



**HAL**  
open science

# Music encoding and deep learning for music transcription and classification based on visually represented audio features

Charbel El Achkar

► **To cite this version:**

Charbel El Achkar. Music encoding and deep learning for music transcription and classification based on visually represented audio features. Computer Vision and Pattern Recognition [cs.CV]. Université Bourgogne Franche-Comté, 2023. English. NNT : 2023UBFCD054 . tel-04483299

**HAL Id: tel-04483299**

**<https://theses.hal.science/tel-04483299>**

Submitted on 29 Feb 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**THÈSE DE DOCTORAT**

**DE L'ÉTABLISSEMENT UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ**

**PRÉPARÉE À L'UNIVERSITÉ DE FRANCHE-COMTÉ**

École doctorale n°37

Sciences Pour l'Ingénieur et Microtechniques

Doctorat d'Informatique

par

**CHARBEL EL ACHKAR**

**Music Encoding and Deep Learning for Music Transcription and  
Classification based on Visually Represented Audio Features**

Encodage de musique et apprentissage en profondeur pour la transcription et la  
classification de la musique basées sur des caractéristiques audio représentées  
visuellement

Thèse présentée et soutenue à Belfort, le 15/12/2023

Composition du Jury :

PROF GHEDIRA GUEGAN CHIRINE	Université Jean Moulin Lyon 3	Rapporteur
PROF VERNIER FLAVIEN	Université Savoie Mont Blanc	Rapporteur
PROF WEMMERT CÉDRIC	Université de Strasbourg	Examineur
PROF COUTURIER RAPHAËL	Université de Franche-Comté	Examineur
PROF MAKHOUL ABDALLAH	Université de Franche-Comté	Directeur de thèse
DR ATÉCHIAN TALAR	Université Antonine, Liban	Codirecteur de thèse



# ABSTRACT

## Music Encoding and Deep Learning for Music Transcription and Classification based on Visually Represented Audio Features

Charbel El Achkar  
University of Bourgogne Franche Comté, 2023

Supervisors: Raphaël Couturier, Abdallah Makhoul and Talar Atéchian

In the last decade, new music scores following the occidental genre have been constantly composed by musicians and many of them are encoded using XML-based formats for analysis purposes. As for the oriental genre, it lacks the support of XML-based formats due to the lower interest of digitisation communities. Thus, the inability to process oriental notations and failure to accurately encode the latter genre. The fast growth of deep learning encouraged both developers and musicians to discover its benefits for the music domain. This thesis focuses on music genre classification and automatic music transcription among many deep learning-related music applications. Consequently, our efforts for encoding Eastern music and leveraging deep learning for music applications are the following:

An ontology named MusicPatternOWL is proposed to structure the knowledge extraction process of a music pattern analysis algorithm for encoding Eastern music scores.

Additionally, the MEI2JSON converter is proposed for transforming MEI-encoded music scores into JSON format, catering to artificial intelligence pre-processing requirements. Comparative analysis with two existing converters reveals that MEI2JSON outperforms the combined approach in terms of data quality and storage efficiency.

In the context of deep learning's impact on music streaming services, we propose a pre-processing method for generating Short Time Fourier Transform (STFT) spectrograms and enhancing a CNN-based music genre classifier. Our approach is benchmarked against state-of-the-art classifiers, demonstrating superior accuracy scores.

Furthermore, we introduce two novel networks, TabInception and Inception Transformer (InT), for guitar tablature transcription. Evaluation against state-of-the-art networks in the field using Multi-pitch and Tablature metrics reveals that both proposed solutions outperform benchmark networks, with InT demonstrating the highest scores in various metrics, making it the preferred choice.

**KEYWORDS:** Ontology, Pattern Analysis, Music Information Retrieval, Music Score Converter, Eastern Music, Music Genre Classification, STFT Spectrogram, CNN, Transformers, Music Recommendation Systems, Deep Learning, Computer Vision, Automatic Music Transcription, Guitar Tablature Transcription, Constant-Q Transform.

# RÉSUMÉ

Encodage de musique et apprentissage en profondeur pour la transcription et la classification de la musique basées sur des caractéristiques audio représentées visuellement

Charbel El Achkar  
Université de Bourgogne Franche Comté, 2023

Encadrants: Raphaël Couturier, Abdallah Makhoul et Talar Atéchian

Au cours de la décennie écoulée, les nouvelles partitions de genre occidental ont été constamment composées par des musiciens, et plusieurs d'entre elles sont encodées à l'aide des formats XML pour les besoins d'analyses. Quant aux partitions de genre oriental, elles ne peuvent pas être encodées à l'aide des formats XML à cause d'un manque d'intérêt de la part de la communauté de numérisation, ainsi l'incapacité à traiter les notations musicales et à coder ce genre avec précision. Le développement rapide de l'apprentissage profond (deep learning) encourage les développeurs et les musiciens à découvrir ses avantages dans le domaine musical. Cette thèse porte sur la classification des genres musicaux et la transcription automatique de la musique, parmi les nombreuses applications musicales liées à l'apprentissage profond. Par conséquent, nous essayons d'encoder de la musique orientale et tirer parti de l'apprentissage profond pour les applications de musique à travers :

Une ontologie, MusicPatternOWL, est proposée pour structurer le processus d'extraction des connaissances d'algorithme d'analyse des séquences musicales pour encoder des partitions orientales.

De plus, le convertisseur MEI2JSON est proposé pour transformer les partitions musicales codées MEI au format JSON, répondant ainsi aux exigences de prétraitement de l'intelligence artificielle. Une analyse comparative avec deux convertisseurs existants révèle que MEI2JSON surpasse l'approche combinée en termes de qualité des données et d'efficacité du stockage.

Dans le contexte de l'impact de l'apprentissage profond sur les services de streaming mu-

sical, nous proposons une méthode de pré-traitement pour générer des spectrogrammes de transformée de Fourier à court terme (STFT) et améliorer un classificateur de genre musical basé sur CNN. Notre approche est comparée à des classificateurs de pointe, démontrant des scores de précision supérieurs.

De plus, nous introduisons deux nouveaux réseaux, TabInception et Inception Transformer (InT), pour la transcription de tablatures de guitare. L'évaluation par rapport aux réseaux de pointe dans le domaine à l'aide des métriques Multi-pitch et Tablature révèle que les deux solutions proposées surpassent les réseaux de référence, InT démontrant les scores les plus élevés dans diverses métriques, ce qui en fait le choix préféré.

**Mots clés:** Ontologie, Analyse de modèles, Récupération d'informations musicales, Convertisseur de partition musicale, Musique orientale, Classification des genres musicaux, Spectrogramme STFT, CNN, Transformateurs, Systèmes de recommandation musicale, L'apprentissage en profondeur, Vision par ordinateur, Transcription automatique de la musique, Transcription des tablatures de guitare, Transformation Constant-Q.

# ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Professor Chirine Guegan Ghedira, Professor Flavien Vernier, and Professor Cédric Wemmert, esteemed members of the Jury, for dedicating their time and expertise to read my thesis, attend my presentation, and evaluate my work. Your valuable insights and feedback are deeply appreciated.

I extend my heartfelt appreciation to my PhD supervisors, Professor Abdallah Makhoul and Dr. Talar Atéchian, for their support during my journey toward earning my doctoral degree. I am extremely thankful to Talar for her guidance and belief in my abilities since my Master's studies at Antonine University. I'm also appreciative of Abdallah for his positive attitude and determined support throughout my doctoral journey, especially for allowing me to choose my thesis topic without any objections.

In addition to my supervisors, I want to express an honourable acknowledgement to Professor Raphaël Couturier. His extensive expertise, mentorship, and commitment to my academic and personal growth have played a vital role in shaping my research and helping me overcome the challenges I encountered along the way. This thesis wouldn't have been achievable without his guidance.

A special thanks to Professor Nidaa Abou Mrad for his support and guidance in understanding the musical aspects of this study, especially the modal semiotic theory and the encoding process of traditional modal monodies. Also, I'm grateful to Mr. Joe Azar for his daily assistance in explaining musical theories and validating the process of transforming paper-based musical compositions into digital scores.

I would also like to thank the professors and colleagues of the Department of Informatics and Complex Systems (DISC in French) in the FEMTO-ST laboratory for their support and positive working atmosphere.

I want to thank the "Agence Universitaire à la Francophonie" (AUF) - Middle East branch for their funding and belief in the value of this research study. Additionally, I'd like to thank "inmind.ai", especially Professor Bechara Al Bouna (CEO and Founder), for allowing me to work in the company while pursuing divergent research goals.

On a personal note, I extend my gratitude to all the members of my big family, especially my parents and sisters. I also want to thank my cousin, Dr. Wissam Bou Nader, for his support during my PhD journey in France.





# CONTENTS

<b>I</b>	<b>Dissertation introduction</b>	<b>5</b>
<b>1</b>	<b>Introduction</b>	<b>7</b>
1.1	Music Encoding . . . . .	7
1.2	Deep Learning for Music Applications . . . . .	8
1.3	Main Contributions of this Dissertation . . . . .	9
1.4	Dissertation Outline . . . . .	10
<b>II</b>	<b>Related Work</b>	<b>11</b>
<b>2</b>	<b>Related Work</b>	<b>13</b>
2.1	Introduction . . . . .	13
2.2	Knowledge extraction and representation of music scores . . . . .	14
2.3	Music scores organization and conversion . . . . .	16
2.4	From music scores to visually represented audio features . . . . .	19
2.5	Computer Vision for music applications . . . . .	28
2.6	Music genre classification using computer vision . . . . .	33
2.7	Guitar tablature estimation using computer vision . . . . .	34
2.8	Music Datasets for experimental studies . . . . .	36
2.9	Conclusion . . . . .	39
<b>III</b>	<b>Knowledge extraction and format conversion of music scores</b>	<b>41</b>
<b>3</b>	<b>Supporting Music Pattern Retrieval and Analysis</b>	<b>43</b>
3.1	Introduction . . . . .	43
3.2	MusicPatternOWL . . . . .	44

3.2.1	Goals and Restrictions . . . . .	44
3.2.2	Structural Aspects . . . . .	45
3.3	PROOF OF CONCEPT . . . . .	47
3.3.1	Score Analyses . . . . .	47
3.3.2	Theme Queries . . . . .	48
3.4	Conclusion . . . . .	50
<b>4</b>	<b>Music score pre-processing and conversion</b>	<b>51</b>
4.1	Introduction . . . . .	51
4.2	The MEI2JSON Converter . . . . .	53
4.2.1	Motivation . . . . .	53
4.2.2	MEI2JSON Components . . . . .	53
4.2.2.1	The MEI2XML Component . . . . .	54
4.2.2.2	The XML2RDF Component . . . . .	57
4.2.2.3	The RDF2JSON Component . . . . .	57
4.3	Implementation . . . . .	58
4.3.1	MEI2JSON Process . . . . .	60
4.3.2	<i>Meico + MusicJSON</i> Process . . . . .	62
4.4	Experiments . . . . .	62
4.4.1	Dataset . . . . .	63
4.4.2	Performance Analysis . . . . .	63
4.4.2.1	Time Complexity . . . . .	64
4.4.2.2	Space Complexity . . . . .	65
4.4.3	Data Quality Assessment . . . . .	67
4.5	Storage assessment . . . . .	70
4.6	Conclusion . . . . .	71
<b>IV</b>	<b>Computer Vision in the music industry</b>	<b>73</b>
<b>5</b>	<b>Music Genre Classification using Computer Vision</b>	<b>75</b>
5.1	Introduction . . . . .	75

5.2	Proposed Approach . . . . .	76
5.2.1	Preprocessing . . . . .	76
5.2.2	Network Contribution . . . . .	77
5.3	Experimental Evaluation . . . . .	79
5.3.1	Hyperparameters and Training Details . . . . .	79
5.3.2	Testing Results . . . . .	80
5.4	Discussion . . . . .	81
5.4.1	Librosa vs SoX . . . . .	81
5.4.2	Genre by Genre precision percentages of the proposed approach . . . . .	84
5.5	Conclusion . . . . .	85
<b>6</b>	<b>Automatic Music Transcription using computer vision</b>	<b>87</b>
6.1	Introduction . . . . .	87
6.2	Data Selection And Preparation . . . . .	89
6.2.1	The GuitarSet Dataset . . . . .	89
6.2.2	Data Preprocessing . . . . .	89
6.3	Proposed Networks . . . . .	91
6.3.1	The TabInception Network . . . . .	91
6.3.2	The Inception Transformer Network . . . . .	93
6.4	Experiments . . . . .	94
6.5	Conclusion . . . . .	98
<b>V</b>	<b>Conclusion &amp; Perspectives</b>	<b>99</b>
<b>7</b>	<b>Conclusion &amp; Perspectives</b>	<b>101</b>
7.1	Conclusion . . . . .	101
7.2	Perspectives . . . . .	103



# LIST OF ABBREVIATIONS

<b>ADAM</b>	. . . . .	Adaptive Moment Estimation
<b>ADC</b>	. . . . .	Analogue to Digital Converter
<b>AI</b>	. . . . .	Artificial Intelligence
<b>AMT</b>	. . . . .	Automatic Music Transcription
<b>BBNN</b>	. . . . .	Bottom-up Broadcast Neural Network
<b>BM</b>	. . . . .	Broadcast Module
<b>BN</b>	. . . . .	Batch Normalization
<b>C-RNN</b>	. . . . .	Convolutional Recurrent Neural Networks
<b>CMC</b>	. . . . .	Computer Music Cloud
<b>CNN</b>	. . . . .	Convolutional Neural Network
<b>CQT</b>	. . . . .	Constant-Q Transform
<b>CV</b>	. . . . .	Computer Vision
<b>dB</b>	. . . . .	decibels
<b>DCNN</b>	. . . . .	Deep Convolutional Neural Networks
<b>DCT</b>	. . . . .	Discrete Cosine Transform
<b>DNN</b>	. . . . .	Deep Neural Networks
<b>DenseNet</b>	. . . . .	Densely Connected Convolutional Networks
<b>DSP</b>	. . . . .	Digital Signal Processing
<b>EEG</b>	. . . . .	Electroencephalography
<b>F0</b>	. . . . .	Fundamental Frequency
<b>FFT</b>	. . . . .	Fast Fourier Transform
<b>FMA</b>	. . . . .	Free Music Archive
<b>FPN</b>	. . . . .	Feature Pyramid Network
<b>GAN</b>	. . . . .	Generative Adversarial Network
<b>GIF</b>	. . . . .	Graphics Interchange Format
<b>GPU</b>	. . . . .	Graphics Processing Unit

<b>Hz</b>	Hertz
<b>InT</b>	Inception Transformer
<b>IRMAS</b>	Instrument Recognition in Musical Audio Signals
<b>JAMS</b>	JSON Annotated Music Specification
<b>JSON</b>	JavaScript Object Notation
<b>LSTM</b>	Long Short-Term Memory Networks
<b>MEI</b>	Music Encoding Initiative
<b>MF</b>	Multi-Pitch F-measure
<b>MFCCs</b>	Mel-Frequency Cepstral Coefficients
<b>MIDI</b>	Musical Instrument Digital Interface
<b>MIR</b>	Music Information Retrieval
<b>MLP</b>	Multilayer Perceptron
<b>MM</b>	Modal Monodies
<b>MNR</b>	Metasyllabic Nuclear Reduction
<b>MPO</b>	MusicPatternOWL Ontology
<b>MRMR</b>	Morphophonological Rhythmic and Melodic Rewriting
<b>MP</b>	Multi-Pitch Precision
<b>MR</b>	Mutli-Pitch Recall
<b>MSD</b>	Million Song Dataset
<b>MiB</b>	MebiByte
<b>NLP</b>	Natural Language Processing
<b>OWL</b>	Web Ontology Language
<b>PDF</b>	Portable Document Format
<b>PIL</b>	Python Imaging Library
<b>PP</b>	Percentage Points
<b>RDF</b>	Resource Description Framework
<b>RDFS</b>	Resource Description Framework Schema
<b>RNG</b>	Regular Language for XML Next Generation
<b>RNN</b>	Recurrent Neural Networks
<b>ReLU</b>	Rectified Linear Unit activation
<b>ResNet</b>	Residual Neural Network
<b>SB</b>	Spectral Bandwidth

<b>SC</b>	Spectral Centroid
<b>SNR</b>	Syllabic Nuclear Reduction
<b>SPARQL</b>	SPARQL Protocol and RDF Query Language
<b>STFT</b>	Short-Time Fourier Transform
<b>SVG</b>	Scalable Vector Graphics
<b>SoX</b>	Sound eXchange
<b>SwinTF</b>	Swin Transformer
<b>TDR</b>	Tablature Disambiguation Rate
<b>TEI</b>	Text Encoding Initiative
<b>TF</b>	Tablature F-measure
<b>TMM</b>	Traditional Modal Monodies
<b>TP</b>	Tablature Precision
<b>TR</b>	Tablature Recall
<b>VAE</b>	Variational Autoencoders
<b>vecTrans</b>	Vector Transcoding
<b>VIT</b>	Vision Transformer
<b>WAV</b>	Waveform Audio File Format
<b>XSD</b>	XML Schema Definition
<b>XML</b>	Extensible Markup Language
<b>XQuery</b>	XML Query
<b>XSLT</b>	Extensible Stylesheet Language Transformations
<b>XSL</b>	Extensible Stylesheet Language
<b>ZCR</b>	Zero Crossing Rate





# I

## DISSERTATION INTRODUCTION



# INTRODUCTION

Over the last decade, researchers have been exploring the benefits of their software-related innovations in the music industry to help musicians in their daily tasks. These tasks include encoding their musical piece as music scores using markup languages such as XML for better digitisation and archiving purposes. Having these music scores in a digital format helps the musicians and the developers create computer-aided tools that facilitate musicians' studies across different categories like analysis, encoding, transcription, and classification. The music analysis is performed by retrieving semantic information from the music scores and exposing all its underlying music features. The music encoding consists of adding the pre-analysed features in an explicit form to the music score for better clustering and similarity studies. Music transcription is the technology of accurately annotating musical pieces from hard copies to digital music scores. As for music classification, it consists of grouping the digitised music scores based on several musical characteristics such as genre, timbre, rhythm, melody, etc. This thesis focuses on improving the state of the art of Music encoding, transcription and classification. It presents two main axes: the first axe handles the knowledge extraction and format conversion of music scores studied on a specific corpus of Eastern music. The second axe handles music genre classification and automatic music transcription using deep learning technologies, especially Computer Vision.

This chapter provides an introduction of the previously mentioned axes while explaining the flow of the thesis for passing from one axe to another, then enumerates briefly the contributions while ending with a brief outline.

