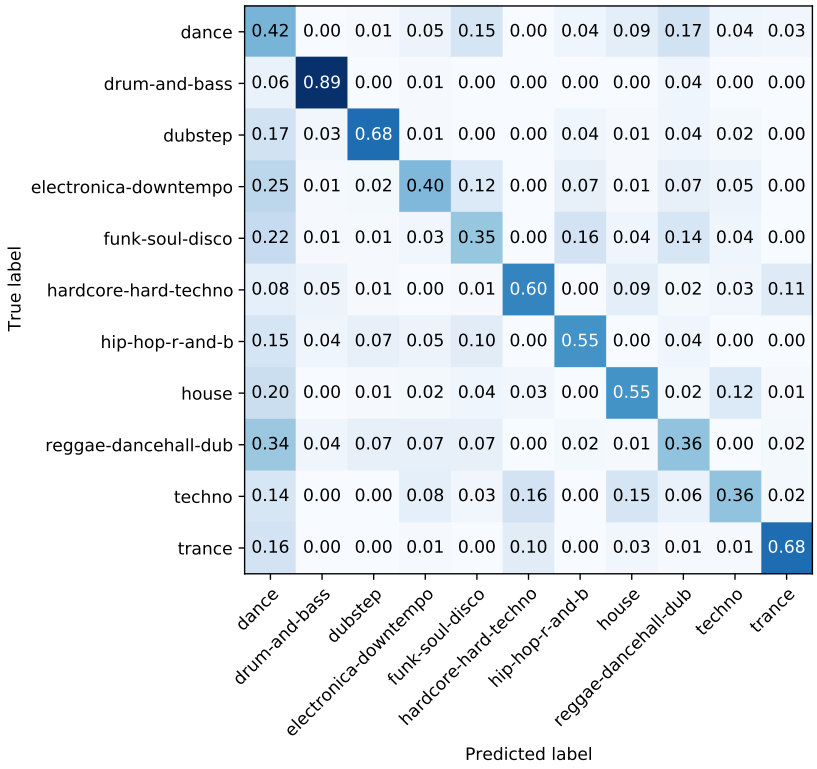

Figure 6.10: Confusion matrix of the evaluation of the sBP dataset using TriDR method.

see that some of the examples in each class are classified as *dance*. The cluster corresponding to this genre is located at the intersection of the clusters and is intertwined with several other genres. This distribution can be explained by the fact that *dance* is a kind of ambiguous EDM genre (as we have seen in previous analyses).

6.6 CONCLUSION

In this chapter, we have proposed another approach to perform a rhythm-oriented genre classification. We have developed a method to estimate automatically the similarity between tracks strictly based on their rhythm characteristics. To do so, we have studied the metric learning principle. We started from a general description that has gradually evolved through various works of the state of the art. Thus the learning metric allows us to automatically learn the distance or metric most likely to represent similarities between two samples. Indeed, in applying such a paradigm one expects similar samples to be close and dissimilar samples to be distant. Using the learning metric, we learn a function that projects the original data into a new trained space, the embedding space. From a mathematical point

of view, several losses have been defined in order to allow the training of models able to learn embedding vectors. Among these, we have selected the triplet loss and adapted it to our task. During training, each sample is successively selected as an anchor and is compared to a positive (a sample from the same class) and a negative (a sample from another class).

The rhythm characteristics are once again represented through the HCQM as input to a convolutional neural network with the same architecture than in the original DR method. We replaced the dense output layers by an embedding vector preceded by an L2-normalization. We called this method TriDR. During training the network learns these embedding vectors from the genre label assigned to the HCQM time slice of the input track. These vectors of size 128 are then reduced to two dimensions thanks to the t-SNE algorithm which allows us to visualize the embedding space.

By projecting the test data in the embedding space, we can observe the cluster formation depending on the similarity/dissimilarity between the examples. We have seen that for BR and gEBR datasets these clusters are perfectly defined. This shows that the method works well in grouping examples with a similar rhythmical structure and in spacing these clusters when they correspond to dissimilar classes. The observation of the embedding space of the sBP dataset reinforced our assumptions and the results to the previous methods. The genre groupings correspond to the rhythmic similarity shared by the different classes. It highlights some ambiguities in the very definition of these genres when analyzing more precisely their rhythmic characteristics. These results are also reflected in the observation of the confusion matrix of this EDM dataset.

To sum up, by applying a metric learning paradigm and analyzing the embedding spaces obtained, we were able to highlight the fact that the use of a descriptor and a network dedicated to rhythm allows a good classification in genre based on the similarities between pieces of music. When genres are defined by their rhythmic structure, this analysis is all the more relevant (as in the case of EDM genres).

Nevertheless, we have limited ourselves to a single metric learning method and we assume that this work remains open to improvements. Recent works in MIR such as the one of Lee et al. (2020) shows us other facets of the use of metric learning than triplet loss for classification tasks. In our future work, it could be interesting to explore other techniques such as proxy-based models.

7 CONCLUSION

To conclude this manuscript, we review in Section 7.1 all the chapters highlighting our main contributions. We then present in Section 7.2 the various points which would deserve to be extended in future work. Finally, in Section 7.3 we end with an overall conclusion on the work we have done during this thesis.

7.1 SUMMARY AND MAIN CONTRIBUTIONS

In Chapter 1, we introduced the manuscript by defining two MIR tasks: (i) automatic tempo estimation and (ii) automatic rhythm-oriented genre classification. We also set the context of our thesis, namely the analysis of EDM genres from their rhythmic description. In other words, the guideline of our thesis was to analyze the rhythmic components of music (in particular through tempo estimation) as the main descriptor of the musical genre, especially in the case of EDM. In order to better understand the origin of the flourishing taxonomy of EDM, we proposed a brief history of this type of music.

In Chapter 2, we presented the state-of-the-art methods dedicated to rhythm description, tempo estimation and genre classification. We also got familiar with the set of tools that we used to develop our methods. We defined the rhythm and its components (metrical structure, tempo and timing) and the musical genre. We have seen that both of these musical concepts do not have a clear, formal definition. The presentation of the state-of-the-art work allowed a general understanding of these concepts. The systems dedicated to rhythm analysis and genre classification fall into two categories: the handcrafted (based on music knowledge) and the data-driven (based on the processing of large scale datasets). For the first category basic signal processing tools as well as the different features calculated for the realization of these automatic analysis were exposed. For the second category, we detailed the general concepts of machine learning and deep learning, with a meaningful focus on supervised learning. Finally, we presented the MIR works related to EDM, which emphasizes rhythm as an essential element for the description of musical genre.