## 1.1/ MUSIC ENCODING

The journey of seeking digitally aided solutions for music applications started with a collaborative project among the engineering and musicology faculties at Antonine University (UA). This project consists of a music encoding and analysis platform where musicians

can analyse and interpret traditional modal monodies of Eastern music. The latter provided lossless analysis to the musicians while providing instant and accurate results that would have taken longer time and more effort when performed manually. The core of this platform relied on the Modal Semiotic Theory proposed in [91]. "This theory offers an innovative approach to modelling traditional monodies across a vast cultural space that extends from South Asia to medieval Europe, including modern Greece, Central Asia, Western Asia and North Africa. It is based on a transformational morphophonological rhythmic-melodic matrix rewriting of the surface of the monodic segments and on a vector syntactic transformational rewriting of the same monody, which allows a complete description of its derivational and integrative elaboration." As for the platform created, which we will address as Modal Monodies (MM) analyser in this thesis, behaves as follows: It expects as input a traditional modal monodies music score encoded in Music Encoding Initiative (MEI) format [8], analyzes the music scores following the programmatically embedded Modal Semiotic Theory, adds a custom module to the MEI schema to integrate the analysis of the underlying theory, and finally generated a PDF file showing the analysis result and the encoding made to the music score as a visual representation. This output is beneficial for music researchers to understand and analyse this rare category of Eastern music. One of MM analyser's main challenges is that neither the MEI format nor any other encoding format such as MusicXML [7] can support eastern music representations due to better interest in occidental music in the global music industry. Thus, the encoding formats focus on occidental music and lack encoding, analysing, and representing when working with Eastern music. This challenge motivated us the most to pursue the study of this music category and provide better support for Eastern music to help musicians in their analysis studies. It helped us create fruitful contributions that we will present further in section 1.3.

## 1.2/ DEEP LEARNING FOR MUSIC APPLICATIONS

The studies held in the music encoding and analysis part helped us improve the support of Eastern music in the digital encoding domain. Nevertheless, these improvements concerned the post-representation phase of music scores, meaning that they benefit the parts of music encoding where music scores are already digitally transcribed using MEI or MusicXML. Since the majority of Eastern music, especially modal monodies, are represented only in hard copy versions and due to the lack of support for easily digitalising this category of music, it was very challenging to gather a minimum amount of music scores for our deep learning and artificial intelligence (AI) interests in the field. The fast growth of AI and the high interest of the music industry in benefiting from its capabilities motivated us greatly to start experimenting in that domain. Thus, we decided to switch

from the music encoding axe and start learning and experimenting with deep learning for different usages in music applications. Based on the digitalization problem of Eastern music and since deep learning requires a lot of data to function correctly and make predictions accurately, it was decided to search for another music dataset openly available for experimental studies. Moving from the eastern music dataset to the occidental dataset digitally available and already exploitable by other research studies, we explored many deep learning applications such as music genre classification, automatic music transcription, multi-instrument audio separation, music generation, etc. This research focuses on both, music genre classification and automatic music transcription, where we found convenient room for improvement due to challenging competitors in some areas as well as further copyright constraints in others e.g. music generation.

### 1.3/ MAIN CONTRIBUTIONS OF THIS DISSERTATION

The main contributions in this dissertation fall within the aforementioned phases of music encoding and deep learning for music applications. The main contributions can be summarized as follows:

1. First, we propose the MusicPatternOWL ontology that structures the knowledge extraction process of a music pattern analysis algorithm for encoding Eastern music scores. This algorithm consists of the core of the MM analyzer previously presented (section 1.1). The proposed ontology relies on contextual and descriptive elements and attributes of music scores to operate the pattern analysis [91, 119]. It supports the entire music score and its produced pattern analysis to perform information retrieval and analysis. The ontology is not exclusive to Eastern music. Thus, it can support other pattern analysis algorithms in the future.
2. Second, we propose the MEI2JSON converter capable of transforming music scores encoded in MEI to JSON format for pre-processing purposes and future usage in AI techniques. The converter relies on the MusicPatternOWL ontology mentioned previously to structure standard music score content in addition to elements and attributes specific to Eastern music. Thus, MEI2JSON shares the same support for Eastern music scores as the mentioned ontology.
3. Third, we propose a pre-processing approach for generating Short Time Fourier Transform (STFT) spectrograms and upgrades to a CNN-based music genre classifier named Bottom-up Broadcast Neural Network (BBNN). The upgrades concerned the expansion of the number of inception and dense blocks, as well as the enhancement of the inception block through reduction block implementation. The pro-

posed music genre classifier is experimented with the well-known GTZAN and FMA datasets.

4. Finally, we evaluate the state-of-the-art guitar tablature transcription network named "TabCNN" against state-of-the-art computer vision networks. We operate the evaluation using the same dataset and the evaluation metrics of the tablature transcription network. Furthermore, we propose a new CNN-based network named TabInception to transcribe guitar tablatures. This network relies on a custom inception block converged by dense layers. Motivated by the fast growth of Transformer-based networks, we propose a new CNN-Transformer-based network named Inception Transformer that relies on an inception block connected to a Transformer Encoder. The proposed networks, the TabCNN network, and state-of-the-art computer vision networks are evaluated against the GuitarSet dataset to study their leverage for automatic guitar tablature transcription.

## 1.4/ DISSERTATION OUTLINE

The rest of this dissertation is organized as follows: Chapter 2 discusses the state-of-the-art studies on both music encoding and deep learning for music applications while providing a flow from the first axe to the other. Chapter 3 presents the proposed MusicPatternOWL ontology, its relation with the previous work of the MM analyzer, and the utility of such ontology for pattern analysis algorithms. Chapter 4 resolves the issue of converting an Eastern music score from MEI to JSON without losing the generated analysis elements and attributes. Chapter 5 suggests improvements in state-of-the-art music genre classifiers for achieving top accuracy scores over two well-known music genre datasets. Chapter 6 proposes a CNN-based network and a hybrid CNN-Transformer network for guitar tablature transcription. Both propositions were evaluated against the state-of-the-art guitar tablature transcription network and well-known computer vision networks. Last but not least, chapter 7 concludes the work performed in this thesis while opening new possibilities for future work endeavours.



## RELATED WORK





## RELATED WORK

### 2.1/ INTRODUCTION

In recent years, a dynamic fusion of technology and music has brought a wave of innovation that has profoundly reshaped the music industry and addressed the continuously evolving needs of musicians. This harmonious integration of technology and music has resulted in a spectrum of transformative advances that provide musicians with both creative opportunities and practical tools in their daily routines. The integration of deep learning and artificial intelligence into the music industry represents a profound shift. Musicians and researchers now have the tools to reach deep into the realm of music, exploring complex patterns and pushing the boundaries of composition, analysis and performance. This convergence is not only transforming the process of how music is created and shared, but also expanding the horizons of musicians and composers, enabling them to reach new dimensions in their creative endeavours. In this chapter, we explore the landscape of related work in two key axes: Music Encoding and Deep Learning for Music Applications. This comprehensive review will clarify the foundational studies that have informed this research and guided its contributions. In the Music Encoding axis, Section 2.2 explores the domain of encoding Eastern music scores using the MEI format, emphasizing the significance of the Web Ontology Language (OWL) in music encoding and analysis tools. These insights provided the foundation for developing the MusicPatternOWL ontology, our introductory contribution. Section 2.3 builds upon this foundation by examining various ontologies for music analysis and exploring related work on music score converters for transforming compositions between different formats. We highlight different converter approaches while focusing on the proposed MEI2JSON converter. Section 2.4 bridges the gap between music encoding and deep learning for music applications by introducing essential audio features that find visual representation in audio signals. We discuss their applications and draw insights from related work that directly influenced our research. In the Deep Learning for Music Applications axis, Section 2.5 presents a synthesis of studies that leveraged visualized audio features (discussed in Section 2.4) for

various music applications. We delve into CNN networks, Transformer networks, and hybrid CNN-Transformer-based networks, as these are central to this study. Moving closer to the domain of music genre classification, Section 2.6 focuses on recent and efficient approaches using visually aided audio features, particularly spectrograms. We highlight the application of these methods on well-known datasets like GTZAN and FMA. Section 2.7 explores the domain of automatic music transcription with a specific focus on guitar tablature estimation. We discuss influential studies and their findings, contributing to the development of effective guitar tablature transcribers. Lastly, in Section 2.8, we provide an overview of essential music datasets used in music application experiments, including those used in our study—GTZAN and FMA for music genre classification and GuitarSet for guitar tablature transcription. This comprehensive review of related work serves as a paramount foundation for understanding the context and significance of the research contributions in subsequent chapters.

## 2.2/ KNOWLEDGE EXTRACTION AND REPRESENTATION OF MUSIC SCORES

Many studies were found in the literature to discover the benefits of digitally annotating and analyzing music scores. A tool for analyzing music patterns was introduced in [83]. It offers a platform where users can perform quantitative analyses on MusicXML files. The platform displays an interface where users can search for music scores based on pattern similarity factors. This approach was limited to support many mandatory parameters (such as dynamics, octaves, and scores with limited notes number). The digitization of musical analysis theories was found to be a collaborative opportunity between developers and musicians. Developers provide encoding solutions, while musicians can obtain precise and efficient analytic results with a reduced amount of time.

Several theory-based solutions took place for analysing music scores and partitions. In this paragraph, we discuss solutions developed for analyzing music and present contextualised modules added to MEI for enriching the MEI standard schema. The author in [46] developed a user interface for Schenkerian Analysis, aiming to analyse musical scores based on the Schenkerian theory proposed by Heinrich Schenker. Challenged by the complexity of its computer implementation, the authors in [87] developed a solution for Lerdahl and Jackendoff's Generative Theory of Tonal Music. The developed solution was implemented and tested using four different analyzers. The results revealed that FATTA (Fully Automatic Time-Span Tree Analyser) outperformed ATTA (Automatic Time-Span Tree Analyser), particularly in the analysis of metrical structures. Additionally, this study explored optimal clustering in the  $\sigma$ -GTTM-II analyzer, demonstrating its superior performance compared to other methods. Furthermore, the study included comparisons with

manual analyses by musicologists, highlighting both alignment and variations in results, emphasising the complex nature of music analysis.

The Text Encoding Initiative (TEI), a standard developed for encoding texts, extended its support to reach the encoding of music within texts. The TEI encodes texts and music occurring within texts, considering musical pieces or notes as images [69]. By adding the *notatedMusic* element to the TEI, the latter was able to support the inclusion of music expressed in MEI, a graphical representation of the music or any other format representing the music [69]. All MEI elements within the *notatedMusic* element are prefixed with "mei:", for example, *mei: music* [69]. The project described in [84] uses both MEI and TEI, creating a data model and using both the MEI and TEI for encoding holdings of the Detmold Court Theatre (1825-1875), providing a catalogue, which was also used as a searching tool for specific data [84, 151]. The Solesmes module proposed in [43] captures Solesmes-specific music notation about Gregorian chant. According to [43], while the MEI supports encoding neume music notation, it lacks certain features specific to Solesmes neume notation. Therefore, a new module is introduced, adding elements and attributes to the MEI schema, allowing for a more accurate representation of Solesmes neume notation features. Another module creation effort was proposed in [60]. It consists of adding layout-related components in MEI since the latter does not encode information concerning the layout. The layout module creation enabled encoding information concerning multiple visual representations of the music while keeping the musical content intact.

A musical analysis theory named "Modal Semiotics theory" proposes an innovative approach to traditional modal monodies (T.M.M) across a vast cultural space stretching from South Asia to medieval Europe, via modern Greece, Central Asia, West Asia and North Africa [91]. It is based on a transformational morphophonological rhythmic-melodic matrix rewriting of the surface of monodic segments and on a transformational syntactic vector rewriting of the same monody, which allows a complete description of its derivational and integrative elaboration. The semantic component of this grammar is based on its phonological, morphological and syntactic components, with particular emphasis on the vector semantic modalities inherent in modal syntax. This approach leads to a neurocognitive perception of the structures revealed by modal semiotic analysis.

Among this vast cultural space of the theory, this thesis focuses on supporting the studies of the Middle Eastern and Mediterranean cultures. In [119], a semantic-based platform is proposed to encode and analyse T.M.M. Thus, The encoder analyses T.M.M music scores by extracting the underlying semantic features, and in the end, adds a custom module to the MEI schema to provide a explicit visualization of the encoding. It is important to mention that the extraction process consists of applying two input matrices that correspond to several combinations of music score patterns. Thus, it is significant to

develop an ontology to structure the elements of a music score.

Ontologies validate the semantic representation of musical concepts, such as notes, rhythms, dynamics, instruments, and musical structures. This semantic representation allows for a more meaningful and machine-understandable description of musical elements in a score. Music notation software and systems often use different file formats and data structures. Ontologies help bridge the gap between these diverse systems by providing a common, standardized vocabulary and structure for representing musical information. Musicologists and researchers can use ontologies to annotate and analyse musical scores. They can assist them in exploring the relationships between musical elements, styles, and contexts. In addition, they enhance the search and retrieval of musical scores and related resources. Users can query a database or digital library using semantic terms defined in the ontology, thus helping search for specific musical elements and attributes. Ontologies can inform the development of music notation standards, ensuring that notation systems remain consistent and expressive while accommodating the needs of different musical genres and traditions.

### 2.3/ MUSIC SCORES ORGANIZATION AND CONVERSION

Numerous studies were proposed to develop and manage ontologies related to music score content. Jones et al. (2017) [110] developed an ontology to semantically annotate and reason upon Western music scores. The proposed ontology helped in exploring the benefits of the web ontology language (OWL) in music-related fields. Also, due to the need to extract knowledge out of music data and manage this extraction process, an ontology took place in Cherfi et al. (2017) [100] to integrate semantic music elements. This work helped in normalising the representation of music theories in a way they can be linked together.

Studies went extensive in the music field, especially when both developers and musicians found fruitful results in their collaborative opportunities. We proposed an ontology named MusicPatternOWL (El Achkar and Atéchan, 2020) [150] to cover the structural and behavioural aspects of a pattern analysis algorithm for encoding eastern music scores. This ontology supports the semantic information retrieval and analysis processes of music score contents. The paper presented a proof of concept of its usage with an algorithm proposed by Abou Mrad (2016) [91] and developed in Asmar et al. (2018) [119] for analysing and encoding traditional modal monodies of the Mashreq, a unique corpus in eastern music.

Many music scores are usually encoded using symbolic formats such as MEI (Roland, 2002) [8] and MusicXML (Good, 2001) [7]. These formats and especially MEI, are XML-

based formats relying on XML schemas to describe the structure of their elements and attributes. This is where frameworks like JXML2OWL took place in Rodrigues et al. (2008) [26] to manually map XML schemas to existing OWL ontologies and later, automate the transformation of XML instances into individuals of the mapped ontology. These frameworks helped efficiently transform the syntactic representation of data (using XML) to a semantic one (using OWL). This transformation provided the ability to perform inference on a knowledge-based model for better data exchange and integrity. Another mapping solution was to develop (Lacoste et al., 2011) [44] an efficient framework for generating ontologies automatically out of XML instances. This framework helped in creating a good description of the OWL model and XML instance files. The introduction of both manual and automatic mapping frameworks (between OWL ontology and the XML schema) allowed accessing XML encoded data from Semantic Web applications that are already connected to OWL ontologies. This is where frameworks like SPARQL2XQuery took place to accentuate the adjacency and interoperability of both OWL and XML. Therefore, the proposed framework (Bikakis et al., 2009) [28] was able to evaluate SPARQL queries over XML data after mapping XML to OWL Schemas.

Mapping frameworks were inspirational especially when the transformation rules between XML and OWL Schemas could be saved and reused upon demand by storing them in XSL stylesheets. The usage of XSL files as the holder of mapping rules encouraged their employment in multiple data format converters, by the fact that they will ensure data conversion without losing data quality through the direct mapping between schemas in terms of datatypes and property rules. As for music-related research, a toolkit named music21 took place in Cuthbert and Ariza (2010) [32] to provide software tools for both musicians with little programming experience and to programmers for analysing, searching and transforming music scores in symbolic forms. This toolkit did not use XSL stylesheets but provided several conversion supports to its specific format. This project supports the conversion of several symbolic formats including MEI, MusicXML, and MIDI (The MIDI Manufacturers Association, 1995) [3]. With the evolution of the MEI format, Verovio, a music engraving library was developed by Pugin et al. (2014) [76] to provide a visual representation of music scores encoded in MEI into SVG. This library also provided the capability to convert MusicXML to MEI and vice-versa based on the MEI XSL stylesheets available on the MEI encoding tools on GitHub (<https://github.com/music-encoding/music-encoding>).

The conversion via Verovio had limited capabilities, it focused on the main elements and attributes of music scores while excluding others. Also, an MEI-related conversion framework named Meico took place in Berndt et al. (2018) [121] to provide a novel tool that processes MEI-encoded music scores. Meico helped in converting MEI data to multiple symbolic formats like MusicXML. The conversion was based on the same XSL stylesheets used in Verovio where it also lacked the conversion of all the elements and attributes of a

music score encoded in MEI. Another study presented in Alvaro and Barros (2010) [30] focused on developing a music composing system named Computer Music Cloud (CMC) as well as a suitable data representation format named MusicJSON to efficiently compose and store music scores in the computer music cloud. The MusicJSON was considered a music interchange tool between different services of the CMC. It was used also as a music data unification tool by converting several symbolic formats including MusicXML to a music representation format in JSON.

While there isn't a specific tool for converting MEI directly to JSON, this process is still achievable using generic programming languages to parse MEI files and extract the desired information in JSON format. Below we list the most common methods for achieving MEI to JSON conversion, along with the major drawbacks of each method.

The first method would be custom scripting, which consists of writing custom scripts in a programming language such as Python or JavaScript to parse MEI files and convert them to JSON. Common XML parsing libraries can be used in this case, such as lxml [9] and BeautifulSoup [22], which are commonly used for web scripting utilities such as data crawling and web scraping. The disadvantages of the latter approach are the lack of capacity to handle all MEI documents and to keep up with changes to the MEI and JSON schemas.

The second method is the idea of converting the score from MEI to the most common format called MusicXML [32, 76] and then converting the MusicXML to JSON. The disadvantage of such an approach is that multiple conversions may result in the loss of some MEI-specific information since the MusicXML format is not able to encode all the schema details of the MEI format.

The third and final method is to use a combination of existing converters, such as using the converter proposed in [121] to achieve the first half of the conversion, and then using the converter proposed in [30] to process the result of [121] and achieve MEI to JSON conversion. The main drawback of this method is the limited coverage of such converters to encode new variants and updated schemas of MEI, which affects the output quality of the combined approach.