In Chapter 3, we described the commonly-used datasets for the task of tempo estimation. Some of them are also used for genre classification since they can

also annotated into genres. We then introduced two datasets that we developed in order to evaluate our method in the perspective of EDM. To do this, we have gathered audio excerpts annotated in genres from Beatport. We made sure to create two datasets balanced in genres. One is annotated with the full Beatport taxonomy (lBP), and for the second we selected some of these genres to highlight differences/similarities in rhythmic structure and tempo ranges (sBP).

In Chapter 4, we described our first method: Deep Rhythm. Its purpose is to estimate the global tempo value of a music track and also to classify it according to its musical genre. These two operations are performed separately but using the same framework. In order to understand the development of this method, we have presented the various works used for inspiration. Starting from a representation of the harmonic components of the rhythm, we proposed the computation of a representation called the HCQM. This 4-dimensional representation allows to represent on different frequency bands the periodic rhythmic components of the onsets of the signal of a music track contained in a time window of 8 seconds. We have been inspired by the work of Peeters (2011) for the harmonic analysis of rhythm and of Marchand and Peeters (2016a) for the separation into frequency bands. The fourth dimension allows to shift the rhythmic components and thus to identify the fundamental frequency – directly corresponding to the global tempo of the music track – and its harmonics. This dimension called h is primarily inspired by the work of Bittner et al. (2017) that relies on such a representation at the input of a CNN (where h is the depth) for fundamental frequency estimation. We also train a CNN for the automatic classification into tempo classes and into music genre classes using the HCQM as input.

We conducted a strong evaluation to analyze the results of the method.

On one hand, we evaluated the model for tempo estimation by performing a large-scale analysis (i.e. 3 large datasets for the training of our network and 7 datasets for the evaluation). We compared the results to the ones of state-of-the-art systems. Thus, we found out that our method performs better than these systems for 2 of the 7 datasets tested: BR dataset of ballroom music whose genres correspond to rhythmic patterns and tGS dataset of EDM. The results of our method with oracle estimation is also at the state of the art for almost all datasets. Finally, by applying a ground-truth smoothing we obtained the best results in terms of Accuracy1.

On the other hand, we evaluated the performance of our model for the classification in musical genres. We mentioned rhythm-oriented classification because our HCQM representation, describing the rhythm components, was given as input to the network. We evaluated each dataset independently for this task because the labels were not the same from one dataset to another. We have observed good

results in terms of average mean recall for all datasets even if they only surpassed the state of the art for the gEBR dataset. By observing the confusion matrix of EDM sBP dataset, we concluded that our method succeded to classify the different genres in a relevant way (with respect to their rhythmic description and tempo ranges on which they are defined). The confusions observed make sense because they mainly concern the most ambiguous EDM genres, i.e. those that can be inseparable from the point of view of their rhythmic structure.

In Chapter 5, we put forward three extensions of the DR method. Each of them consider a different aspect inherent to music.

The first proposal is to consider the possible relationships between the different frequency bands. Indeed, we have seen that the rhythmic structure rely on the relation between the components detected in various frequency bands. Thus, we opted for the use of complex inputs and convolution by using a complex-valued HCQM as input. The results obtained using this method were higher than those of the original DR method for tempo estimation (with oracle selection) but lower for genre classification.

The second idea is to take into account the rhythm/genre relationship and treat them as two intrinsically linked concepts. Thus, we developed a joint learning method through a multitask learning paradigm, allowing to classify in genre and tempo simultaneously. The model was parameterized so that the learning procedure would take into account the influences of both tasks. The results obtained were encouraging, given that the network has only half the number of parameters of the original DR network. We also found out that such a method has an impact on the genre classification of EDM since the results for datasets associated with this type of music were better.

The third extension is to consider a second musical descriptor in addition to rhythm as input of our network. We have chosen to add a branch dedicated to timbre analysis to the CNN of DR with a VGG-Net architecture and a mel-spectrogram input. The objective of this branch is to let the model to capture features related to timbre alongside features related to rhythm for genre classification. The results for the classification in genre surpassed the previous ones in terms of average mean recall.

Finally, in Chapter 6, we proposed a metric learning paradigm to analyze the automatic genre classification based on the learning of the similarity between tracks. We first described the metric learning paradigm and in particular the triplet loss. This type of method allowed us to learn a function whose role is to project the data in an embedding space such that the distance between the embbeded data informed about their similarity/dissimilarity. We adapted the DR method to

a metric learning paradigm. To evaluate this method (named TriDR), we have visualized the embedding spaces of the test data of each dataset. From the latter, we were able to identify clusters specific to each genre. In the case of the sBP dataset, the observations allowed us to validate one more time the rhythmic characteristics of EDM genres. Indeed, we managed to map the different genres with respect to their respective similarities. Finally, by applying a simple classifier to the predicted embedded data, we achieved a classification into musical genres. The results were not as high as the MI method but still were encouraging.

7.2 FUTURE WORKS

To extend the research done during the thesis, we are propose here some future works. Some of them may have already been addressed in the course of our manuscript, we have chosen to leave them aside in order to concentrate more effectively on the different methods developed in this manuscript.

In Chapter 3, we mentioned that the use of annotated datasets may have some limitations. Among those, we can cite a problem inherent to deep learning when training networks on a limited amount of data. Indeed, if we had access to annotated datasets in tempo or genre on a very large scale, we could reinforce the robustness of our models and allow them to better generalize. Unfortunately, access to such datasets has not been possible for copyright reasons or simply because very large-scale datasets annotated in tempo are not available. As far as EDM is concerned, having access to correct annotation in global tempo value for the 64,000 tracks we gathered from Beatport would have allowed us to extend our analysis. To do so, it would be sufficient to analyze the comments collected on Beatport to correct the initial global tempo annotations. Note that for the genre classification of EDM, we could have use the 64,000 annotated tracks for larger-scale analysis. However, the computation time involved would have been considerable.

In Chapter 4, we discussed on the Oracle frame selection method. The results of it showed that our DR method and some of its extensions have a certain margin in terms of automatically estimating the best temporal candidate for the global tempo value. An attempt at automatically selecting the frames was done by developing an attention mechanism. Other techniques have been explored without success (such as the use of RNN). As the attention mechanism has shown good results in other tasks of music classification, we believe that this lead should not to be excluded. Knowing that the HCQM is a 4-dimensional representation, we would like to find a way to train an end-to-end network with the attention mechanism as an output. We believe that with a larger computation capacity we could obtain better results.

As we have seen in Chapter 5, the multi-input method proved to be quite promising. In this, we have added another type of features related to timbre (as a complement to rhythm) by combining mel-spectrograms with a commonly used CNN. Adding other types of timbre-related, and pitch-related as well, features would be a good starting point towards further investigations.

7.3 OVERALL CONCLUSION

Our main objective was to highlight the importance of the rhythmic structure of a piece of music for the description of its genre, especially in the case of Electronic/Dance Music. Indeed, with the constant emergence of new genres in music (as it is the case in EDM), the categorization of music does no longer only depend on its musical characteristics. Through the different methods we have developed and their analysis, we have emphasized the rhythmic relationships between some of the main genres in EDM. These methods have been developed with the additional objective of performing the two tasks of tempo estimation and classification into musical genres automatically. We have developed a method allowing, not without limits, to perform these two tasks while taking into account an essential aspect of music: rhythm.

BIBLIOGRAPHY

Atlas, Les and Shihab A Shamma (2003). 'Joint acoustic and modulation frequency.' In: *Advances in Signal Processing, EURASIP Journal on* 2003, pp. 668–675 (cit. on p. 58).

Aucouturier, Jean-Julien and Francois Pachet (2003). 'Representing musical genre: A state of the art.' In: *Journal of New Music Research* 32.1, pp. 83–93 (cit. on pp. 16, 22).

Baxter, Jonathan (1997). 'A Bayesian/information theoretic model of learning to learn via multiple task sampling.' In: *Machine learning* 28.1, pp. 7–39 (cit. on p. 91).

Bello, Juan Pablo et al. (2004). 'On the use of phase and energy for musical onset detection in the complex domain.' In: *IEEE Signal Processing Letters* 11.6, pp. 553–556 (cit. on p. 19).

Bello, Juan Pablo et al. (2005). 'A tutorial on onset detection in music signals.' In: *Audio, Speech and Language Processing, IEEE Transactions on* 13.5, pp. 1035–1047 (cit. on p. 19).

Bertin-Mahieux, Thierry et al. (2011). 'The million song dataset.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)* (cit. on p. 98).

Bilmes, Jeffrey Adam (1993). 'Timing is of the essence: Perceptual and computational techniques for representing, learning, and reproducing expressive timing in percussive rhythm.' PhD thesis. Massachusetts Institute of Technology (cit. on p. 13).

Bittner, Rachel M, Brian McFee, and Juan P Bello (2018). 'Multitask learning for fundamental frequency estimation in music.' In: *arXiv preprint arXiv:1809.00381* (cit. on p. 91).

Bittner, Rachel M et al. (2017). 'Deep Salience Representations for Fo Estimation in Polyphonic Music.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)*. Suzhou, China (cit. on pp. 55, 57–62, 119).

Böck, Sebastian and E. P. Matthew Davies (2020). 'Deconstruct, Analyse, Reconstruct: How to Improve Tempo, Beat, and Downbeat Estimation.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)*. Montreal, Canada, pp. 574–582 (cit. on p. 32).

Böck, Sebastian, Matthew EP Davies, and Peter Knees (2019). 'Multi-task Learning of Tempo and Beat: Learning One to Improve the Other.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)* (cit. on pp. 32, 72, 91).

Böck, Sebastian, Florian Krebs, and Gerhard Widmer (2015). 'Accurate Tempo Estimation Based on Recurrent Neural Networks and Resonating Comb Filters.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)*. Malaga, Spain (cit. on pp. 32, 37, 65, 68).

Böck, Sebastian and Markus Schedl (2011). 'Enhanced beat tracking with context-aware neural networks.' In: *Proc. of DAFx (International Conference on Digital Audio Effects)*, pp. 135–139 (cit. on p. 32).

Bogdanov, Dmitry et al. (2013). 'Essentia: An audio analysis library for music information retrieval.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)*. Curitiba, PR, Brazil (cit. on p. 35).

Brewster, Bill and Frank Broughton (2014). *Last night a DJ saved my life: The history of the disc jockey*. Open Road+ Grove/Atlantic (cit. on p. 6).

Brown, Judith C and Miller S Puckette (1992). 'An efficient algorithm for the calculation of a constant Q transform.' In: *JASA (Journal of the Acoustical Society of America)* 92.5, pp. 2698–2701 (cit. on p. 56).

Butler, Mark Jonathan (2006). *Unlocking the groove: Rhythm, meter, and musical design in electronic dance music*. Indiana University Press (cit. on pp. 4, 7, 34).

CS 230 - Deep Learning. https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks#. Accessed: 2020-09-30 (cit. on pp. 26–29, 31).

Camara, Luis Gomez (2017). 'Feature Analysis and Subgenre Classification of Electronic Dance Music [Master's Thesis].' In: (cit. on p. 35).

Caparrini, Antonio et al. (2020). 'Automatic subgenre classification in an electronic dance music taxonomy.' In: *Journal of New Music Research*, pp. 1–16 (cit. on p. 35).

Cemgil, Ali Taylan et al. (2000). 'On tempo tracking: Tempogram representation and Kalman filtering.' In: *Journal of New Music Research* 29.4, pp. 259–273 (cit. on p. 21).

Chen, Ching-Wei et al. (2009). 'Improving perceived tempo estimation by statistical modeling of higher-level musical descriptors.' In: *Audio Engineering Society Convention 126*. Audio Engineering Society (cit. on p. 32).

Chen, Joyce L, Virginia B Penhune, and Robert J Zatorre (2008). 'Listening to musical rhythms recruits motor regions of the brain.' In: *Cerebral cortex* 18.12, pp. 2844–2854 (cit. on p. 11).

Choi, Hyeong-Seok et al. (2019). 'Phase-aware speech enhancement with deep complex u-net.' In: *arXiv preprint arXiv:1903.03107* (cit. on p. 83).

Choi, Keunwoo, George Fazekas, and Mark Sandler (2016). 'Automatic tagging using deep convolutional neural networks.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)* (cit. on pp. 33, 82, 95–98).

Choi, Keunwoo et al. (2017). 'Convolutional recurrent neural networks for music classification.' In: *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*. IEEE. New Orleans, USA, pp. 2392–2396 (cit. on pp. 33, 95–97).

Chopra, Sumit, Raia Hadsell, and Yann LeCun (2005). 'Learning a similarity metric discriminatively, with application to face verification.' In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. IEEE, pp. 539–546 (cit. on pp. 103, 104).

Clevert, Djork-Arné, Thomas Unterthiner, and Sepp Hochreiter (2016). 'Fast and accurate deep network learning by exponential linear units (elus).' In: *International Conference on Learning Representations* (cit. on pp. 27, 63).

Cohen-Hadria, Alice and Geoffroy Peeters (2017). 'Music structure boundaries estimation using multiple self-similarity matrices as input depth of convolutional neural networks.' In: *Audio Engineering Society Conference: 2017 AES International Conference on Semantic Audio* (cit. on p. 96).