The most common and essential drawbacks of all the latter methods are the challenge of continuously changing the configuration of the converters to support MEI schema changes, as well as the loss of semantic information when converting music data from one format to another where some music notations, elements and attributes are not available. To address these challenges, an effective solution involves the use of ontology-based approaches. These approaches provide a structured and adaptable framework for representing music data, allowing ontologies to evolve along with schema changes. The use of ontologies significantly optimises the ongoing reconfiguration process and offers a promising solution to these persistent problems.

## 2.4/ FROM MUSIC SCORES TO VISUALLY REPRESENTED AUDIO FEATURES

In this section, we will discuss essential audio signal processing techniques to extract visual representations of the music datasets. These visual representations serve as vital preprocessing steps and play a fundamental role in enabling our deep learning-related contributions, particularly within the domain of computer vision. Also, we will elaborate on our motivation for handling audio data as a computer-vision use case, found to be an unconventional approach for this type of data.

Taking a detour to define audio signal processing and highlight its visual representation studies, this field consists of the analysis and transformation of audio signals represented in digital forms. It includes various techniques and algorithms for extracting, modifying and visualizing information from audio data. Audio signal processing can be used as a converter from audio to visual representation for various audio-related use cases, such as music analysis [20, 59, 13], speech recognition [57, 64, 41], sound synthesis [37, 2, 34], music generation [115, 133, 104], etc.

Audio signal processing consists of the following steps:

The first step, Data Acquisition, concerns capturing analogue audio signals using a microphone or other audio-capturing sensors and converting them into digital form for further processing [5]. This step involves analogue-to-digital converters (ADCs) to transform continuous analogue audio signals into discrete digital samples [24]. The result of this step is digitised audio data represented as a time-domain waveform, where amplitude values are recorded depending on the time.

The second step, Digital Signal Processing (DSP), involves applying several techniques for modifying and analysing digitised audio data [21]. This step is widely exploitable by researchers where a wide range of studies can be performed such as filtering, equalisation, compression, and effect processing [36]. The majority of DSP studies rely on Fourier transforms for passing from time domains to frequency domains, as well as filtering through convolution and basic mathematical operations over audio samples.

The third step, Feature extraction, identifies and extracts relevant parameters from audio data for better interpretation and visualisation. This step is essential for deep learning studies since it reduces the dimensionality of audio data while preserving the needed information only. Thus, it is considered a pre-processing technique for AI use cases [65, 93, 108]. The extracted features consist of three categories: domain features, frequency-domain features, and time-frequency features. We will showcase the most used and beneficial features in the upcoming step while providing a visual representation of each feature.



The fourth and last step, Visualization, involves the previously extracted features in a visual format to provide assistance for analysis, interpretation, or use of case studies. This step is influential for quality control and debugging especially in applications like music production, audio analysis, or anomaly detection through audio surveillance.

In the following, we display common audio feature visualization processed on a 30 seconds Blues music from the GTZAN dataset [6]. These visualisations represent an essential preprocessing phase, which we will discuss in more detail in subsequent sections. These steps are essential for preparing the data to feed into deep learning models, which we utilise extensively for various analytical studies, including music classification and transcription, both of which are central to this study.

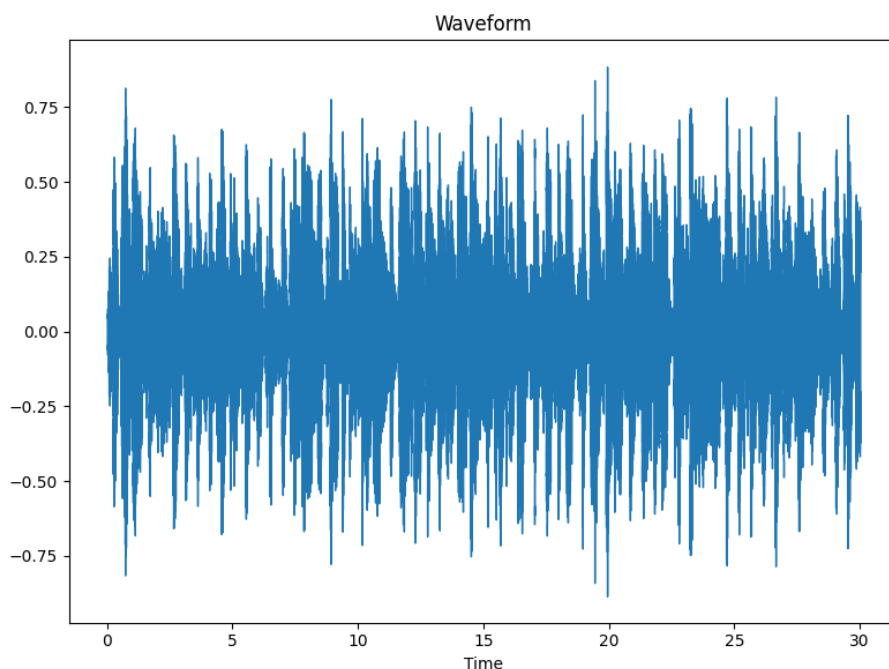


Figure 2.1: Waveform Visualization of a Blues Track in the GTZAN Dataset

1. **Waveform**, is a visual representation of the audio signal that shows the amplitude of the signal as a function of time. The vertical axis of the waveform represents the intensity of the signal, usually measured in decibels (dB), while the horizontal axis represents time, usually measured in seconds. The shape of a waveform varies depending on the sound being represented. Common waveform shapes include sine waves, square waves, triangle waves and complex waveforms (see figure 2.1) which are a combination of different frequencies and amplitudes. Although waveforms are a basic visual representation, researchers have not been interested in exploiting this visual representation because the community is interested in analysing and processing the signal, not just the basic representation of audio signals. Nevertheless, the music industry, and producers in particular, rely heavily on the waveform

of the signal to identify and manipulate specific parts of the signal, such as cutting, copying and pasting, or applying special effects to different parts of the signal [55, 166, 31].

2. **Mel-Frequency Cepstral Coefficients (MFCCs)**, are a set of features that represent the spectral characteristics of an audio signal. The Mel scale is a perceptual pitch scale that approximates the human auditory system's response to different frequencies. MFCCs are computed by dividing the audio signal into small overlapping frames and applying a series of filters spaced according to the Mel Scale to produce a filter-based spectral energy representation. The output of the previous step is compressed using the mathematical logarithmic scale for better perception of the loudness of the audio signal. Finally, a Discrete Cosine Transform (DCT) is applied to the result to decorrelate the coefficients and reduce the dimensionality of the extracted feature. The resulting coefficients are the MFCCs (see figure 2.2).

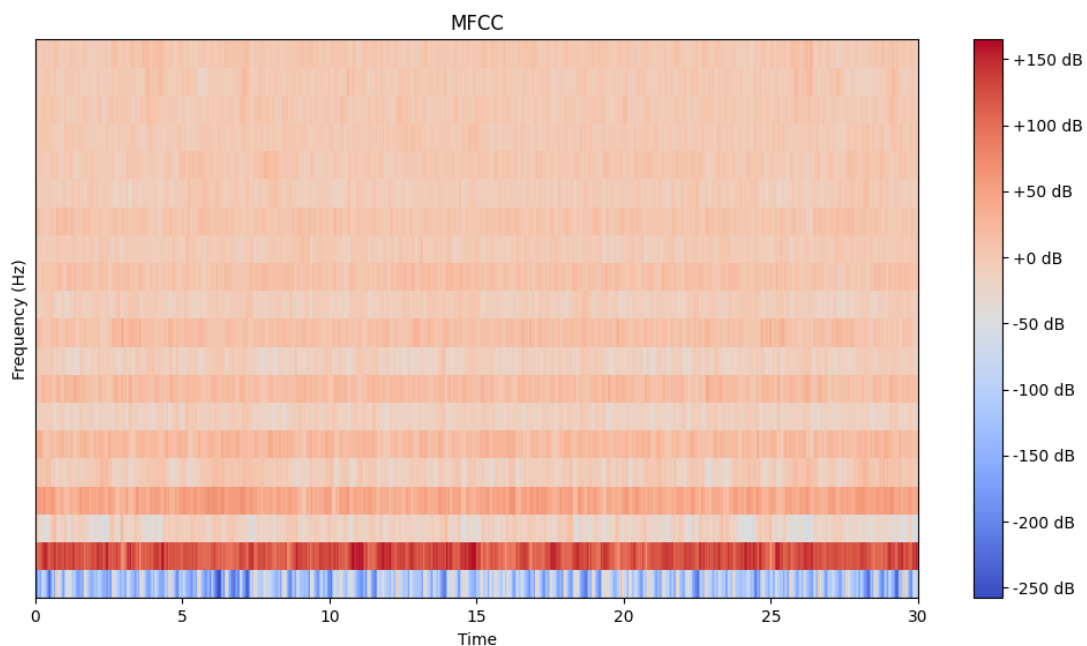


Figure 2.2: Mel-Frequency Cepstral Coefficients Visualization of a Blues Track in the GTZAN Dataset

In short, MFCCs are popular in various audio processing applications such as speech recognition [1, 63, 82], music information retrieval [4], and other audio-related tasks such as music genre classification [10] and music transcription [66] due to their efficiency in capturing spectral information while reducing the dimensionality of the feature vector.

3. **Spectrogram**, is a visual representation of the spectrum of frequencies in an audio

signal as they vary with time. The time is represented on the horizontal axis, while the frequency is on the vertical axis. The intensity of the frequencies at a given time is represented in colors or shades. The low-level frequencies are colored in light colors and the high-level frequencies are colored in dark colors. Spectrograms can be used for a variety of applications in audio processing. They are flexible in helping us visualize and compute many audio features such as the MFCCs shown earlier in figure 2.2, the Short Time Fourier Transform presented in the figure 2.3 and used in our contribution [165], as well as the constant-Q Transform (CQT) (figure 2.4), and many more. Since our contributions exploited STFT and CQT spectrograms the most, it is important to explain both techniques.

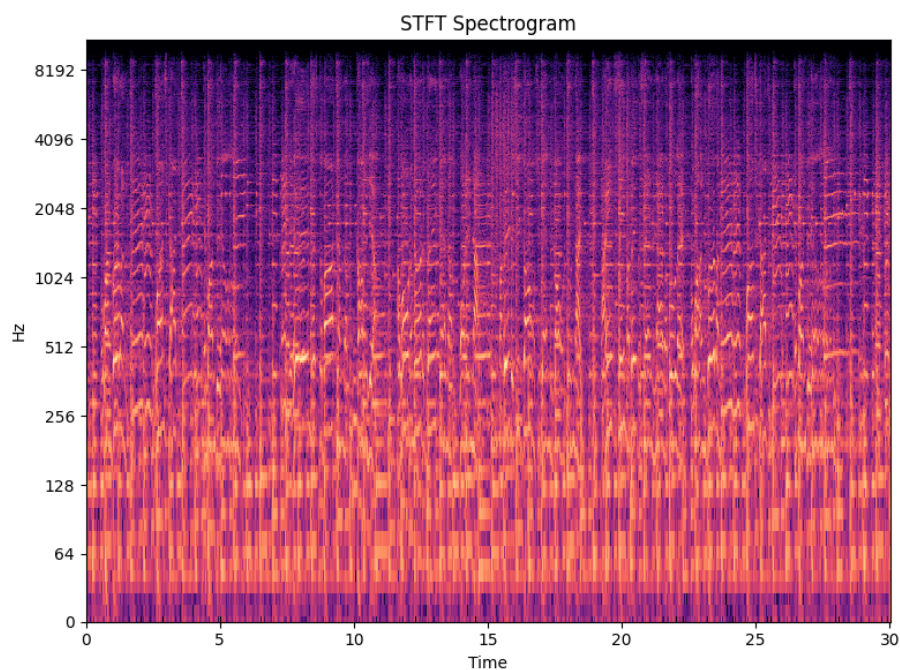


Figure 2.3: STFT Spectrogram Visualization of a Blues Track in the GTZAN Dataset

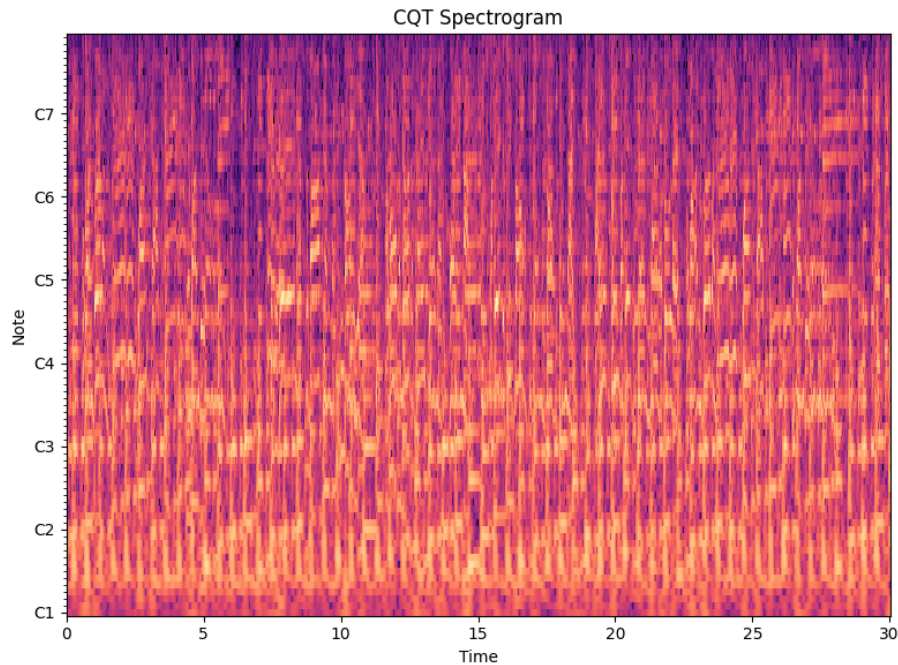


Figure 2.4: CQT Spectrogram Visualization of a Blues Track in the GTZAN Dataset

STFT is a technique to transform time-domain signals into time-frequency representations. It works by dividing the signal into small overlapping time frames and performing a Fourier transform on every frame. Unlike STFT, CQT relies on a logarithm-spaced frequency scale to transform time-domain signals into time-pitch representations. This provides a better frequency resolution at lower frequencies found to be useful for music analysis purposes. The choice between STFT and CQT depends on the analysis task. STFT are more commonly used in general audio signal processing due to its versatility [99, 33], while CQT is preferred in perceptual-related tasks [59], such as pitch detection and musical note transcription [51].

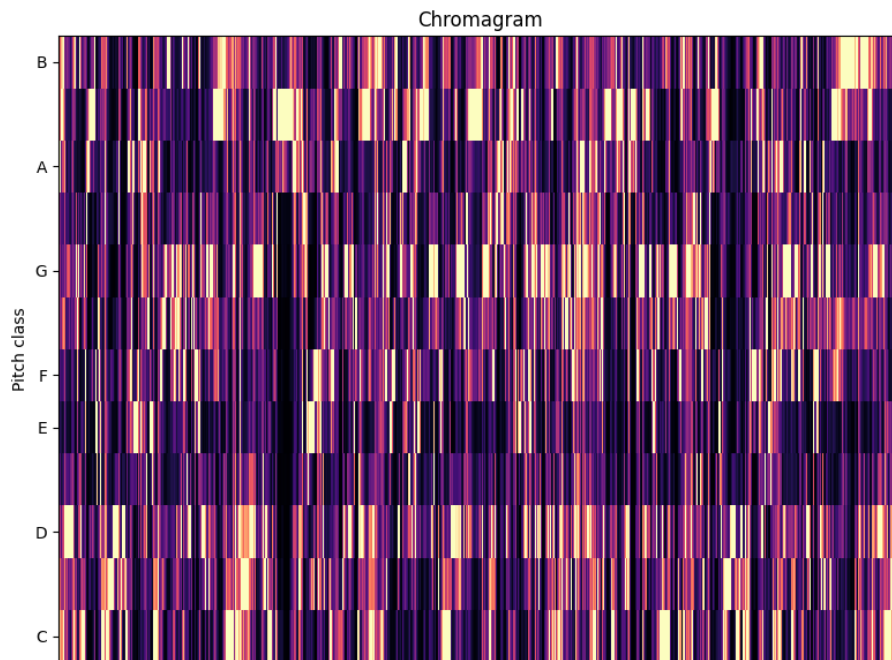


Figure 2.5: Chromagram Visualization of a Blues Track in the GTZAN Dataset

- 4. Chromagram**, is a representation of the twelve music pitches in an audio signal. It provides a method to summarise the distribution of musical notes in a specific audio signal. Each bin in the chromagram corresponds to one of the twelve pitches (C, C#, D, D#, E, F, F#, G, G#, A, A#, B) and the values inside each bin represent the energy of that pitch in the audio signal.

Chromagrams are created by computing the previously mentioned STFT spectrograms at the first stage. The energy of each pitch class is then computed by grouping the bins of the spectrogram that fall within the same pitch range. Thus, the spectrogram bins are grouped by the twelve different pitch classes. The values of the chromagram are then normalized to ensure a relative pitch energy among each pitch class, and a logarithmic scale is applied in the end to provide a realistic human perception of each present pitch class (see figure 2.5). Chromagrams are generally exploitable in music analysis tasks [174, 42], melody or pitch extraction [61, 140], and chord recognition [14, 112].

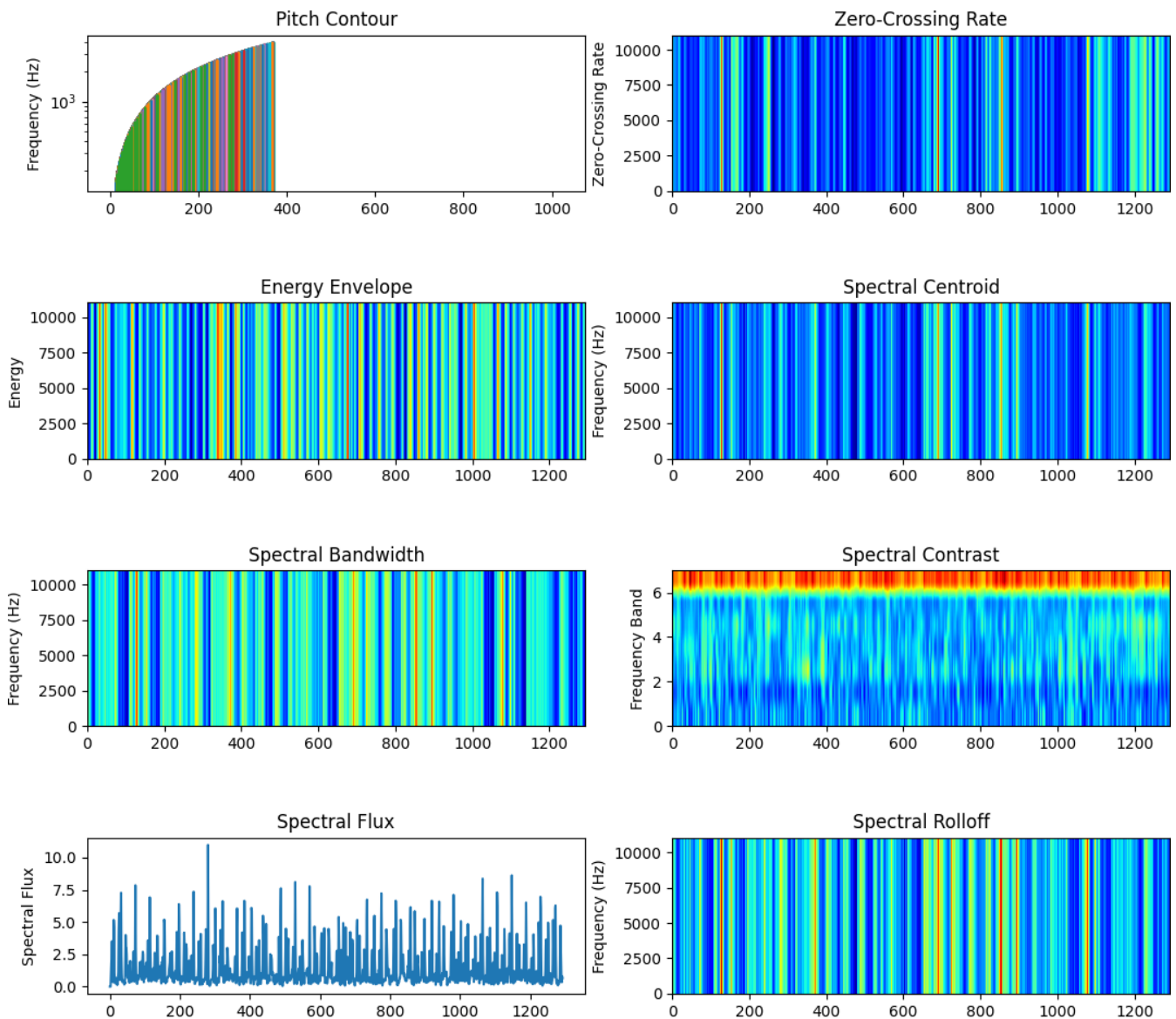


Figure 2.6: Common Audio Features Visualization of a Blues Track in the GTZAN Dataset

The audio feature visualizations presented above are the features that we leverage the most in this study. Nevertheless, other common features can be exploited out of the audio signal. Figure 2.6 gathers all these features together while showing their visualizations when processed over the same Blues track used across this section. We will list these features below while mentioning the best studies that leveraged them.

**Pitch Contour** is a rendering of the fundamental frequency (F0) variations over time in an audio signal. F0 is the frequency of the lowest harmonic in a sound wave. It corresponds to the perceived pitch of a sound. The pitch contour visualization is used in several audio and music processing tasks such as speech processing [16], music analysis [61], and emotion recognition [39]. It helps identify specific audio characteristics like the melody, intonation patterns in speech [68], and other musical or prosodic features [72].

**Zero Crossing Rate (ZCR)** is a visual representation of how many times the signal crosses the 0 dB (from negative to positive or vice versa) in addition to the rate of this crossing within a given time frame. This feature is mostly useful for speech and music analysis [185, 27] tasks due to the information that it provides concerning the frequency distribution as well as the periodicity of the signal. Since the ZCR feature is very specific, it is often combined or leveraged alongside other features to extract useful information from the audio signal [86]. It can also be used in broader tasks such as electrohysterogram signals in [175] or speech analysis for medical diagnosis in [177].

**Energy Envelope** is the representation of the energy (also known as magnitude) variations in an audio signal as a function of time. While there aren't any major researchers who have relied on this feature extensively, it remains a very useful feature for speech and music analysis and is embedded in most music production platforms [166, 55]. An interesting use of the latter feature is to perform envelope extraction. This extraction is often used in voice activity detection systems to distinguish between speech, silence, and background noise in an audio signal [48, 192]. It can also help in speaker identification tasks to identify and separate different speakers in an audio recording [176]. The energy envelope helps by segmenting the audio into speaker-related locations.

**Spectral Centroid** represents the weighted average of the frequencies present in an audio signal's spectrum. This audio feature can be calculated using the following mathematical formula :

$$SC = \frac{\sum_f (f \cdot S(f))}{\sum_f S(f)}$$

$f$  representing the frequency of the spectral component.  $S(f)$  is the energy or magnitude of the spectral component at the given frequency  $f$ . The result given in the  $SC$  variable is the sum of all the frequencies multiplied by their corresponding magnitude over the sum of all magnitudes. Similar to previous features, the Spectral Centroid is often used in audio classification [186, 181], speech processing, and music analysis applications [194, 195].

**Spectral Bandwidth** is the feature that measures the range of the frequencies present in the audio signal. A common way of calculating the Spectral Bandwidth is by adopting the standard deviation of the frequencies in the signal. It can be calculated using the following mathematical formula :

$$SB = \sqrt{\frac{\sum (f_i - f_{mean})^2}{N}}$$

$f_i$  represents individual frequency components in the spectrum, while  $f_{mean}$  is the mean frequency of all frequency components, and  $N$  is the total number of frequency components. While this audio feature has not been exploited as a key feature for audio-related research fields, it remains available in processing libraries such as [196, 193] for music analysis purposes.

**Spectral Contrast** is a feature that interprets the difference in magnitude peaks and valleys in the audio signal's spectrum. It measures the degree of contrast between different frequency bands in a spectrogram. Computing the Spectral Contrast can be performed by first dividing the audio signal into frames of equal duration sizes. The STFT technique is then applied to each frame to compute the magnitude spectrum. Last but not least, the resulting spectrum is divided into frequency bins where the spectral contrast can be calculated by differentiating the maximum magnitude in a frequency bin from the minimum magnitude of the adjacent frequency bin.

Some insightful usages of this feature are combining it with other features to classify emotional states [164, 163], or adopting the combining approach for music-related studies such as music genre classification [19].

**Spectral Flux** is the feature that measures how swiftly the energy distribution evolves in the frequency domain from a time frame to its adjacent. It can be calculated by computing the squared Euclidean distance between the magnitude spectrum of the current frame and the magnitude spectrum of the previous frame. The spectral flux is typically computed for each time frame of the audio signal, thus resulting in a time series of spectral flux values that describe the spectral changes over time. This feature is often used to detect onset events for music analysis purposes [52, 77, 23].

**Spectral Rolloff** is the feature used to represent the shape of the spectral distribution of an audio signal. It is defined as the frequency below which a specified percentage of the total spectral energy is located. The latter frequency represents the threshold in the frequency spectrum below which the specified percentage of energy is concentrated. Spectral Rolloff is used in audio-related tasks that rely on the distribution of energy in the frequency domain. Some interesting applications of the spectral rolloff feature include speech recognition [137], audio classification [67, 58], and musical instrument classification [157, 62].

When explaining the previous features, we defined some as representing energy and others as representing magnitude. These two measures are related to the field of audio signal processing, but they are not the same. Both provide information about the amplitude of the signal. However, energy measures the overall power of the signal over time, while magnitude measures the instantaneous amplitude at a given point in time. Mathematically, the energy is calculated by summing the squared values of the signal samples. As for the magnitude, it is the absolute value of each sample in the signal.



## 2.5/ COMPUTER VISION FOR MUSIC APPLICATIONS

The processed audio features to visual representations presented in the section 2.4 are key elements for the use of Computer Vision (CV) in this domain. They provide visual representations of audio data, which can then be processed and analysed using computer vision techniques for various purposes in music applications. Computer vision is an interdisciplinary field of artificial intelligence that aims to replicate human visual perception by extracting meaningful data or features from visual data. Therefore, in the music application context, CV refers to processing visual data associated with music-related content, such as spectrograms and chromagrams.

We present the three main practices of computer vision in music:

The first practice, music transcription, corresponds to the conversion of musical recordings into symbolic notations, such as sheet music [8, 7] or MIDI files [3]. This practice is essential for music education, analysis and the creation of new arrangements. A major project called "Magenta" was initiated by Google, where several researchers studied music transcription using deep learning models. They demonstrated the use of WaveNet [93], CNNs and deep generative models [132] for music transcription tasks. The latter project involved note detection, where CV models are trained to detect musical notes and their corresponding attributes within spectrograms. The extracted knowledge can then be used to transcribe music into symbolic notation. Another initiative in this area is the use of Transformer-based models [187] for polyphonic music transcription, which corresponds to the transcription of multiple instruments played simultaneously in a music recording [191, 198].

The second practice, music classification, involves sorting music recordings into different categories (genres, instruments, etc.). This practice is valuable for music recommendation systems in music streaming services (Spotify, Anghami, Deezer), where such classification can help organise content and predict new music to play based on user preferences. An insightful approach to music classification is to extract visual features from album covers using CV models [45]. The extracted features can be combined with other audio features to detect the genre or mood of the music [103]. While some researchers have relied on visual features alone, others have combined both audio and visual information, such as music videos and album art, to achieve an improved version of their classification models [92]. This combining technique is also called multimodal fusion, where information belonging to different modalities can be integrated for a converged purpose [117, 173]. Last but not least is the use of computer vision techniques to facilitate the analysis and classification of rendered audio features based on acoustic characteristics [159, 171].

The last practice, music generation, involves creating new music compositions based

on previously trained music datasets. This practice is reflected in several applications such as music composition and soundtrack creation for cinematic movies. Some music generation efforts include symbolic music transcription where both Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) are used as CV-based networks to achieve multi-track music generation [122]. Other efforts relied on both CNN and transformer-based networks to generate coherent music compositions by modelling dependencies in music sequences [126, 154].

While mentioning the main practices of computer vision for music applications, several research studies were cited. These citations presented different classes of deep learning for music-related applications. Some researchers relied on CNN and Transformer-based networks, while others relied on VAEs and GANs. Since our thesis focuses the most on CNNs and Transformers, we will elaborate on each in the following, while mentioning some of their most efficient usages in music applications. Also, we will elaborate on the most recent studies that leveraged combining both CNN and Transformers networks, as we did in our guitar tablature transcription contribution developed in Chapter 6.

Convolutional Neural Networks (CNNs) are a class of deep learning designed to process images and, in our case, spectrograms. CNNs have a great ability to automatically extract hierarchical features from raw pixel data using their many layers. These layers include, but are not limited to, convolutional layers, pooling layers, and fully connected layers. Convolutional layers are small filters applied to the input image to extract local patterns or features. They are reliable for capturing low-level details such as edges, textures and shapes. In the music context, this behavior is essential to capture the harmonics and the beat from the spectrogram, which often resides in the lowest part of the spectrogram [168, 165]. The pooling layers are the downsamplers of the data to reduce its spatial dimension. They are used as essential layers to minimise the networks' complexity and to avoid overfitting scenarios. The deeper the network, the greater its ability to compute complex features. Thus, we rely on fully connected layers to ensure a connection among all neurons of the networks and learn global patterns that help us predict accurate results.

The researchers have exploited CNN-based networks in many music applications that we have already cited in this chapter. We elaborate below on some of the studies that had an influential impact on this research field. A paper proposed in [101] explored various CNN architectures to achieve music classification as well as content-based music recommendation. Other studies leveraged CNN for achieving instrument recognition [89]. Moreover, studies went broader in the field, where some researchers were interested in modelling the body and finger movements of musicians when they perform music [152].

Transformers are a class of deep learning originally introduced for Natural Language Processing (NLP) use cases [118]. The latter class relies on a self-attention mechanism that enables the modelling of complex dependencies in data. The self-attention mechanism

is responsible for computing weighted sums of all elements in the data sequence based on their relevance to each other. This mechanism allows each element in the sequence to examine the importance of the remaining elements. Transformers enable analytical decision-making by using multiple attention heads to capture different dependencies and features in parallel. The analysis of elements per sequence in Transformers is not executed in a mathematical order. Therefore, they use positional coding at the input layers of the network to provide order information about each element.

The positive impact of the Transformers in NLP-related use cases encouraged researchers to adopt this deep learning class in the vision domain. These studies were also reflected in music applications since it can be seen from a mathematical point of view as a data series of notes, making it an approachable field with NLP studies. A Transformer-based network took place in [126] to reduce the intermediate memory requirement to linear in the sequence length of Transformers. The latter approach proved its capability to generate minute-long music compositions while being evaluated using the JSB Chorales [54] and the Piano-e-Competition [142] datasets. Inspired by the sequence modelling capabilities of Transformers, another music-related study took place in [154] where it created a Pop Music Transformer that can compose piano music of the Pop genre with improved rhythmic structure with regard to previous efforts.

As enlisted previously, both CNNs and Transformers have a great contribution in multiple music application use cases. CNN-based networks can efficiently capture the local patterns and features through convolutional layers for classification tasks. They typically involve fewer training parameters than Transformers, making them the convenient approach for resource-constrained devices, such as mobile applications. Transformers-based networks can capture global context information efficiently making them suitable for music generation tasks. Also, Transformers require fewer preprocessing steps in comparison with CNNs, which is helpful in versatile use cases such as speech recognition.

Consequently, CNNs are ideal for tasks involving spatially structured data with local patterns, while Transformers are more suitable for sequential data.

The clear advantages of CNNs in capturing local features and the advantages of Transformers in perceiving global dependencies encouraged the researchers to study a combination of both networks for various use cases. A hybrid model that combines CNN and Transformers was proposed for decoding motor imagery Electroencephalography (EEG) signals. The results showed that the created CNN-Transformer model is a competing strategy for improving EEG classification use cases [190]. Another network named CTC-NET relied on combining a CNN-based network with the decoder of the Swin Transformer [169] for image segmentation studies. The proposed network surpassed CNNs and Transformers performances on different medical applications, including multi-organ segmentation and cardiac segmentation [202]. A further study on multi-organ segmen-

tation led to create the CoTr network for achieving accurate 3D medical image segmentation. This approach bridges CNN and Transformers' efforts to achieve a recognisable solution for processing high-resolution 3D feature maps. Likewise, a combination of CNN and the Swin Transformer networks created a pyramid structure network for feature encoding and decoding [201]. This approach proved its importance while outperforming state-of-the-art image segmentation methods on breast ultrasound lesion datasets. On the path to creating resource-efficient networks for mobile vision applications, another hybrid network named EdgeNeXt took benefit from combining CNN and Transformers. The proposed network was able to outperform the well-known MobileViT network in terms of accuracy score [199]. The leverage of image super-resolution in many industrial fields, such as medical diagnosis by providing a detailed image through their zoom abilities or surveillance and security by enhancing image details for forensic analysis, encouraged researchers to apply the hybrid approach to such techniques [183]. Last but not least, a recent hybrid approach took place to provide quality assessment for full-reference and no-reference images. The approach relied on combining a Vision Transformer [149] Encoder with a CNN-based decoder for quality estimation. The proposed solution achieved great results over all the datasets used in the paper [203].

As shown in the previous paragraph, many researchers studied the leverage of a network combination in different areas, where the majority resulted in fruitful improvements in their corresponding domains of use. However, and to the best of our knowledge, there haven't been any CNN and Transformer combination efforts for music-related applications.

The deep learning contributions of our thesis had two different approaches. The first approach relied on CNN-based only to achieve music genre classification, and the second approach leveraged a CNN-based approach and a hybrid CNN-Transformer-based approach to achieve automatic music transcription. The common point between these two approaches is the fact that their CNN-based improvements were inspired by the architecture of the well-known Inception network [81], and precisely the 4th version of it [116]. We list below the best efforts related to Inception for music applications while explaining the advantages of each version in comparison with its precedent.

The Inception network or GoogLeNet is a CNN-based architecture built by the Google research team for image classification and object detection tasks. The inspiration behind this network is the idea of creating multiple filters with multiple scales to capture different depth views of the same feature in an image. The Inception network then relies on convolutional layers and pooling layers both in parallel at the same layer level to capture fine-grained and large-scale features simultaneously [81]. The first version, Inception v-1, was considered computation-efficient since it consisted of fewer parameters than other deep learning networks. It introduced a parallelism behavior for convolution and pooling layers before being fed to the next layer in a concatenated form. This parallelism behavior,

also known as the Inception module, helped reduce overfitting scenarios [162, 161]. The added value that the Inception network was able to bring led to the release of improved versions for achieving better performances. The Inception-v2 introduced the "Batch Normalization" layer for training stability and faster convergence. It also implemented factorized convolutions to reduce the computational cost [79]. The Inception-v3 proposed replacing big convolutional filters with multiple filters of smaller size to reduce complexity [96]. Inception-v4 introduced the concept of "reduction blocks" to reduce spatial dimension and increase efficiency. It was also combined later on with a residual connection to increase performance and achieve faster convergence. This version was known as Inception-ResNet [116]. Furthermore, some minor design refinements were made in the same version and a last combination was proposed with Feature Pyramid Network (FPN) [111] for object detection usages.

As for the implementation of Inception Networks in music-related applications, a CNN architecture based on Inception v2 and v3's improvements took place to achieve music genre classification over the GTZAN [70], Ballroom [11], and Extended Ballroom [90] datasets. The proposed approach accentuates the role of multi-scale time-frequency information extraction from spectrograms to discriminate the genre of unknown music recordings [168]. Another approach leveraged the use of Residual-Inception blocks for music emotion classification. The proposed contribution takes Mel-spectrogram as input parameters and takes advantage of Inception's reduced complexity to design a music playback algorithm [189]. Interested in the field of music information retrieval, a network of 1D CNN with the Inception-GRU structure took place to extract features of different dimensions and perform music compression and decompression tasks for music emotion recognition purposes [197]. As for audio classification studies, an effort to test pre-trained ImageNet standard CNN models ( Inception, DenseNet [109], ResNet [88]) showed that they can achieve state-of-the-art results when tested over the UrbanSound8k and the GTZAN dataset. This approach relied on generated Mel-spectrograms out of the audio dataset as input features to the CNN networks [156]. An effort to generalise broadcast networks for music streaming services investigated many variants of broadcast networks including inception schemes. The experiments showed that such variants can efficiently localise temporal features for music classification use cases [184]. They achieved state-of-the-art results over the GTZAN [6], Free Music Archive (FMA) [85], HOMBURG [15], and Extended Ballroom [90] datasets. Intrigued by the huge amount of songs released on music streaming services, a study for music information retrieval took place by examining harmonic-percussion source separation, the Mel-spectrogram and the modulation spectrogram for the feature extraction stage, and different versions of an inception block for nonlinear features. The proposed model achieved the best accuracy in comparison to its related work [178].