Collins, Nick et al. (2013). *Electronic music*. Cambridge University Press (cit. on p. 6).

Collobert, Ronan and Jason Weston (2008). 'A unified architecture for natural language processing: Deep neural networks with multitask learning.' In: *Proceedings of the 25th international conference on Machine learning*, pp. 160–167 (cit. on p. 91).

Cooper, Grosvenor W, Grosvenor Cooper, and Leonard B Meyer (1963). *The rhythmic structure of music*. University of Chicago Press (cit. on pp. 11–13).

Davies, Matthew EP and Mark D Plumbley (2007). 'Context-dependent beat tracking of musical audio.' In: *Audio, Speech and Language Processing, IEEE Transactions on* 15.3, pp. 1009–1020 (cit. on p. 20).

Deng, Li, Geoffrey Hinton, and Brian Kingsbury (2013). 'New types of deep neural network learning for speech recognition and related applications: An overview.' In: *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*. IEEE. Vancouver, British Columbia, Canada, pp. 8599–8603 (cit. on p. 91).

Deshpande, Hrishikesh, Unjung Nam, and Rohit Singh (2001). 'Mugec: Automatic music genre classification.' In: *Technical Report, Stanford University* (cit. on p. 21).

Diakopoulos, Dimitri et al. (2009). '21st Century Electronica: MIR Techniques for Classification and Performance.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)*. Kobe, Japan, pp. 465–470 (cit. on p. 35).

Dieleman, Sander and Benjamin Schrauwen (2014). 'End-to-end learning for music audio.' In: *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*. Florence, Italy, pp. 6964–6968 (cit. on p. 96).

Dixon, Simon (2006). 'Onset detection revisited.' In: *Proc. of DAFx (International Conference on Digital Audio Effects)*. Vol. 120. Citeseer. Montreal, Canada, pp. 133–137 (cit. on p. 19).

Dixon, Simon and Emilios Cambouropoulos (2000). 'Beat tracking with musical knowledge.' In: *ECAI*, pp. 626–630 (cit. on p. 13).

Dixon, Simon, Elias Pampalk, and Gerhard Widmer (2003). 'Classification of dance music by periodicity patterns.' In: (cit. on p. 20).

Doras, Guillaume and Geoffroy Peeters (2020a). 'A Prototypical Triplet Loss for Cover Detection.' In: *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*. IEEE. Barcelona, Spain, pp. 3797–3801 (cit. on p. 108).

— (2020b). 'Cover detection using dominant melody embeddings.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)* (cit. on p. 108).

Drake, Carolyn and Caroline Palmer (2000). 'Skill acquisition in music performance: Relations between planning and temporal control.' In: *Cognition* 74.1, pp. 1–32 (cit. on p. 14).

Duong, Long et al. (2015). 'Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser.' In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 845–850 (cit. on p. 91).

Ellis, Daniel PW (2007). 'Beat tracking by dynamic programming.' In: *Journal of New Music Research* 36.1, pp. 51–60 (cit. on p. 20).

Ermolinskiy, A, P Cook, and G Tzanetakis (2001). 'Musical Genre classification based on the analysis of harmonic content features in audio and midi.' In: *Work Report* (cit. on p. 22).

Faraldo, Angel, Sergi Jorda, and Perfecto Herrera (2017). 'A multi-profile method for key estimation in EDM.' In: *Audio Engineering Society Conference: 2017 AES International Conference on Semantic Audio*. Audio Engineering Society (cit. on p. 42).

Fikentscher, Kai (2000). *" You Better Work!": Underground Dance Music in New York*. Wesleyan University Press (cit. on p. 6).

Foote, Jonathan, Matthew L Cooper, and Unjung Nam (2002). 'Audio Retrieval by Rhythmic Similarity.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)*. Paris, France (cit. on p. 21).

Foroughmand, Hadrien and Geoffroy Peeters (2017). 'Multi-source musaicing using non-negative matrix factor 2-d deconvolution.' In: *18th International Society for Music Information Retrieval Late-Breaking Demo Session* (cit. on p. 1).

Foroughmand, Hadrien and Geoffroy Peeters (2018). 'Music retiler: Using NMF2D source separation for audio mosaicing.' In: *Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion*. (ACM), p. 27 (cit. on p. 1).

— (2019). 'Deep-Rhythm for Tempo Estimation and Rhythm Pattern Recognition.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)*. Delft, The Netherlands (cit. on p. 80).

— (2020). 'Extending deep rhythm for tempo and genre estimation using complex convolutions, multitask learning and multi-input network.' In: *Proceedings of Joint Conference on AI Music Creativity, Royal Institute of Technology (KTH), Stockholm, Sweden*. (MuMe) (cit. on p. 101).

Fortmann-Roe, S (2012). *Understanding the Bias-Variance Tradeoff*. http://scott.fortmann-roe.com/docs/BiasVariance.html. Accessed: 2020-09-30 (cit. on p. 25).

Fraisse, Paul (1974). *Psychologie du rythme*. FeniXX (cit. on p. 11).

— (1982). 'Rhythm and tempo.' In: *The psychology of music* 1, pp. 149–180 (cit. on p. 11).

Friberg, Anders and Andreas Sundström (1997). 'Preferred swing ratio in jazz as a function of tempo.' In: *TMH-QPSR* 38.4, pp. 019–027 (cit. on p. 15).

Gainza, Mikel and Eugene Coyle (2011). 'Tempo detection using a hybrid multiband approach.' In: *Audio, Speech and Language Processing, IEEE Transactions on* 19.1, pp. 57–68 (cit. on p. 20).

Gemmeke, Jort F et al. (2017). 'Audio set: An ontology and human-labeled dataset for audio events.' In: *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*. IEEE. New Orleans, USA, pp. 776–780 (cit. on p. 65).

Giannakopoulos, Theodoros (2015). 'pyaudioanalysis: An open-source python library for audio signal analysis.' In: *PloS one* 10.12, e0144610 (cit. on p. 35).

Gilbert, Jeremy, Ewan Pearson, et al. (1999). *Discographies: Dance music, culture and the politics of sound*. Psychology Press (cit. on p. 5).

Girshick, Ross (2015). 'Fast r-cnn.' In: *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448 (cit. on p. 91).

Gkiokas, Aggelos, Vassilios Katsouros, and George Carayannis (2012). 'Reducing Tempo Octave Errors by Periodicity Vector Coding And SVM Learning.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)*. Porto, Portugal (cit. on p. 32).

Goldberger, Jacob et al. (2005). 'Neighbourhood components analysis.' In: *Advances in neural information processing systems*, pp. 513–520 (cit. on p. 103).

Goodfellow, Ian et al. (2016). *Deep learning*. Vol. 1. 2. MIT press Cambridge (cit. on p. 30).

Goto, Masataka (2001). 'An audio-based real-time beat tracking system for music with or without drum-sounds.' In: *Journal of New Music Research* 30.2, pp. 159–171 (cit. on p. 20).

Goto, Masataka and Yoichi Muraoka (1994). 'A beat tracking system for acoustic signals of music.' In: *Proceedings of the second ACM international conference on Multimedia*. ACM, pp. 365–372 (cit. on p. 31).

Gouyon, Fabien and Perfecto Herrera (2003). 'Determination of the meter of musical audio signals: Seeking recurrences in beat segment descriptors.' In: *Audio Engineering Society Convention 114*. Audio Engineering Society (cit. on p. 21).

Gouyon, Fabien et al. (2004). 'Evaluating rhythmic descriptors for musical genre classification.' In: *Proceedings of the AES 25th International Conference*, pp. 196–204 (cit. on p. 38).

Gouyon, Fabien et al. (2005). *A computational approach to rhythm description-Audio features for the computation of rhythm periodicity functions and their use in tempo induction and music content processing*. Universitat Pompeu Fabra (cit. on pp. 11–13).

Gouyon, Fabien et al. (2006). 'An experimental comparison of audio tempo induction algorithms.' In: *Audio, Speech and Language Processing, IEEE Transactions on* 14.5, pp. 1832–1844 (cit. on pp. 16, 21, 41, 95).

Grahn, Jessica A and Matthew Brett (2007). 'Rhythm and beat perception in motor areas of the brain.' In: *Journal of cognitive neuroscience* 19.5, pp. 893–906 (cit. on p. 11).

Grosche, Peter and Meinard Müller (2009). 'A Mid-Level Representation for Capturing Dominant Tempo and Pulse Information in Music Recordings.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)*. Kobe, Japan, pp. 189–194 (cit. on p. 20).

Hadsell, Raia, Sumit Chopra, and Yann LeCun (2006). 'Dimensionality reduction by learning an invariant mapping.' In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 2. IEEE, pp. 1735–1742 (cit. on p. 104).

Hainsworth, Stephen Webley (2004). 'Techniques for the Automated Analysis of Musical Audio.' PhD thesis. UK: University of Cambridge (cit. on pp. 37, 41).

He, Kaiming et al. (2016). 'Identity mappings in deep residual networks.' In: *European conference on computer vision*. Springer, pp. 630–645 (cit. on p. 108).

He, Xinwei et al. (2018). 'Triplet-center loss for multi-view 3d object retrieval.' In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1945–1954 (cit. on p. 108).

Henaff, Mikael et al. (2011). 'Unsupervised learning of sparse features for scalable audio classification.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)*. Vol. 11. 445, p. 2011 (cit. on p. 33).

Hockman, Jason, Matthew EP Davies, and Ichiro Fujinaga (2012). 'One in the Jungle: Downbeat Detection in Hardcore, Jungle, and Drum and Bass.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)*. Porto, Portugal, pp. 169–174 (cit. on pp. 34, 46).

Holzapfel, André and Yannis Stylianou (2011). 'Scale transform in rhythmic similarity of music.' In: *Audio, Speech and Language Processing, IEEE Transactions on* 19.1, pp. 176–185 (cit. on pp. 20, 21, 40, 76).

Holzapfel, Andre et al. (2012). 'Selective sampling for beat tracking evaluation.' In: *Audio, Speech and Language Processing, IEEE Transactions on* 20.9, pp. 2539–2548 (cit. on pp. 14, 21, 44, 51).

Honing, Henkjan (2001). 'From time to time: The representation of timing and tempo.' In: *Computer Music Journal* 25.3, pp. 50–61 (cit. on p. 15).

Honing, Henkjan and W Bas De Haas (2008). 'Swing once more: Relating timing and tempo in expert jazz drumming.' In: *Music Perception* 25.5, pp. 471–476 (cit. on p. 15).

Honingh, Aline et al. (2015). 'Perception of timbre and rhythm similarity in electronic dance music.' In: *Journal of New Music Research* 44.4, pp. 373–390 (cit. on pp. 11, 34).

Hörschläger, Florian et al. (2015). 'Addressing tempo estimation octave errors in electronic music by incorporating style information extracted from Wikipedia.' In: *Proc 12th SMC* (cit. on p. 35).

Iloga, Sylvain, Olivier Romain, and Maurice Tchuenté (2018). 'A sequential pattern mining approach to design taxonomies for hierarchical music genre recognition.' In: *Pattern Analysis and Applications* 21.2, pp. 363–380 (cit. on p. 33).

Ioffe, Sergey and Christian Szegedy (2015). 'Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.' In: *Proc. of ICML (International Conference on Machine Learning)*, pp. 448–456 (cit. on pp. 30, 63).

Jones, Mari R and Marilyn Boltz (1989). 'Dynamic attending and responses to time.' In: *Psychological review* 96.3, p. 459 (cit. on p. 13).

Kingma, Diederik P and Jimmy Ba (2014). 'Adam: A method for stochastic optimization.' In: *arXiv preprint arXiv:1412.6980* (cit. on pp. 31, 64).

Klapuri, Anssi P, Antti J Eronen, and Jaakko T Astola (2006). 'Analysis of the meter of acoustic musical signals.' In: *Audio, Speech and Language Processing, IEEE Transactions on* 14.1, pp. 342–355 (cit. on p. 20).

Klapuri, Anssi (1999). 'Sound onset detection by applying psychoacoustic knowledge.' In: *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*. Vol. 6. IEEE, pp. 3089–3092 (cit. on pp. 18, 20).

Knees, Peter et al. (2015). 'Two Data Sets for Tempo Estimation and Key Detection in Electronic Dance Music Annotated from User Corrections.' In: *Proc. of*

ISMIR (International Society for Music Information Retrieval). Malaga, Spain (cit. on pp. 35, 40, 50).

Kong, Qiuqiang, Xiaohui Feng, and Yanxiong Li (2014). 'Music genre classification using convolutional neural network.' In: *Proc. of Int. Society for Music Information Retrieval Conference (ISMIR)* (cit. on p. 33).

Kong, Qiuqiang et al. (2019). 'Weakly labelled audioset tagging with attention neural networks.' In: *Audio, Speech, and Language Processing, IEEE/ACM Transactions on* 27.11, pp. 1791–1802 (cit. on pp. 65, 66).

Lambrou, Tryphon et al. (1998). 'Classification of audio signals using statistical features on time and wavelet transform domains.' In: *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*. Vol. 6. IEEE, pp. 3621–3624 (cit. on p. 21).