## 2.6/ MUSIC GENRE CLASSIFICATION USING COMPUTER VISION

Our previous work and interests in creating the MEI2JSON [182] converter while relying on the proposed MusicPatternOWL [150] ontology helped us understand the importance and the benefits of ensuring a lossless data extraction, conversion, and processing solution for further usages. Our path of pursuing the studies over Eastern music scores was interrupted by the fact that this music genre lacked digital availability compared to occidental music, where a sufficient number of music scores is digitally ready to be used in experimental studies such as deep learning. At this stage, we were interested in experiencing the impact of deep learning on the music industry. Thus, it was decided to search for publicly available open-source occidental music datasets to pursue our first deep learning experiment, music genre classification.

Many studies took advantage of deep learning technologies to build efficient music genre classifiers. They adapted visual-related features (audio spectrogram) to build CNNs for audio classification tasks [105, 78, 97]. The audio data is converted to spectrograms and used as input features to CNN classifiers. These spectrograms are the visual representation of the spectrum of frequencies of the audio signal. In this thesis, the proposed contribution is validated through experimental results. These experiments are applied using both the GTZAN dataset [6] and the FMA dataset [85]. Thus, the most recent and relevant experiments on the two datasets are presented below.

Starting with GTZAN-related contributions, a framework achieved an accuracy of 93.7% over the GTZAN dataset by producing a multilinear subspace analysis. It reduced the dimension of cortical representations of music signals [35]. Further studies took profit from DNNs and CNNs to try to reach higher accuracies over music datasets. Inspired by multilingual techniques for automatic speech recognition, a multilingual DNN was used in [78] for music genre classification purposes. It was able to achieve an accuracy of 93.4% through 10-fold cross-validation over the GTZAN dataset. Several approaches used CNN-based networks but were not able to exceed the accuracy of 91% such as [105, 97, 102, 141]. Others tried refining their results by overcoming the blurry classification of certain genres inside the GTZAN dataset. Their study did not surpass the accuracies mentioned previously [128]. After several attempts to outperform the accuracy reached in [35], three literature studies succeeded in using Mel spectrograms as input features to their DNNs. The use of convolutional long-short term memory-based neural networks (CNN LSTM) in combination with a transfer learning model helped in achieving an accuracy of 94.20% in [124]. As for the two remaining academic efforts, the BBNN network proposed in [168] was able to achieve an accuracy of 93.90% by fully exploiting Mel spectrograms as a low-level feature for the music genre classification. The GIF generation method proposed in [172] was able to achieve the highest accuracy of 94.70% by providing efficient audio processing for animated GIF generation through acoustic features. Although this dataset

has several faults [70], it is still the most dataset used in music genre classification use cases. These faults are taken into consideration in the preprocessing process that we will develop in later sections. Concerning the FMA-related scholarly contributions, a method of vertically slicing STFT spectrograms took place, in addition to applying oversampling and under-sampling techniques for data augmentation purposes. This method achieved an F-score of 62.20% using an MLP classifier [138]. Another study trained a convolutional recurrent neural network (C-RNN) using raw audio to provide a real-time classification of FMA's music genres. It achieved an accuracy of 65.23% [147].

Motivated by FMA's challenges, an approach of two Deep Convolutional Neural Networks (DCNN) was proposed to classify music genres. The first DCNN was trained by the whole artist labels simultaneously, and the second was trained with a subset of the artist labels based on the artist's identity. This approach achieved an accuracy of 57.91% taking Mel spectrograms as input features to the DCNNs created [114]. Moreover, a method proposed in [141] took advantage of Densely Connected Convolutional Networks (DenseNet), found to be better than Residual Neural Networks (ResNet) in music classification studies. It achieved an accuracy of 68.20% over the small subset of FMA.

The extensive research on music genre classification, particularly on the GTZAN and FMA datasets, has provided valuable insights and findings that serve as a solid foundation for our upcoming discussion. In light of these research contributions, we are inspired to present a novel music genre classifier that builds on the insights we've gained and takes us a step further in improving music genre classification. Key factors such as the selection of appropriate visual-audio features and their pre-processing techniques, the use of a CNN network with densely connected layers, and the use of the powerful Inception network have emerged as critical findings. These findings will play a central role in guiding our exploration as we explore the details of this classifier in chapter 5.

## 2.7/ GUITAR TABLATURE ESTIMATION USING COMPUTER VISION

The experiments held in the music genre classification field [165] helped us understand the leverage of deep learning and especially computer vision in such techniques. However, the quest to make substantive contributions in this area has been challenging, largely due to the high level of commitment from prestigious research institutions. This led us to take a deliberate detour and shift our focus to identifying areas within deep learning where we could make distinctive contributions. Our investigation involved music generation, music composition and transcription, multi-instrument music separation, as well as many recent areas of interest until we reached the automatic music transcription field, especially tablature transcription of the string-based instruments (guitar, lute, vihuela, etc.).

Many studies are proposed for automatic tablature transcription, but only a few seek to detect the real fretting of the guitarist. One of the first approaches leverages the fundamentals and partials for candidate pitches to determine the most used string per pitch, I. Barbancho et al. (2012) [50]. This approach is limited to detecting no more than four pitches sounding simultaneous. Two years later, a system for applying the Blind Harmonic Adaptive Decomposition Algorithm was developed to classify several performance parameters, including the detection of the note's guitar string, implemented in Fuentes et al. (2012) [56]. This system is not evaluated for framewise tablature estimation. Nevertheless, it is considered an insightful approach for multi-pitch estimation and guitar tablature estimation.

Additionally, several studies focused on the guitar in their pursuit of automatic transcription. For instance, A. M. Barbancho et al. (2012) [49] transcribed guitar chords and fingering using a hidden Markov Model, while Humphrey and Bello (2014) [73] took the benefit of a convolutional neural network (CNN) model to achieve chord recognition.

The results of the latter approach encouraged the researchers to take advantage of CNN for similar music-related tasks. A combination of a CNN for framewise acoustic modelling and a recurrent neural network (RNN) model is proposed for piano transcription in Sigtia et al. (2016) [95].

The use of neural networks for music-related tasks helped in providing solutions for tablature arrangement problems (Tuohy et al. (2006) [18]). It tackled various music information retrieval tasks such as instrument classification in Gómez et al. (2018) [125] and Han et al. (2017) [107], music genre classification in El Achkar et al. (2021) [165], and singing voice detection in Schlüter and Lehner (2018) [134]. It also helped in achieving the first guitar tablature estimation model using CNNs. The model was trained using solo acoustic guitar performances of the GuitarSet dataset presented in Xi et al. (2019) [139], while outperforming state-of-the-art multi-pitch estimation algorithms. This paper also introduced a set of metrics found to be specific for evaluating guitar tablature estimation models, as described in Wiggins and Kim (2019) [146].

Several attempts took place to improve the TabCNN's results presented in Wiggins and Kim (2019) [146]. One of those attempts was the thesis report in Maaiveld et al. (2021) [170]. It yielded insights into the CNNs' functioning for automatic music transcription. The proposition relied on several adaptations such as data augmentation, Oracle method adaptation, and increasing the amount of training data. The latter study was not able to outperform the results of the TabCNN (Wiggins and Kim (2019) [146]) but presented intuitive conclusions, such as the fact that Dense layers play a major role in tablature estimation CNNs and that the size of the dataset is a key factor in the model's performance. The fast growth of neural networks encouraged researchers to test the latest approaches in the music industry. An unsupervised pitch estimation model was reported by Wiggins and Kim (2020) [158] to analyse audio clips by estimating their pitches and amplitudes.



The model was not tested through experiments but gave thoughtful ideas for further unsupervised acoustic guitar transcription attempts. Also, a method for generating note-level transcription for guitar transcription is proposed to demonstrate successful transcription using notes rather than frames [188]. This work outperformed the conventional frame-level CNN methods. Nevertheless, it did not outperform all TabCNN's estimation metrics results [146]. Last but not least, a unified model and methodology for estimating pitch contours took place to transcribe guitar tablatures [180]. It produced pitch estimates with a higher resolution than modern models. However, and to the best of our knowledge, neither the approaches listed in this section nor any other associated work can outperform all TabCNN's [146] estimation metrics for guitar tablature transcription.

The extensive research in the field of guitar tablature estimation, in particular on the GuitarSet dataset, has provided valuable insights that serve as a primary source of inspiration for our upcoming research. Key factors such as the selection of appropriate visual-audio features and the necessary pre-processing methods are at the core of our approach. We also investigate different networks, including CNN-based, Transformer-based and hybrid CNN-Transformer models. At the same time, we conduct a comprehensive review of state-of-the-art computer vision networks. These essential insights and discoveries will serve as the basis for our pursuit as we proceed to the in-depth investigation of this transcription solution in chapter 6.

## 2.8/ MUSIC DATASETS FOR EXPERIMENTAL STUDIES

In this section, we showcase some of the most used occidental and Middle Eastern open music datasets since our thesis concerned a private Middle Eastern dataset for traditional modal monodies for the first two contributions, in addition to publicly available occidental datasets for the last two contributions that leveraged deep learning capabilities. Before starting to enumerate the datasets, it is important to note that the TMM dataset used in the MEI2JSON contribution is a private dataset owned by Antonine University. It consists of a large number of music compositions archived in hard copy format. The common interest of musicians and engineers made it possible to encode 150 music compositions as music scores in MEI [8], which were used for experimental studies for our MEI2JSON converter. We list below the most used Middle Eastern and occidental music datasets, including the GTZAN, FMA, and GuitarSet datasets that we leverage in this thesis.

1. **AudioSet [106]** is a public dataset created to facilitate research in music applications such as audio tagging [123, 200], sound event detection [145, 136] and environmental sound classification [179]. This dataset contains over two million 10-second audio clips (27,778 hours). This large number makes it suitable for many

audio analysis studies. The AudioSet includes musical instruments, speech, natural sounds and various environmental sounds. Each audio clip is tagged with one or more audio event categories, allowing for fine-grained categorisation. The majority of the audio clips are extracted from YouTube videos. The clips are linked to their corresponding video to make additional contextual and metadata information more accessible. The dataset has been manually tagged for better assessment and is hosted in the cloud for easy access and download. The AudioSet contains a Middle Eastern music dataset consisting of 2088 video clips with 5.8 hours of audio data. This subset covers a vast region of countries stretching from Morocco to Iran.

2. **IRMAS [53]** ("Instrument Recognition in Musical Audio Signals") is a dataset of different musical instruments that can be used to analyse music from different cultures, including the Middle East. The dataset was created to address the challenges of instrument recognition [127] given an audio clip. It has been used in music information retrieval studies and recommendation systems to improve previous recognition solutions [129, 157]. IRMAS consists of short audio clips, each containing a single musical instrument. The clips are a few seconds long and are all sampled at a standard rate, as most deep learning use cases require normalisation of sampling rates. The dataset consists of 10 different instrument categories, in addition to the human voice as an 11th category. It includes variations in pitch, dynamics and playing style for each instrument category, making the recognition task more challenging and realistic. In total, IRMAS contains around 7000 audio clips. Although this size is relatively small compared to other datasets, it is one of the most valuable resources for evaluating contributions to instrument recognition.
3. **Million Song Dataset (MSD) [40]** is one of the largest and most diverse datasets in the field of Music Information Retrieval (MIR). The MSD dataset supports various research studies, including music recommendations for music streaming services [71, 80], genre classification [113, 131], mood analysis [98], tempo estimation [135], and more. It contains approximately one million songs, each song having an associated metadata file. This metadata is useful for recommendation tasks, where streaming services rely on artist information, song release dates, and other music-related data to suggest new music to users. The MSD dataset provides a set of audio features extracted from the audio recordings (timbral, rhythmic and harmonic descriptors) that can be used alongside the metadata information to perform general tasks. While the MSD dataset is not widely available to everyone, most of its subsets remain available for research purposes.
4. **Ballroom [11] and Extended Ballroom [90]** datasets consist of audio recordings that are often used for music genre classification studies [168, 160, 75]. The main purpose of the Ballroom dataset is to focus on dance music genres that are com-

monly played in ballroom dancing settings. This dataset consists of approximately 700 audio recordings, where each recording has its genre label. The genres include but are not limited to, waltz, tango, Viennese waltz, foxtrot, quickstep, etc. The researchers leveraged this dataset by extracting spectral and rhythmic descriptors from the recordings to solve the challenging variety of music played in ballroom dance styles. The Extended Ballroom dataset is an augmented version of the Ballroom dataset, which includes an extensive range of music genres and increases the number of audio clips to approximately 4180. The extended version was adopted to benchmark music genre classification studies that have previously been evaluated on large datasets, to better assess the performance of the work.

5. **Free Music Archive [85]** (FMA) is a large tagged dataset for music-related research and analysis. Similar to MSD and AudioSet, it has a metadata file associated with each audio track. Each metadata file contains the artist name, track title, genre labels and release date. FMA contains tens of thousands of audio tracks covering a wide range of popular music genres such as rock, hip-hop, jazz, electronic and more. It is often used for music genre classification and music genre recommendation use cases [130, 167]. The majority of tracks in the FMA can be used for research and creative purposes without violating copyright restrictions. The FMA dataset consists of three subsets: FMA Small, FMA Medium and FMA Large. These subsets differ in the number of tracks and their audio quality, allowing researchers to choose the most appropriate subset for their study.
6. **GTZAN [6]** is a dataset created for benchmarking music genre classification and audio analysis [153, 25]. This dataset contains 1000 audio tracks of 30 seconds each. These audio tracks are divided into ten popular music genres. These genres include rock, blues, jazz, reggae, hip-hop, country, classical, pop, disco and metal, and thus have common genre similarities with the FMA dataset. The GTZAN dataset is manually tagged with the aforementioned genres to provide ground truth information for deep learning use cases such as genre classification. This dataset is also used for feature extraction and analysis, helping to identify discriminative audio features for the development of music recommendation systems.
7. **GuitarSet [139]** is an audio dataset focusing on guitar-related audio analysis and various areas related to stringed instrument playing, chord recognition, finger positions (tablature notations) and strumming patterns. Thus, this dataset supports the training and evaluation of deep learning models for guitar-related use cases [146, 155, 148]. This dataset covers different playing styles and techniques of the guitar instrument. Each audio recording is annotated with chord labels and tablature labels. The chord labels indicate the guitar chords played at a particular time interval in the audio, and the tablature labels indicate the finger positions on the guitar

fretboard when each chord is played. Both annotations are great features for chord recognition tasks and tablature recognition tasks for research studies. The GuitarSet includes a strumming annotation, which provides another area of research for identifying and improving strumming patterns and techniques. All of the above annotations are stored in JAMS [74] files as JSON annotations for reproducible MIR research.

The datasets presented above do not represent all the efforts in creating Middle Eastern and Occidental music datasets. However, they are the most widely used by researchers due to their availability, the quality of the recordings, and fewer copyright restrictions for free and open-access use. It is important to note that the majority of these datasets are manually labelled or verified by humans, especially musicians, thus providing a reliable background for researchers to build real music applications and replicate real scenarios while training and evaluating their solutions with authentic music datasets.

## 2.9/ CONCLUSION

This chapter has guided us on an extensive journey through the landscape of related research in two key areas: Music Encoding and Deep Learning for Music Applications. By diving into these areas, we've uncovered the preceding studies that formed the foundation for our contributions. From fundamental research on Eastern music encoding using the MEI format and the creation of the MusicPatternOWL ontology and the MEI2JSON converter, to the exploration of visualised audio features and their integration into advanced Deep Learning models, this review has given us a comprehensive view of the existing knowledge that informs this study. We have bridged the gap between music encoding and deep learning, demonstrating how traditional music representation is integrated with state-of-the-art Deep Learning techniques. Our review of studies using visualised audio features, convolutional neural networks (CNNs), transformer networks and hybrid models has highlighted the evolution of music analysis and classification. Moreover, we have looked at specific applications such as music genre classification and guitar tablature estimation, providing insights into the latest methods and influential studies in these areas. We've also emphasised the importance of relevant music datasets, including those that played a key role in our experiments. This comprehensive review of related work not only enhances our understanding of the research landscape but also serves as a solid foundation upon which the subsequent chapters of this thesis are built.





KNOWLEDGE EXTRACTION AND FORMAT  
CONVERSION OF MUSIC SCORES



## SUPPORTING MUSIC PATTERN RETRIEVAL AND ANALYSIS

Analyzing music notations is found useful for musicology purposes. This can be applied by retrieving semantic information from digitally annotated music scores. In this chapter, we propose an ontology that structures the knowledge extraction process of a music pattern analysis algorithm. In addition to mandatory elements that describe music scores, the proposed ontology relies on contextual elements and attributes for pattern analysis. The ontology then supports the semantic information retrieval and analysis processes of music score contents. We illustrate the whole mechanism by explaining the workflow of the ontology integrated inside a music encoding platform for eastern music.

### 3.1/ INTRODUCTION

New music scores are being constantly composed by musicians, and many of them are encoded for analysis purposes through XML formats such as MusicXML [7] or MEI [8]. The analysis of a music score consists of extracting its underlying features. Therefore, many ontologies are proposed to structure music score content for better information retrieval [100, 110].

Researchers have developed text-based platforms to archive and restore music scores. These platforms store music scores based on their meta-data. Thus, users can search for many scores based on the composer's name, the date of the publication, the title, etc. However, there are other important features for musicians, known as semantic features, that are not covered by many platforms such as the number of instruments in the music score, the tonality in which it was written, as well as the number and the order of occurrence of several specific notes inside a music score. All these features and many more hold essential information, that, when organized, constitute a criterion for music scores.

In this chapter, we present a new ontology named MusicPatternOWL that aims to explore



the advantages of the Web Ontology Language (OWL) on encoding and annotating music scores, based on their pattern analysis. The ontology is based on the schema structure of the MEI format.

The remainder of this chapter is organized as follows: In Section 2, we introduce the MusicPatternOWL ontology, describing its main goals, restrictions, and structural aspects. Section 3, explores the use of the proposed ontology in a musical analysis platform, followed by a conclusion in Section 4.

## 3.2/ MUSICPATTERNOWL

In this section, the MusicPatternOWL ontology is detailed. This ontology structures music score elements for music pattern retrieval and analysis. The main goal of MusicPatternOWL, its restrictions and structural features are explained below.