LeCun, Yann, Yoshua Bengio, et al. (1995). 'Convolutional networks for images, speech, and time series.' In: *The handbook of brain theory and neural networks* 3361.10, p. 1995 (cit. on p. 28).

Lee, Jongpil et al. (2020). 'Metric Learning vs Classification for Disentangled Music Representation Learning.' In: (cit. on p. 117).

Lerdhal, F and Ray Jackendoff (1983). 'A generative theory of tonal grammar.' In: *Cambridge, MA* (cit. on pp. 11–13).

Levy, Mark (2011). 'Improving Perceptual Tempo Estimation with Crowd-Sourced Annotations.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)*. Miami, Florida, USA (cit. on p. 32).

Li, Tao, Mitsunori Ogihara, and Qi Li (2003). 'A comparative study on content-based music genre classification.' In: *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pp. 282–289 (cit. on p. 33).

Lloyd, Stuart (1982). 'Least squares quantization in PCM.' In: *IEEE transactions on information theory* 28.2, pp. 129–137 (cit. on p. 102).

Lowe, David G (1995). 'Similarity metric learning for a variable-kernel classifier.' In: *Neural computation* 7.1, pp. 72–85 (cit. on p. 102).

Maaten, Laurens van der and Geoffrey Hinton (2008). 'Visualizing data using t-SNE.' In: *Journal of machine learning research* 9.Nov, pp. 2579–2605 (cit. on p. 110).

Maher, Robert C and James W Beauchamp (1994). 'Fundamental frequency estimation of musical signals using a two-way mismatch procedure.' In: *JASA (Journal of the Acoustical Society of America)* 95.4, pp. 2254–2263 (cit. on p. 54).

Marchand, Ugo, Quentin Fresnel, and Geoffroy Peeters (Oct. 2015). *GTZAN-Rhythm: Extending the GTZAN Test-Set with Beat, Downbeat and Swing Annotations*. Late-Breaking Demo Session of the 16th International Society for Music Informa-

tion Retrieval Conference, 2015. URL: https://hal.archives-ouvertes.fr/hal-01252607 (cit. on p. 41).

Marchand, Ugo and Geoffroy Peeters (2014). 'The Modulation Scale Spectrum and its Application to Rhythm-Content Description.' In: *Proc. of DAFx (International Conference on Digital Audio Effects)*. Erlangen, Germany, pp. 167–172 (cit. on pp. 21, 82).

— (2016a). 'Scale and shift invariant time/frequency representation using auditory statistics: Application to rhythm description.' In: *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, pp. 1–6 (cit. on pp. 21, 36, 39, 75, 76, 80, 119).

— (Aug. 2016b). 'The Extended Ballroom Dataset.' In: *Late-Breaking/Demo Session of ISMIR (International Society for Music Information Retrieval)*. Late-Breaking Demo Session of the 17th International Society for Music Information Retrieval Conf.. 2016. New York, USA. URL: https://hal.archives-ouvertes.fr/hal-01374567 (cit. on pp. 21, 39).

— (2016c). 'The extended ballroom dataset.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)* (cit. on p. 82).

McAuley, R and T Quatieri (1986). 'Speech Analysis/Synthesis Based on a Sinusoidal Representation.' In: *Acoustics, Speech and Signal Processing, IEEE Transactions on* 34, pp. 744–754 (cit. on p. 54).

McFee, Brian et al. (2015). 'librosa: Audio and music signal analysis in python.' In: *Proceedings of the 14th python in science conference*, pp. 18–25 (cit. on p. 58).

McKinney, Martin F and Dirk Moelants (2004). 'Extracting the perceptual tempo from music.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)* (cit. on p. 14).

— (2006). 'Ambiguity in tempo perception: What draws listeners to different metrical levels?' In: *Music Perception* 24.2, pp. 155–166 (cit. on p. 14).

McLeod, Kembrew (2001). 'Genres, subgenres, sub-subgenres and more: Musical and social differentiation within electronic/dance music communities.' In: *Journal of Popular Music Studies* 13.1, pp. 59–75 (cit. on p. 6).

Moelants, Dirk and Martin McKinney (2004). 'Tempo perception and musical content: What makes a piece fast, slow or temporally ambiguous.' In: *Proceedings of the 8th International Conference on Music Perception and Cognition*, pp. 558–562 (cit. on p. 14).

Müller, Meinard (2015). *Fundamentals of music processing: Audio, analysis, algorithms, applications*. Springer (cit. on p. 22).

Muller, Meinard et al. (2011). 'Signal processing for music analysis.' In: *IEEE Journal of selected topics in signal processing* 5.6, pp. 1088–1110 (cit. on p. 16).

Nair, Vinod and Geoffrey E Hinton (2010). 'Rectified linear units improve restricted boltzmann machines.' In: *Proc. of ICML (International Conference on Machine Learning)*. Haifa, Israel, pp. 807–814 (cit. on pp. 27, 63).

Nanni, Loris et al. (2016). 'Combining visual and acoustic features for music genre classification.' In: *Expert Systems with Applications* 45, pp. 108–117 (cit. on p. 33).

Nanni, Loris et al. (2018). 'Ensemble of deep learning, visual and acoustic features for music genre classification.' In: *Journal of New Music Research* 47.4, pp. 383–397 (cit. on p. 33).

Ness, Steven R et al. (2009). 'Improving automatic music tag annotation using stacked generalization of probabilistic svm outputs.' In: *Proceedings of the 17th ACM international conference on Multimedia*, pp. 705–708 (cit. on p. 33).

Oh Song, Hyun et al. (2016). 'Deep metric learning via lifted structured feature embedding.' In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4004–4012 (cit. on p. 108).

Oord, Aaron Van den, Sander Dieleman, and Benjamin Schrauwen (2013). 'Deep content-based music recommendation.' In: *Advances in neural information processing systems*, pp. 2643–2651 (cit. on p. 96).

Pálmason, Haukur et al. (2017). 'Music genre classification revisited: An in-depth examination guided by music experts.' In: *International Symposium on Computer Music Multidisciplinary Research*. Springer, pp. 49–62 (cit. on p. 33).

Panteli, Maria, Niels Bogaards, Aline K Honingh, et al. (2014). 'Modeling Rhythm Similarity for Electronic Dance Music.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)*. Taipei, Taiwan, pp. 537–542 (cit. on p. 34).

Panteli, Maria et al. (2017). 'A model for rhythm and timbre similarity in electronic dance music.' In: *Musicae Scientiae* 21.3, pp. 338–361 (cit. on p. 34).

Paulus, Jouni and Anssi Klapuri (2003). 'Model-based event labeling in the transcription of percussive audio signals.' In: *Proc. of DAFx (International Conference on Digital Audio Effects)*. Citeseer. London, UK, pp. 73–77 (cit. on p. 20).

Peeters, Geoffroy (2004). *A large set of audio features for sound description (similarity and classification) in the CUIDADO project*. Cuidado Project Report. Ircam (cit. on p. 54).

— (2006). 'Template-based estimation of time-varying tempo.' In: *Advances in Signal Processing, EURASIP Journal on* 2007.1, p. 067215 (cit. on pp. 20, 37, 55, 58, 65).

— (2010). 'Template-based estimation of tempo: using unsupervised or supervised learning to create better spectral templates.' In: *Proc. of DAFx (International Conference on Digital Audio Effects)*. Graz, Austria, pp. 209–212 (cit. on pp. 55, 58).

— (2011). 'Spectral and temporal periodicity representations of rhythm for the automatic classification of music audio signal.' In: *Audio, Speech and Language*

Processing, IEEE Transactions on 19.5, pp. 1242–1252 (cit. on pp. 16, 20, 36, 55, 56, 58–60, 119).

Peeters, Geoffroy and Joachim Flocon-Cholet (2012). 'Perceptual tempo estimation using GMM-regression.' In: *Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies*. ACM, pp. 45–50 (cit. on pp. 14, 32, 38).

Peeters, Geoffroy and Ugo Marchand (2013). 'Predicting agreement and disagreement in the perception of tempo.' In: *International Symposium on Computer Music Multidisciplinary Research*. Springer, pp. 313–329 (cit. on p. 14).

Peeters, Geoffroy et al. (2011). 'The timbre toolbox: Extracting audio descriptors from musical signals.' In: *JASA (Journal of the Acoustical Society of America)* 130.5, pp. 2902–2916 (cit. on p. 96).

Percival, Graham and George Tzanetakis (2014). 'Streamlined tempo estimation based on autocorrelation and cross-correlation with pulses.' In: *Audio, Speech, and Language Processing, IEEE/ACM Transactions on* 22.12, pp. 1765–1776 (cit. on pp. 20, 32, 37, 65).

Phung, Van Hiep, Eun Joo Rhee, et al. (2019). 'A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets.' In: *Applied Sciences* 9.21, p. 4500 (cit. on p. 30).

Pons, Jordi, Thomas Lidy, and Xavier Serra (2016). 'Experimenting with musically motivated convolutional neural networks.' In: *2016 14th International Workshop on Content-Based Multimedia Indexing (CBMI)*. IEEE, pp. 1–6 (cit. on p. 33).

Pons, Jordi and Xavier Serra (2017). 'Designing efficient architectures for modeling temporal features with convolutional neural networks.' In: *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*. IEEE, pp. 2472–2476 (cit. on pp. 33, 63, 96).

Pons, Jordi et al. (2017a). 'End-to-end learning for music audio tagging at scale.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)* (cit. on p. 33).

Pons, Jordi et al. (2017b). 'Timbre analysis of music audio signals with convolutional neural networks.' In: *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, pp. 2744–2748 (cit. on pp. 33, 95–97).

Prechelt, Lutz (1998). 'Early stopping-but when?' In: *Neural Networks: Tricks of the trade*. Springer, pp. 55–69 (cit. on p. 31).

Prétet, Laure, Gaël Richard, and Geoffroy Peeters (2020). 'Learning to rank music tracks using triplet loss.' In: *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*. IEEE. Barcelona, Spain, pp. 511–515 (cit. on p. 108).

Pye, David (2000). 'Content-based methods for the management of digital music.' In: *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*. Vol. 4. IEEE, pp. 2437–2440 (cit. on p. 21).

Rabiner, Lawrence R and Bernard Gold (1975). 'Theory and application of digital signal processing.' In: *tads* (cit. on p. 16).

Raffel, Colin (2016). 'Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching.' PhD thesis. Columbia University (cit. on p. 41).

Ramírez, Jaime and M Julia Flores (2019). 'Machine learning for music genre: multifaceted review and experimentation with audioset.' In: *Journal of Intelligent Information Systems*, pp. 1–31 (cit. on pp. 16, 22, 33).

Reynolds, Simon (2013). *Energy flash: A journey through rave music and dance culture*. Faber & Faber (cit. on p. 6).

Rocha, Bruno, Niels Bogaards, and Aline Honingh (2013). 'Segmentation and timbre similarity in electronic dance music.' In: *IEEE International Conference on Systems, Man and Cybernetics* (cit. on p. 34).

Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams (1986). 'Learning representations by back-propagating errors.' In: *nature* 323.6088, pp. 533–536 (cit. on p. 26).

Scheirer, Eric D (1998). 'Tempo and beat analysis of acoustic musical signals.' In: *JASA (Journal of the Acoustical Society of America)* 103.1, pp. 588–601 (cit. on pp. 19, 20).

Scheirer, Eric David (2000). 'Music-listening systems.' PhD thesis. Massachusetts Institute of Technology (cit. on p. 14).

Schindler, Alexander, Thomas Lidy, and Andreas Rauber (2016). 'Comparing Shallow versus Deep Neural Network Architectures for Automatic Music Genre Classification.' In: *FMT*, pp. 17–21 (cit. on p. 33).

Schlüter, Jan and Sebastian Böck (2014). 'Improved musical onset detection with convolutional neural networks.' In: *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*. IEEE. Florence, Italy, pp. 6979–6983 (cit. on p. 96).

Schreiber, Hendrik and M Müller (2018a). 'A crowdsourced experiment for tempo estimation of electronic dance music.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)*. Paris, France (cit. on pp. 35, 40, 50).

— (2018b). 'A single-step approach to musical tempo estimation using a convolutional neural network.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)*. Paris, France (cit. on pp. 32, 39, 42, 60, 62, 67, 68, 71, 85, 87).

Schreiber, Hendrik and Meinard Müller (2017). 'A Post-Processing Procedure for Improving Music Tempo Estimates Using Supervised Learning.' In: *Proc. of*

ISMIR (International Society for Music Information Retrieval). Suzhou, China (cit. on pp. 41, 44, 68).

Schreiber, Hendrik and Meinard Müller (2019). 'Musical Tempo and Key Estimation using Convolutional Neural Networks with Directional Filters.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)* (cit. on p. 32).

Schreiber, Hendrik, Julián Urbano, and Meinard Müller (2020). 'Music Tempo Estimation: Are We Done Yet?' In: *Proc. of ISMIR (International Society for Music Information Retrieval)* 3.1 (cit. on p. 52).