### 3.2.1/ GOALS AND RESTRICTIONS

Numerous studies have engaged in developing and managing ontologies related to music score content. Jones et al. [110] introduced an ontology tailored for semantically annotating and reasoning upon Western music scores. Their research primarily focused on investigating the benefits of using the web ontology language (OWL) in various music-related applications. Similarly, Cherfi et al. [100] proposed an ontology that integrate semantic music elements for facilitating the extraction and management of knowledge from music data. Their work contributed to standardizing the representation of music theories and enabling interconnectedness between different musical elements. However, it's important to note that neither study explicitly addressed the support for Eastern music within their ontological frameworks, thus neglecting essential elements such as scales, modes, rhythmic patterns, and instruments unique to Eastern traditions.

Ontologies provide the semantic means to represent any data formats. They improve data integration and data-driven analytics for structured and unstructured data. The proposed ontology serves for many music-oriented web services relying on pattern analysis. It accentuates the role of Semantic Web in knowledge extraction platforms. The platform developed in [119] extracts underlying features of a music score using a pattern analysis algorithm. It serves as knowledge extractor for traditional modal monodies of the Middle East and the Mediterranean cultures ( including medieval European monodic music, as well as Mashriq and Maghreb traditions). Our ontology provides a structured separation of music score elements that helps the platform in the knowledge extraction process. Therefore, the MusicPatternOWL ontology focuses only on the music score and excludes meta-data features. Its structural properties are based on the MEI schema [8], adding to

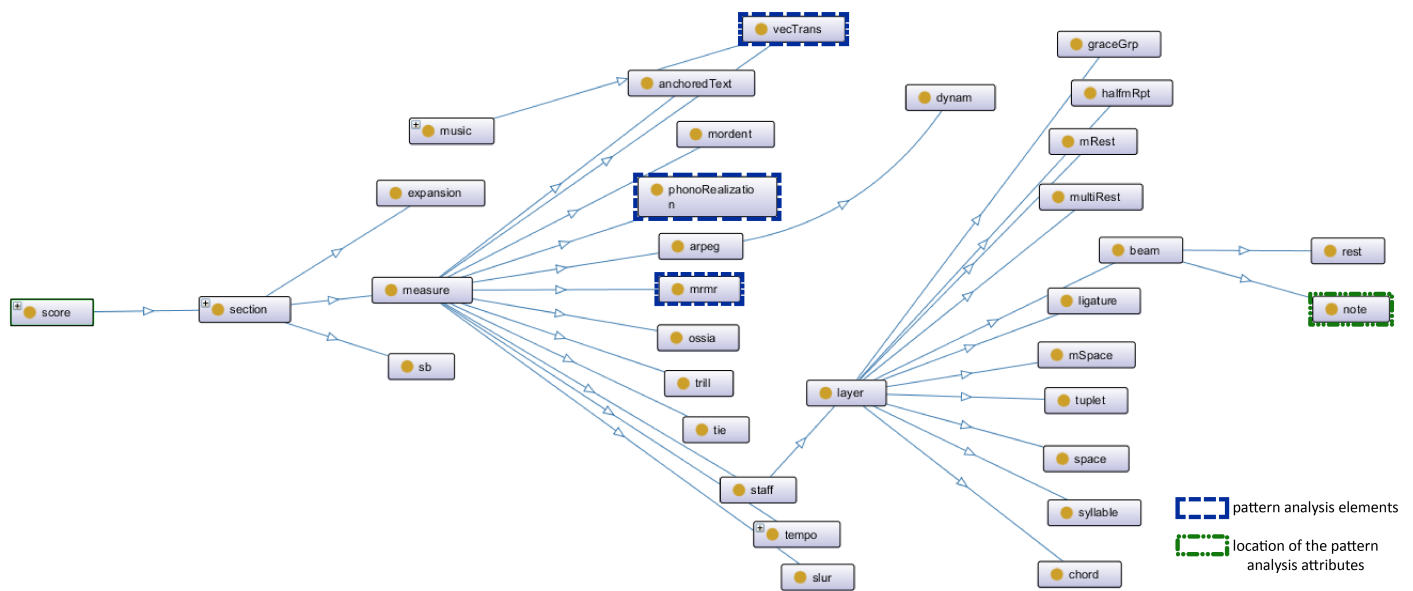


Figure 3.1: MusicPatternOWL - General Overview.

that several contextualized elements and attributes for oriental music analysis.

### 3.2.2/ STRUCTURAL ASPECTS

Based on the MEI schema, the MusicPatternOWL ontology shares the same contribution proposed in [119]. It proposes new elements and attributes to provide a structured form for analyzing the pattern of notes inside a music score (see Figure 3.1).

Before citing the elements and the attributes related to pattern analysis, it is important to mention that a single music score contains many measures, where each measure is composed of multiple notes.

**snr** (Syllabic Nuclear Reduction) Attribute for the note element. It contains only two values  $\alpha$  or  $\beta$ . The pattern analysis algorithm developed in [119] considers the final note of a music score as a main input element. Then, for each note of the music score,  $\alpha$  or  $\beta$  values are assigned. According to a phonological component of the theory developed in [91],  $\alpha$  represents a primary/basic note and of  $\beta$  represents a secondary note. Thus, at the level of the music score containing many measures of notes, we derive a pattern formed of  $\alpha$  and  $\beta$  values.

**mnr** (Metasyllabic Nuclear Reduction) Attribute for the note element. It contains only boolean values. This attribute presents a rhythmic parameter (morphological rewriting), where a value of "true" is assigned to a note containing a **snr** attribute and has the highest duration among adjacent notes. A value of "false" is assigned when one of the

previous conditions is not applied. The highest duration is calculated through a matrix of notes provided by the musicians. Thus, using different matrices for a single music score generates different pattern analysis possibilities.

The Figure 3.2 presents the first measure of a music score. It is encoded in MEI format [8] after applying the analysis (3.2.a), and rendered to SVG (3.2.b), placing **snr** and **mnr** values above notes respectively. **snr** with a value of "\alpha" in (a) will be given a value of  $\alpha$  in (b), and "\beta" a value of  $\beta$  respectively. **mnr**s with a value of "true" will hold the same value of **snr** in its current note. **mnr**s with a value of "false" are left without labels. As presented in the MEI preview in (3.2.a), the **snr** of the third note is assigned a value of "\alpha" and the **mnr** a value of "true"; this is interpreted by a value of  $\alpha$  for the **mnr** when rendered to SVG in (3.2.b). It is important to mention that notes that are neither primary nor secondary will not be assigned any **snr** and **mnr** values. Therefore the second note in Figure 3.2 is not labeled.


MEI	SVG
<pre> ... &lt;measure&gt;   ...   &lt;beam xml:id="b1"&gt;     &lt;note xml:id="n1" ... snr="\alpha" mnr="true"/&gt;     &lt;note xml:id="n2" ... /&gt;   &lt;/beam&gt;   &lt;beam xml:id="b2"&gt;     &lt;note xml:id="n3" ... snr="\alpha" mnr="true"/&gt;     &lt;note xml:id="n4" ... snr="\beta" mnr="false"/&gt;   &lt;/beam&gt;   ... &lt;/measure&gt; ... </pre>	<div style="text-align: center;"> <p><b>mnr</b> :    <math>\alpha</math>    <math>\boxed{\alpha}</math></p> <p><b>snr</b> :    <math>\boxed{\alpha}</math>    <math>\alpha</math>    <math>\beta</math></p>  </div>
(a) Score encoded in MEI	(b) Score rendered to SVG

Figure 3.2: A sample of a music score encoded in MEI (a) then rendered to SVG (b) throughout the analysis.

**mrmr** (Morphophonological Rhythmic and Melodic Rewriting) Child of the measure element. It contains matrices and mathematical equations. This element holds different patterns extracted from the music score, based on the melodic attribute (**snr**) and the rhythmic attribute (**mnr**) already presented.

**phonoRealization** (Phonological Realization) Child of the measure element. This element holds matrices and equations like the **mrmr** element, but encodes only the underlying phonological features of a music score.

**vecTrans** (Vector Transcoding) Child of the measure element and the music element.

This element contains generated vectors formed from combining **snr** and **mnr** attributes. The result is a vector at the level of a measure and a series of vectors at the level of the entire music score (music element). The generation of vectors is considered the last step of the algorithm proposed in [119], providing meaningful knowledge extraction for the musicians.

In addition, we note the **number** attribute for the **mrmr**, **phonoRealization** and **vecTrans** elements. It serves as an identifier for each measure of a music score.

The remaining elements of the MusicPatternOWL shown in Figure 3.1 constitute typical elements of a music score. Based on the MEI schema developed in [8], our ontology achieves full coverage of any element and attribute needed to digitally encode or annotate music scores in MEI format. It is important to mention that the elements and attributes discussed in this section are optional, as we aim to extend this ontology to cover western music's pattern analysis.

## 3.3/ PROOF OF CONCEPT

### 3.3.1/ SCORE ANALYSES

The music encoding algorithm proposed in [119] is used as a starting point to create the MusicPatternOWL. In short, the algorithm has been integrated into a music encoding platform for the traditional modal monodies of the Mashreq called "MM analyzer". The platform expects as input an MEI document and two matrices mentioned earlier in the section 2 above. It outputs another MEI document rendered with all the corresponding analysis in a PDF file using Verovio [76] and SVG processing for placing alphas and betas above notes.

Analyzing music scores exposes the underlying features by assigning  $\alpha$  and  $\beta$  values to the **snr** attribute. **snrs** of a music score will be grouped in SNR, and **mnr**s with a value of "true" will be assigned the same **snr** value of their current note and grouped in MNR. The process above creates a pattern of  $\alpha$  and  $\beta$  grouped in SNR and afterwards in MNR for each music score (see Fig. 3.3). Therefore, the proposed ontology keeps track of the analysis to provide a structured knowledge extraction in each progressive step of the encoding. It covers the analysis from the step of pattern establishment to the generation of the vectors through its elements and attributes already presented. Also, it will restrict any false or abnormal insertion of any musical notation through its elements, handling not only the analysis itself but the structure and the regulations of the music score. It is important to note that the entire analysis process is expressed in terms of mathematical expressions. This enhances the need to have a rule-based ontology to validate all related



Figure 3.3: SVG output from MM analyzer.

properties for error-less platform behavior.

### 3.3.2/ THEME QUERIES

It was previously mentioned that the proposed ontology structures the information retrieval for music pattern analysis. Following that, this section comes to show several examples of search criteria and their correspondent SPARQL Queries. The following prefixes are used for the queries below:

rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

owl: <http://www.w3.org/2002/07/owl#>

rdfs: <http://www.w3.org/2000/01/rdf-schema#>

xsd: <http://www.w3.org/2001/XMLSchema#>

mpos: <https://github.com/elachkarcharbel/MPOWL/blob/master/mpowl.owl#>

**Query 1:** The following query searches the mandatory attributes of each note. The query will result in a gathering of all the information of all notes which are essential to start the analysis. Also, the counter of the notes is added to easily index the final note of a music score - the necessary criteria to initiate the pattern analysis algorithm [119].

```
SELECT ?measure ?note ?duration ?octave ?pitchname ?dots
(COUNT(?note)AS noteSum)
WHERE {
  ?score mpos:hasPerformersParts ?part .
  ?part mpos:isSinglePartContainer ?section .
  ?section mpos:isSingleMeasureContainer ?measure .
  ?measure mpos:isNote ?note .
```

```

?note mpo:hasPname ?pitchname .
?note mpo:hasDur ?duration .
?note mpo:hasOct ?octave .
OPTIONAL{ ?note mpo:hasDots ?dots}
} GROUP BY ?measure ?note ?dots ?duration ?octave
?pitchname

```

**Query 2:** Once the information of the note are gathered, the platform assigns **snr** values. The query below extracts the duration of all the notes and their **snr** values so that the pattern analysis algorithm receives needed information to achieve the next step of the analysis, which is the assignment of **mnr** values.

```

SELECT ?measure ?note ?duration ?snr
WHERE {
  ?score mpo:hasPerformersParts ?part .
  ?part mpo:isSinglePartContainer ?section .
  ?section mpo:isSingleMeasureContainer ?measure .
  ?measure mpo:isNote ?note .
    ?note mpo:hasDur ?duration .
    OPTIONAL{ ?note mpo:hasSNR ?snr}
} GROUP BY ?measure ?note ?duration ?snr

```

**Query 3:** It shows a retrieval example of the total amount of notes and explores the values of **snr** and **mnr** at the level of each specific note inside a music score. This query serves for the last step of the pattern analysis algorithm, where **snr** and **mnr** values will be gathered to form **vecTrans**'s generated vectors.

```

SELECT ?score ?measure ?note ?snr ?mnr
  (COUNT(?note) AS ?noteSum)
WHERE {
  ?score mpo:hasPerformersParts ?part .
  ?part mpo:isSinglePartContainer ?section .
  ?section mpo:isSingleMeasureContainer ?measure .
  ?measure mpo:isNote ?note .
    OPTIONAL{ ?note mpo:hasSNR ?snr.
              ?note mpo:hasMNR ?mnr}
} GROUP BY ?score ?measure ?note ?snr ?mnr

```

It is important to mention that the platform will provoke multiple inserts to the Music-PatternOWL: generated vectors are stored in the **vecTrans** element and mathematical

equations are stored in **mrmr** and **phonoRealization** elements, based on their musical role respectively.

### 3.4/ CONCLUSION

In this chapter, we proposed the MusicPatternOWL ontology that covers the structural and behavioral aspects of a pattern analysis algorithm for encoding Eastern music scores. As explained, the proposed ontology structures the entire music score in addition to its pattern analysis, to achieve information retrieval and analysis of music score content. The ontology has the potential for future expansion to accommodate different pattern analysis theories in the field of music. It already covers essential elements and attributes for music scores, both in Western and Eastern music compositions. Additionally, it is designed to be scalable, allowing for the introduction of new optional elements and attributes, as outlined in this study. This scalability simplifies the process of incorporating similar pattern analysis components for more extensive music analysis algorithms. However, it's important to note that when adding new elements to the ontology, caution must be exercised to prevent potential complications to the existing elements. These complications might relate to increased complexity and potential performance issues (decrease in query speed, lower computational efficiency) when overscaling the size of the ontology.

# MUSIC SCORE PRE-PROCESSING AND CONVERSION

Converting music score content from symbolic formats to simplified data formats is found useful for artificial intelligence purposes. The conversion can be applied using XSL stylesheets and ontologies to ensure the preserving of the data quality throughout the transformation. In this chapter, we proposed a new converter capable of transforming music scores encoded in MEI to JSON format for pre-processing purposes, and future usage into artificial intelligence techniques. The proposed converter uses an eastern music score ontology capable of structuring standard music scores content in addition to elements and attributes specific to eastern music. Thus, the converter shares the same support for eastern music scores. We illustrate the conversion process by assessing the performance analysis, the data quality, and the storage of the proposed converter in comparison with a combined approach composed of two state-of-the-art converters.

## 4.1/ INTRODUCTION

Combining artificial intelligence (AI) techniques with software solutions was found interesting for researchers and developers in the recent decade. The usage of AI helped in providing digital assistance as well as handling repetitive jobs for employees in their daily tasks. It helped with digital platforms where the need to reduce errors is one of the most essential and challenging criteria to improve its performance and reliability. Studies went deeper until they reached music-related interests. Many researchers and musicians took benefit of AI in their music-related studies. The latter provided digital assistance in music annotation platforms, such as predicting the next note of a real-time annotated music score or generating new music scores depending on a pre-defined dataset. However, both, the prediction of the next note and the generation of an entire music score require a well-defined pre-processing process to prevent data loss and reach higher accuracy post-training. This process and especially in music-related fields consist of applying sev-



eral progressive tasks, such as finding the needed elements and attributes of a music score, filter the music score upon the use case, and finally, reshape the data and convert it to a specific format for training ingestion. A platform for encoding and analysing eastern music scores named traditional modal monodies encoder (MM analyser) in Asmar et al. (2018) [119] was capable of encoding a corpus specific to modal monodies of the Mashreq. The MusicPatternOWL ontology proposed in [165] assisted in the analysis process of the encoder, ensuring errorless export of eastern music scores encoded in MEI format. The results of both, the encoder and the ontology, encouraged the use of resultant music scores in machine learning use cases, by the fact that they provide ready-to-ingest eastern music scores in MEI format. While gathering the music scores out of the MM analyser, it was found that the MEI format is not the optimal format used to feed AI models. MEI is an XML-based format that holds multiple elements, each element gathers multiple music-related attributes to encode detailed music score content. This is where we highlight the need to convert the MEI outputs to simplified formats to reach our target of applying AI techniques on music score content. Based on a related work investigation, we found that the MEI format can be converted to multiple formats such as MIDI and MusicXML. The latter formats were similar to the MEI in the matter of providing simplified data to the AI models. MusicXML is also an XML-based format and MIDI represents only recorded and played audio information. Further investigations led us to discover the MusicJSON format proposed in Alvaro and Barros (2010) [30] capable of converting MusicXML music scores to JSON. JSON is an easy-to-use data ingestion format over XML. Its improved readability and lightweight approach support a bigger amount of information for feeding the AI models.

The MEI to MusicXML converters use the MEI encoding tools (<https://github.com/music-encoding/musicencoding>) provided by the MEI community for applying MEI conversions. These tools lack encoding and representing eastern music scores elements and attributes. Thus, the usage of existing MEI to MusicXML converters at the first stage, and the conversion of the resultant MusicXML outputs to JSON format at a second stage, generate JSON data that does not support eastern music score content. In this chapter, we present a new data converter named MEI2JSON that aims to convert the music scores encoded in MEI to JSON format while preserving their eastern music score content. The converter is based on the MusicPatternOWL ontology [165], in addition to a modified schema of MEI proposed in Asmar et al. (2018) [119] capable of providing a structured knowledge extraction of music scores elements and attributes for eastern music encoded in MEI. The MEI2JSON is also capable of providing an MEI to JSON conversion without the need to combine multiple converters from multiple sources. The remainder of this chapter is organized as follows: In Section 2, we introduce the MEI2JSON converter, describing its main components, their behaviour, and the role of the MusicPatternOWL ontology inside these components. Section 3 explores the full implementation of the pro-

posed converter through its application on an eastern music scores dataset encoded in MEI. In Section 4, we compare through experiments the proposed converter with a combination of two existent converters, followed by a conclusion in Section 5.