Schroff, Florian, Dmitry Kalenichenko, and James Philbin (2015). 'Facenet: A unified embedding for face recognition and clustering.' In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823 (cit. on pp. 107, 108, 110).

Schubotz, Ricarda I, Angela D Friederici, and D Yves Von Cramon (2000). 'Time perception and motor timing: a common cortical and subcortical basis revealed by fMRI.' In: *Neuroimage* 11.1, pp. 1–12 (cit. on p. 11).

Senac, Christine et al. (2017). 'Music feature maps with convolutional neural networks for music genre classification.' In: *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing*, pp. 1–5 (cit. on p. 33).

Seppanen, Jarno (2001). 'Tatum grid analysis of musical signals.' In: *Proceedings of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (Cat. No. 01TH8575)*. IEEE, pp. 131–134 (cit. on p. 21).

Seppänen, Jarno, Antti J Eronen, and Jarmo Hiipakka (2006). 'Joint Beat & Tatum Tracking from Music Signals.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)*, pp. 23–28 (cit. on p. 20).

Serra, Xavier and Julius Smith (1990). 'Spectral Modeling Synthesis: A Sound Analysis/Synthesis System Based on a Deterministic Plus Stochastic Decomposition.' In: *Computer Music Journal* 14.4, pp. 12–24. ISSN: 01489267, 15315169. URL: http://www.jstor.org/stable/3680788 (cit. on p. 54).

Seyerlehner, Klaus, Gerhard Widmer, and Dominik Schnitzer (2007). 'From Rhythm Patterns to Perceived Tempo.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)*. Vienna, Austria (cit. on p. 31).

Shapiro, Peter and Iara Lee (2000). *Modulations: a history of electronic music: throbbing words on sound*. Caipirinha Productions (cit. on p. 6).

Silla Jr, Carlos N, Alessandro L Koerich, and Celso AA Kaestner (2010). 'Improving automatic music genre classification with hybrid content-based feature vectors.' In: *Proceedings of the 2010 ACM Symposium on Applied Computing*, pp. 1702–1707 (cit. on p. 33).

Simo-Serra, Edgar et al. (2015). 'Discriminative learning of deep convolutional feature point descriptors.' In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 118–126 (cit. on pp. 105, 106).

Simonyan, Karen and Andrew Zisserman (2015). 'Very deep convolutional networks for large-scale image recognition.' In: *International Conference on Learning Representation* (cit. on p. 97).

Sohn, Kihyuk (2016). 'Improved deep metric learning with multi-class n-pair loss objective.' In: *Advances in neural information processing systems*, pp. 1857–1865 (cit. on p. 108).

Srivastava, Nitish et al. (2014). 'Dropout: a simple way to prevent neural networks from overfitting.' In: *Journal of Machine Learning Research* 15.1, pp. 1929–1958 (cit. on pp. 30, 63).

Stevens, Stanley Smith, John Volkmann, and Edwin B Newman (1937). 'A scale for the measurement of the psychological magnitude pitch.' In: *JASA (Journal of the Acoustical Society of America)* 8.3, pp. 185–190 (cit. on p. 95).

Thornton, Sarah (1996). *Club Cultures: Music, Media, and Subcultural Capital. Wesleyan*. Paperback (cit. on p. 5).

Trabelsi, Chiheb et al. (2017). 'Deep complex networks.' In: *arXiv preprint arXiv:1705.09792* (cit. on pp. 83, 84).

Tzanetakis, George and Perry Cook (2000). 'Marsyas: A framework for audio analysis.' In: *Organised sound* 4.3, pp. 169–175 (cit. on p. 21).

— (2002). 'Musical genre classification of audio signals.' In: *Audio, Speech and Language Processing, IEEE Transactions on* 10.5, pp. 293–302 (cit. on pp. 15, 20, 21, 32, 33, 41).

Ullrich, Karen, Jan Schlüter, and Thomas Grill (2014). 'Boundary Detection in Music Structure Analysis using Convolutional Neural Networks.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)*. Taipei, Taiwan, pp. 417–422 (cit. on p. 96).

Vitos, Botond (2014). 'Along the lines of the Roland TB-303: Three perversions of acid techno.' In: *Dancecult: Journal of Electronic Dance Music Culture* 6.1, p. 10 (cit. on p. 6).

Wakefield, Gregory H (1999). 'Mathematical representation of joint time-chroma distributions.' In: *Advanced Signal Processing Algorithms, Architectures, and Implementations IX*. Vol. 3807. International Society for Optics and Photonics, pp. 637–645 (cit. on p. 22).

Ward, Ed, Geoffrey Stokes, and Ken Tucker (1986). *Rock of ages: The Rolling Stone history of rock & roll*. Prentice Hall (cit. on p. 4).

Weinberger, Kilian Q, John Blitzer, and Lawrence K Saul (2006). 'Distance metric learning for large margin nearest neighbor classification.' In: *Advances in neural information processing systems*, pp. 1473–1480 (cit. on p. 105).

Wu, Fu-Hai Frank and Jyh-Shing Roger Jang (2014). 'A supervised learning method for tempo estimation of musical audio.' In: *22nd Mediterranean Conference on Control and Automation*. IEEE, pp. 599–604 (cit. on p. 21).

Xiao, Linxing et al. (2008). 'Using Statistic Model to Capture the Association between Timbre and Perceived Tempo.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)*. Philadelphia, PA, USA (cit. on pp. 20, 32).

Xing, Eric P et al. (2003). 'Distance metric learning with application to clustering with side-information.' In: *Advances in neural information processing systems*, pp. 521–528 (cit. on p. 103).

Yang, Yongxin and Timothy M Hospedales (2016). 'Trace norm regularised deep multi-task learning.' In: *arXiv preprint arXiv:1606.04038* (cit. on p. 90).

Zapata, Jose Ricardo et al. (2012). 'Assigning a confidence threshold on automatic beat annotation in large datasets.' In: *Proc. of ISMIR (International Society for Music Information Retrieval)*. Porto, Portugal, pp. 157–162 (cit. on p. 14).

Zapata, Jose and Emilia Gómez (2011). 'Comparative evaluation and combination of audio tempo estimation approaches.' In: *Audio Engineering Society Conference: 42nd International Conference: Semantic Audio*. Audio Engineering Society (cit. on p. 16).

Zhang, Weibin et al. (2016). 'Improved Music Genre Classification with Convolutional Neural Networks.' In: *Interspeech*, pp. 3304–3308 (cit. on p. 33).

Zils, Aymeric and François Pachet (2001). 'Musical mosaicing.' In: *Proc. of DAFx (International Conference on Digital Audio Effects)*. Vol. 2. Limerick, Ireland, p. 135 (cit. on p. 1).