## 4.2/ THE MEI2JSON CONVERTER

### 4.2.1/ MOTIVATION

The MM analyser in Asmar et al. (2018) [119] and the *MusicPatternOWL* ontology [165] treated one of the most primary problems in music-related platforms. The latter is the lack of support for eastern music encoding and analysis. The MM analyser helped in encoding and analysing eastern music scores, and the *MusicPatternOWL* assisted in that analysis process by ensuring an errorless knowledge extraction at each progressive step of the encoding. At this stage, we were able to export lossless music scores encoded in MEI format.

The advantages of combining AI techniques with music-related platforms (presented in Section 1) motivated us to integrate those techniques and improve the MM analyser. Similar to any AI use case, the data must be prepared and simplified as much as possible before its training ingestion in neural networks. Therefore, it was needed to convert our MEI exports to another data format by the fact that MEI holds many elements and attributes that can be reduced upon the use case. Based on the music-related converters presented in Sections 1 and the Related Work chapter, the absence of a converter capable of transforming MEI music scores into JSON format was noticed, in addition to one of the essential criteria in question: converting music scores without losing data quality and preventing errors.

All the reasons mentioned above motivated us to create the MEI2JSON converter capable of transforming MEI music score to a simplified JSON format while preserving data quality and reducing data manipulation errors, especially for eastern music score datasets.

### 4.2.2/ MEI2JSON COMPONENTS

By definition, “MEI is a community-driven, open-source effort to define a system for encoding musical documents in a machine-readable structure.” Its schema is developed using a literal programming XML format and expressed using the Relax NG (RNG) schema language. This music representation format and other primary ones focus on supporting occidental music because of its major worldwide usage. Therefore, it is not accurate for encoding eastern music scores as mentioned in Asmar et al. (2018) [119] and El Achkar and Atéchan (2020) [165]. As stated in Sections 1 and the related work chapter, our need

is to obtain the most simplified format out of eastern music scores encoded in MEI for future AI usage. The absence of a converter capable of handling eastern music elements and attribute at a first step, and the disability to convert MEI music scores to a simplified format using a single converter at a second, led us to create the MEI2JSON converter. The MEI2JSON converter consists of three main components. Each component is responsible for a specific task to achieve successful MEI to JSON music scores conversion. As illustrated in Figure 4.1, any MEI in question should enter the MEI2XML component at a first phase, redirect the result of the first component to the XML2RDF component at a second phase, and at last convert the RDF data to JSON through the RDF2JSON component.

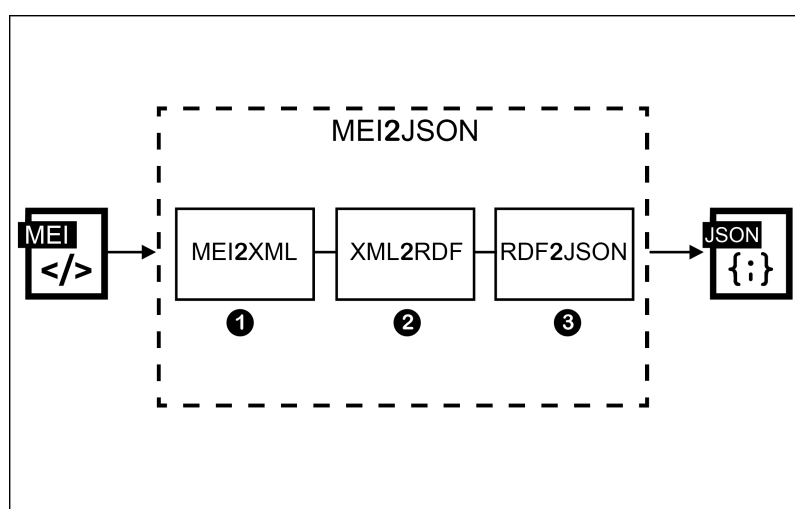


Figure 4.1: MEI2JSON Main Components Overview

#### 4.2.2.1/ THE MEI2XML COMPONENT

As mentioned earlier, MEI is an XML-based format expressed using the RNG schema language. Thus, the MEI2XML component re-structures the schema of the MEI, producing a simplified XML output for the second component. Inside the MEI2XML, we use the modified MEI schema proposed in Asmar et al. (2018) [119] to keep track of all the schema structure proposed by the MEI community, in addition to the elements and attributes proposed in Abou Mrad (2016) [91] and contributed to the MEI schema in Asmar et al. (2018) [119]. These elements and attributes are essential when analysing and encoding eastern music scores. For this purpose, we configured an XSL stylesheet to embed our MEI to XML transformation rules. These rules hold structuring aspects to make the XML output the simplest possible. The proposed method consists of converting the RNG schema of MEI to a legacy XML schema. This consists of transforming the attributes of the MEI elements to sub-elements of the element itself and filters the concluding in the most optimal way possible.

After running the XSL stylesheet over several MEI files, we found that the modified MEI schema and the custom rules configured, lack consistency and normalisation. The configuration of custom rules resulted in different forms of output for a single MEI music score. Thus, we perceived the need to replace our custom rules with the *MusicPatternOWL* ontology [165]. The *MusicPatternOWL* contains all the rules and restrictions needed to structure a music score encoded in MEI. It supports the same elements and attributes existent in the modified MEI schema, in addition to its power to extract and preserve the semantic information in an errorless manner.

Based on the XML to OWL frameworks mentioned in the Related Work chapter, the mapping between XML schemas and OWL schemas can be build using two different approaches. In case the OWL schema is existent, the XML and OWL schemas should be mapped manually, and in case the OWL schema does not exist, the OWL schema can be generated out of the existent XML schema, and by that, obtain an automatic mapping between them. Both, manual and automatic mapping approaches are used to transform the XML instances into OWL individuals. On the other side, our approach was not to transform MEI to OWL individuals directly but to transform them to XML instances with OWL rules included, to structure and filter the needed data and exclude irrelevant ones. Since the *MusicPatternOWL* is inspired by the MEI schema and shares the same contribution as the modified MEI schema proposed in Asmar et al. (2018) [119], a half-way mapping was already established. As for the MEI2XML component, we completed this mapping process by configuring an XSL stylesheet holding all the necessary mapping and transformation rules to convert an MEI music score into a simplified XML format.

The mapping rules are classified into three distinct types:

- Class mapping
- Datatype property mapping
- Object property mapping

The class mapping concerns creating a link between a node of the modified MEI schema with an OWL concept of the *MusicPatternOWL* ontology. The datatype property mapping links an MEI node to a datatype property of the *MusicPatternOWL*. The object property mapping relates two-class mappings to an OWL object property of the *MusicPatternOWL*. As for the transformation rules, in addition to the ones embedded through mapping, we note the re-structuring shown in the XML representation below, to obtain the optimal XML output possible. The first representation is a measure of an MEI score entered as input, and the second one is the same measure converted to the XML output through the configured XSL stylesheet. Since the *MusicPatternOWL* ontology excludes meta-data features, the latter is excluded from the XML output generated, due to the absence of mapping between the MEI schema and the *MusicPatternOWL* for this purpose. This feature is

found essential by the fact that it can automatically filter irrelevant data, focusing on the music-score itself to achieve a successful pre-processing process.

At this stage, the MEI2XML component relies on an XSL stylesheet capable of transforming MEI scores to XML format, while preserving music elements and attributes specific to eastern music.

---

The measure representation in an MEI file

```
<measure xml:id="m-32" label="1" left="rptstart" n="1">
  <staff xml:id="m-34" n="1">
    <layer xml:id="m-35" n="1">
      <beam xml:id="m-37">
        <note xml:id="m-36" dur="8" dur.ges="128p" oct="4" pname="d" pnum="50"
          stem.dir="up" snr="\alpha"/>
        <note xml:id="m-38" dur="8" dur.ges="128p" oct="4" pname="g" pnum="55"
          stem.dir="up" snr="\beta" mnr="yes"/>
        <note xml:id="m-40" dur="8" dur.ges="128p" oct="4" pname="f" pnum="54"
          stem.dir="up">
          <accid xml:id="m-41" accid="s"/>
        </note>
      </beam>
    </layer>
  </staff>
  <tie xml:id="m-39" endid="#m-40" startid="#m-38"/>
</measure>
```

---

The measure representation in the XML output

```
<measure number="1">
  <beam>
    <note>
      <pname>d</pname>
      <oct>4</oct>
      <snr>\alpha</snr>
      <dur>8</dur>
    </note>
    <note>
      <pname>g</pname>
      <oct>4</oct>
      <snr>\beta</snr>
      <mnr>\beta</mnr>
      <dur>8</dur>
    </note>
  </beam>
</measure>
```

```
<note>
  <pname>f</pname>
  <oct>4</oct>
  <dur>8</dur>
  <accid>s</accid>
</note>
</beam>
</measure>
```

---

#### 4.2.2.2/ THE XML2RDF COMPONENT

The usage of XSL stylesheet in the MEI to XML conversion of the first component encouraged us to take the same approach in the next one. The objective of the XML2RDF component is to convert the XML data into RDF without losing any semantic information. Therefore, we decided to use the XSL stylesheet proposed in Breitling (2009) [29]. The latter contains all the standard transformation rules capable of providing efficient XML to RDF conversion. In other terms, we can apply this converter to any XML dialect which supports then both, the elements and attributes proposed in the modified MEI schema in Asmar et al. (2018) [119]. At this stage, we were able to convert MEI scores to XML using MEI2XML and convert the XML to RDF using the XML2RDF component, without losing any semantic information related to eastern music scores.

Note that mentioning the support of eastern music scores does not eliminate the fact that music encoding formats were initially built to support occidental music scores. Therefore, the MEI2JSON converter supports the latter if encoded in MEI format.

#### 4.2.2.3/ THE RDF2JSON COMPONENT

The previous components of the MEI2JSON converter managed to convert MEI scores to RDF using several methods to prevent loss of semantic information and data quality. Thus, the job of the RDF2JSON component is to proceed with the conversion process to convert the music score encoded in MEI to JSON format. As mentioned in Section 1, the MEI2JSON converter aims to transform MEI files to JSON for pre-processing purposes. The pre-processing process, in addition to data cleaning and filtering, consists of applying feature engineering selection to choose the needed input variables for training ingestion. In music related cases, these input variables are the elements and attributes of a music score, where we must select the needed ones only, to solve targeted use cases. As an example, when the use case is to predict the next note of a music score, we must select the input variables (elements and attributes) that affect only the note element of a mu-

music score. Therefore, we use the *MusicPatternOWL* ontology [165] as element selector and validator in the RDF2JSON component. This way we can produce the most optimal JSON output by selecting the needed elements from the music score upon the use case, validating once again the three mapping types of the MEI2XML component, and finally building the JSON output to achieve a successful pre-processing process.

It is important to mention that this component excludes the attributes of a music score since the MEI2XML component transforms the attributes to sub-elements as shown in the earlier XML representation. Also, the RDF2JSON component contains a knowledge graph builder so that using SPARQL queries the *MusicPatternOWL* is capable of selecting the needed features through a simple query builder. Note that the query result is passed through JSON libraries to ensure the creation and validity of the output.

To recapitulate and converge, the MEI2JSON relies on three components. The first component, the MEI2XML, converts the structure of an MEI music score to XML by transforming its schema represented in RNG to the XML schema. Its conversion relies on a mapping between the modified MEI schema presented in Asmar et al. (2018) [119] and the *MusicPatternOWL* [165]. This mapping is implemented in an XSL stylesheet, the core of the MEI to XML conversion. The second component, the XML2RDF, uses the existing XSL stylesheet proposed in Breitling (2009) [29] for converting XML to RDF. Since this stylesheet supports any XML dialect, we only configured this component to reach the RDF format for input in the last component. The third and last component, the RDF2JSON component uses the *MusicPatternOWL* as a music score validator ensuring a lossless flow of information. It converts RDF data to JSON while implementing the idea of query builder where users can filter and retrieve their needed music elements upon future AI use cases. Therefore, the unification of these components constitutes the MEI2JSON capable of transforming eastern music scores to a ready-to-ingest format in AI models.

### 4.3/ IMPLEMENTATION

The previous section presented each component of the MEI2JSON converter. It exposed the role, the composition, and the benefit of each component to achieve a successful conversion of music scores encoded in MEI to JSON output. The present section exposes the necessary technical details to achieve the full implementation of the proposed converter, in addition to the implementation of two combined converters for further experimental comparison.

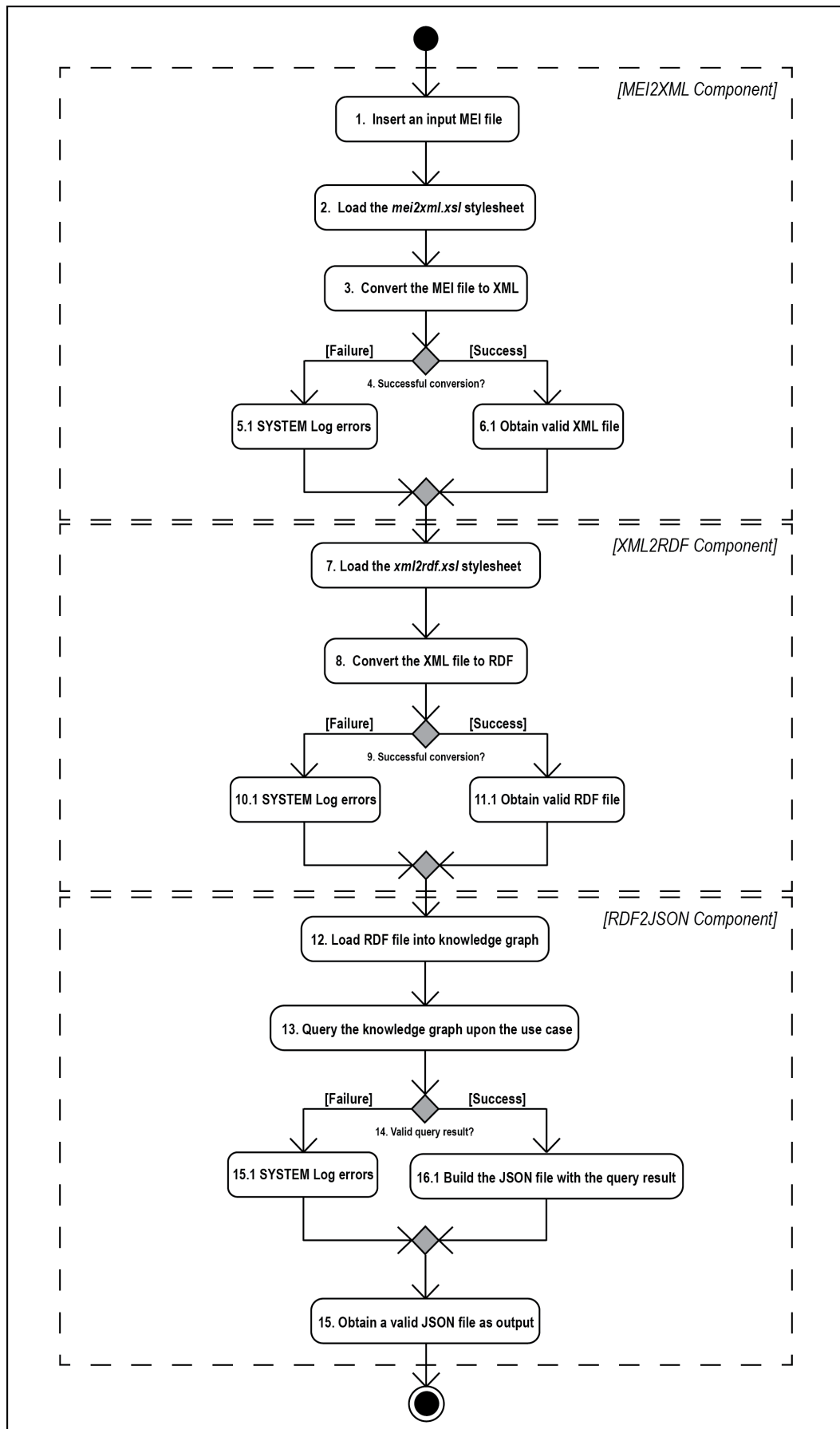


Figure 4.2: MEI2JSON Activity Diagram



### 4.3.1/ MEI2JSON PROCESS

Figure 4.2 presents the MEI2JSON converter through an activity diagram. The first part of the diagram illustrates the progressive steps of the MEI2XML component to achieve successful conversion (MEI to XML). The converter pulls an MEI score taken from an MEI file (.mei extension), loads the custom XSL stylesheet created and converts if possible, the MEI file to XML. The custom XSL stylesheet named *mei2xml.xsl* is the file responsible for handling the conversion needed. Thus, the *mei2xml.xsl* needs to be loaded by an XSLT processor to perform this conversion. Therefore, we use the Saxon XSLT and XQuery processor (Kay, 2010) [12] based on its previous usage in most of the converters presented in Section 2. In the case of a successful conversion, the generated XML file will be redirected to the second component to perform further steps. Otherwise, the system logs the errors so that we can easily find and solve the problems related to the failure in conversion.

The generated XML file proceeds its path to the XML2RDF component. Like the previous component, the *xml2rdf.xsl* proposed in Breitling (2009) [29] is loaded using the Saxon processor to apply the corresponding conversion to the XML file. Also, the generated RDF file proceeds to the next component in success cases, and in case of failure, the error loggings will guide the user to solve the problems faced.

Finally, the generated RDF file is loaded in the RDF2JSON component using the library (RDFLib, <https://github.com/RDFLib/rdfLib>). This library provides powerful parsers and serialisers to load the knowledge graph out of RDF/XML data. Once loaded, the RDF file can be queried through custom SPARQL queries to extract the semantic information needed for pre-processing purposes. The SPARQL query then can be customised upon the use case. In case of a successful query, the result will be sorted and formed in a JSON file as output. The RDF2JSON component ensures the validity of the JSON file by applying schema syntax definitions such as the JSON schema proposed in Pezoa et al. (2016) [94]. Thus, the MEI2JSON made several progressive steps passing from a component to another, to achieve a successful conversion of MEI scores to JSON.

Note that the MEI2JSON converter is currently implemented using the Python language (Van Rossum and Drake, 2009) [38], although, it can be implemented using other programming languages since we are loading the XSL stylesheet through command-line usage of the Saxon library. Also, the RDF related libraries are available in many programming languages which helps in providing enhanced coverage of the MEI2JSON converter.

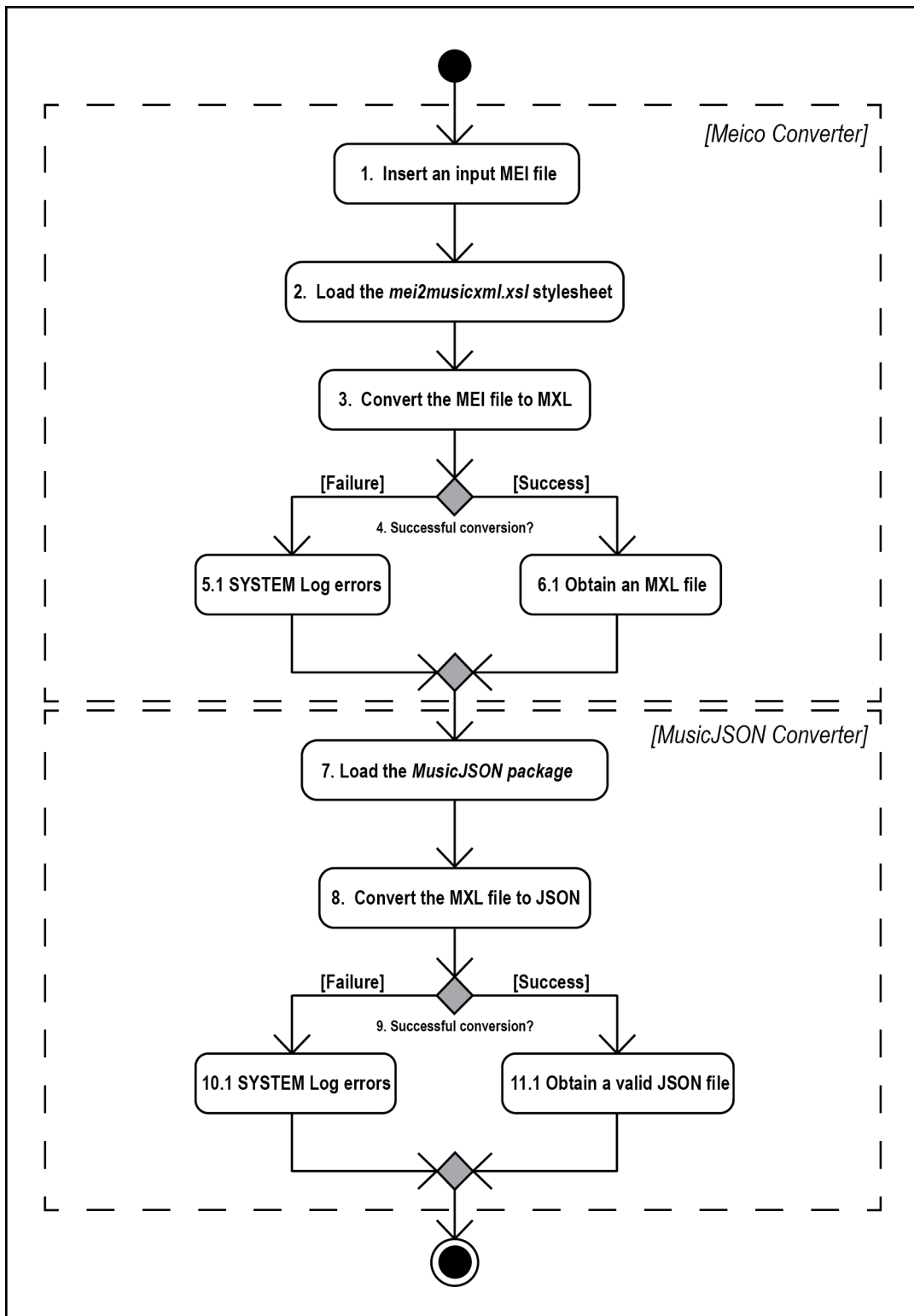


Figure 4.3: Meico + MusicJSON Activity Diagram

### 4.3.2/ *Meico* + *MusicJSON* PROCESS

The related work presented all the converters capable of transforming MEI scores to other symbolic formats such as MusicXML or MIDI. Also, it was mentioned that the music community does not have a straightforward approach to convert MEI scores to JSON. In this part, we present the usage of two different converters so that once implemented, they can be used in combination to assess the MEI2JSON converter. The first converter named *Meico* can convert MEI scores to MusicXML, and the second one, the *MusicJSON* converter can convert MusicXML scores to JSON format.

Figure 4.3 presents the combined converters through an activity diagram. The first portion of this diagram concerns the insertion of the MEI score as input to the *Meico* converter, loading the `mei2musicxml` stylesheet provided by MEI encoding tools (<https://github.com/music-encoding/music-encoding>), and converting the MEI score to MusicXML format. In case of a successful conversion, the obtained MusicXML file proceeds to the second converter. Otherwise, the system logs the errors found while converting to detect and solve related problems. The second portion of this diagram concerns the insertion of the MusicXML file generated by the *Meico* converter and converting this file to JSON format using *MusicJSON* proposed in Alvaro and Barros (2010) [30]. In successful cases, the result will be a valid JSON output that respects the schema proposed by the *MusicJSON* contributors.

It is important to mention that both, the *Meico* (Berndt et al., 2018) [121] and Verovio (Pugin et al., 2014) [76] converters rely on the same XSL stylesheet provided by MEI encoding tools (<https://github.com/musicencoding/music-encoding>) to run the transformation over MEI scores and convert them to MusicXML. Also, they use the same approach of using the command-line interface to apply this conversion which makes them identical in this matter. Therefore, using *Meico* with *MusicJSON* as the combined approach or using Verovio with *MusicJSON* will result in the same experimental results in terms of data quality and complexity metrics. Note that the *MusicJSON* converter consists of a package written in JavaScript programming language, loaded using the Node.js runtime environment to achieve the corresponding conversion. This package, in addition to the whole process, is called using Python programming language (Van Rossum and Drake, 2009) [38] for better comparison with the proposed converter.

## 4.4/ EXPERIMENTS

In the implementation section, we presented both the MEI2JSON converter and the two combined converters *Meico+MusicJSON*. We elaborated the two processes using activity diagrams to technically describe the role of each component inside both approaches. In

In this section, we aim to compare the MEI2JSON with the *Meico+MusicJSON* in terms of performance analysis and the quality of the data produced out of these converters. For this purpose, we use a dataset of 150 traditional modal monodies music scores encoded in MEI. These music scores are considered a unique corpus in eastern music. Also, these MEI scores are the output of the MM analyser, the platform proposed in Asmar et al. (2018) [119] to analyse and encode eastern music. Thus, the music scores contain elements and attributes specific to eastern music analysis, in addition to the standard elements and attributes present in any music score encoded in MEI.

#### 4.4.1/ DATASET

Considering that our primary objective is to provide successful MEI to JSON conversion of eastern music scores, we chose a dataset related to traditional modal monodies. Modal monodies are eastern music scores, thus the dataset used in our experiments contains the following four eastern music modes:

- Hijāz (31 music scores)
- Dūlab Bāyāti (33 music scores)
- Dūlab Rāst (40 music scores)
- Jāhārkā (46 music scores)

Once grouped, we obtain 150 eastern music scores encoded in MEI format. The size of an MEI score varies between 4.4 to 42.6 kB of music score data. Musicians transcribe modal monodies from eastern music score books such as Abou Mrad (2016) [119] to MEI format. They encode and validate their digitalized transcriptions using the MM analyser, and provide us with the needed MEI scores for further studies.

#### 4.4.2/ PERFORMANCE ANALYSIS

The performance analysis concerns analysing algorithms based on an input size required to run it. The complexity then is expressed as a function of  $n$ , where  $n$  is the input size. In this chapter, we compare the MEI2JSON with the *Meico+MusicJSON* in terms of two complexity metrics, the time and space complexity. The time complexity describes the amount of time to run an algorithm, and space complexity reports the amount of memory space to run an algorithm. We calculated the time complexity through experimental evaluation, and used the *memory-profiler* python module to evaluate the space complexity for both approaches.

## 4.4.2.1/ TIME COMPLEXITY

Figure 4.4 presents the time complexity chart for the MEI2JSON and the *Meico+MusicJSON* converters. We use an orange-dashed line to visualize the MEI2JSON converter, and a blue line to illustrate the *Meico+MusicJSON* converter. The two approaches use the same number of input (150 music scores) and the same number of elementary operations performed by the algorithm for better comparison purposes. Figure 4.4 shows the same performance of both approaches when the input size is smaller than 40 music scores. However, the time complexity changes clearly after reaching a value of 60 music scores, taking a different trajectory for each approach.

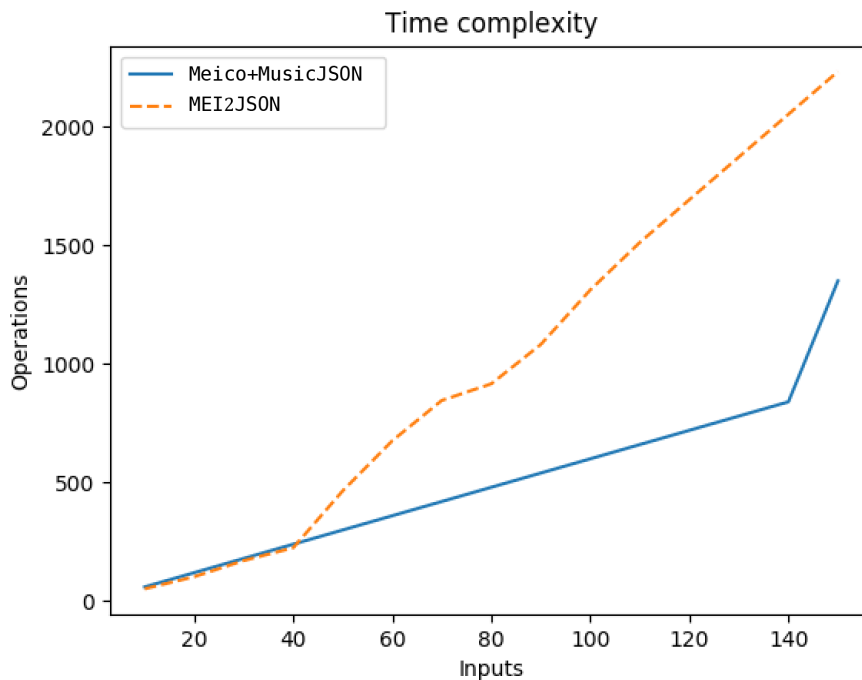


Figure 4.4: Time Complexity Chart

The algorithm of the *Meico+MusicJSON* is not distinctly affected by the size of each music score. On the other hand, the proposed MEI2JSON converter lacks this stability due to his final component, the RDF2JSON component. The usage of the RDF format for data selection and second validation through the *MusicPatternOWL* forces the algorithm to load the entire RDF file inside the knowledge graph of the RDF2JSON component. The latter slows the converter dependent on the size of the RDF file in question, the time to load the RDF inside the knowledge graph, and the selection of the needed elements using SPARQL queries.

Therefore, the *Meico+MusicJSON* outperforms the MEI2JSON in terms of time complexity due to the dependency of the latter on the size of each music score. Note that the

number of operations of the *Meico+MusicJSON* augmented quickly after exceeding a value of 140 music scores as input size. This unpredicted augmentation could result in a draw between the two approaches when using bigger datasets.

#### 4.4.2.2/ SPACE COMPLEXITY

Regarding the space complexity, we use the *memory-profiler* python module to monitor the memory consumption of both the *Meico+MusicJSON* and the MEI2JSON converters. Figures 4.5 and 4.6 present the memory consumption (in MiB) expressed as a function of time to respectively estimate the space complexity of *Meico+MusicJSON* and MEI2JSON. The space complexity in both graphs is calculated by running the *memory-profiler* on the algorithms using the entire dataset of 150 music scores.

In Figure 4.5, we can visualise an approximate constant line with a value of 48.0 MiB during the whole process. We interpret the chart by the fact that the Meico converter start by loading the *mei2musicxml.xsl* stylesheet to convert the MEI score to MusicXML. This loading allocates an amount of memory equal to 48.0 MiB until the conversion completes. Once completed, the *MusicJSON* converter allocates an amount of 48.7 MiB to load its package and convert the MusicXML scores to JSON format (from 310 to 330 millisecond in Figure 4.5). The entire process took 330 milliseconds to convert the 150 music scores to JSON.

As for Figure 4.6, we can visualise two main variations of memory consumption. The first is a continuous line taking a value of 48.0 MiB from the beginning till a time equal to 380 milliseconds. The second is an approximate line to 72.0 MiB from 380 milliseconds to the end (630 milliseconds). The first line stable on 48.0 MiB concerns loading the custom *mei2xml.xsl* stylesheet responsible of converting the MEI scores to XML format.

Once the first conversion completes, the same memory consumption is given to load the *xml2rdf.xsl* stylesheet responsible for converting the XML format to RDF. Both stylesheets use the processor proposed in Kay (2010) [12], which explains the fact that they have the same memory consumption (from 0 to 380 milliseconds). Therefore, the MEI2XML and XML2RDF components allocate the same amount of memory. Once the role of the XML2RDF component completes, the line chart varies to reach a value stable on 72.0 MiB approximatively, to highlight the third component, the RDF2JSON. The load of the RDF results using the proposed library (RDFLib, <https://github.com/RDFLib/rdfliib>) is responsible for reaching this memory consumption value. This library handles the loading of RDFs into the knowledge graph at first, and querying the needed elements out of the RDF file loaded in second. The entire process took 630 milliseconds to convert the 150 music scores to JSON.

Note that MEI2JSON's last component is the part responsible for increasing the time and memory consumption in comparison with the *Meico+MusicJSON* approach. Therefore,

the RDF2JSON component is responsible for increasing the space complexity and the time complexity as seen in the previous interpretations. Improving this component in the future can make the proposed converter outperform the two combined ones in terms of time and space complexity.

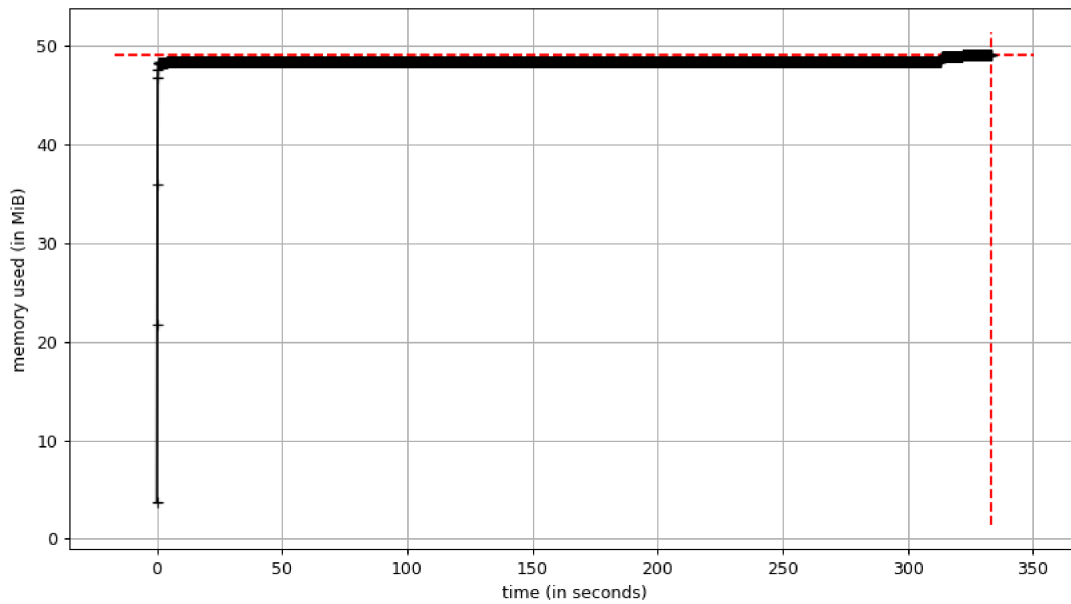


Figure 4.5: Space Complexity Chart - Meico+MusicJSON

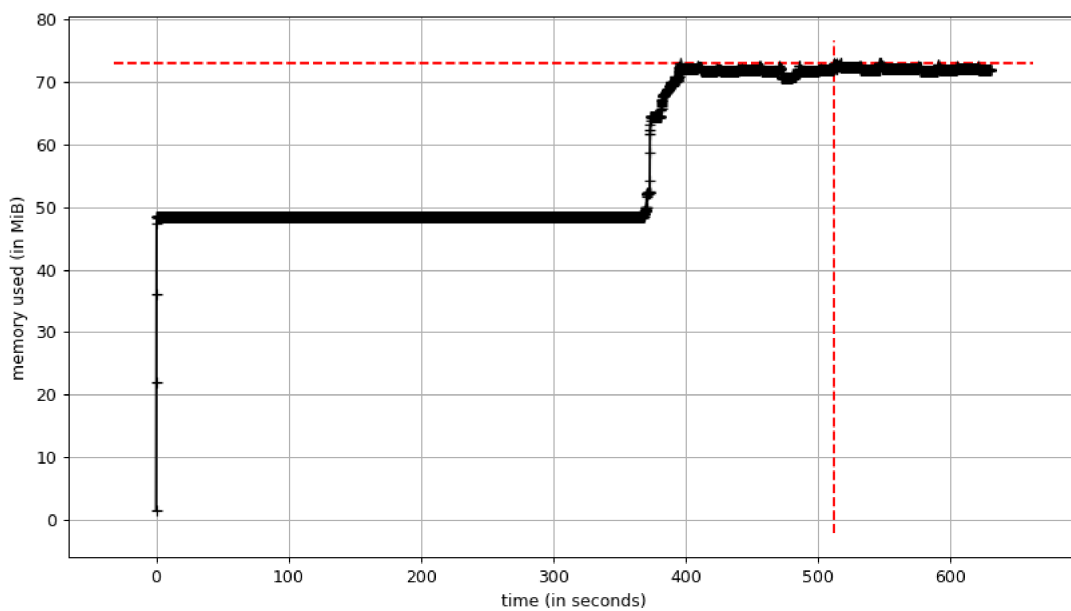


Figure 4.6: Space Complexity Chart - MEI2JSON

#### 4.4.3/ DATA QUALITY ASSESSMENT

The performance of converters is usually evaluated by calculating its complexity and ensuring it preserves the quality of the data produced out of its transformation. In Figure 4.7, we present four different data quality metrics used on both the *Meico+MusicJSON* and the MEI2JSON converters to assess their quality preserving upon the used dataset. The MEI2JSON metrics are visualised using orange bars in the histogram, and the *Meico+MusicJSON*'s using blue bars. Before explaining the quality metrics, we note the usage of the jsonix (<https://github.com/highsource/jsonix>) mapping library to obtain a JSON schema out of the modified MEI schema proposed in Asmar et al. (2018) [119]. Thus, we used the resulted JSON schema and the *MusicJSON* schema to respectively evaluate the output of the *Meico+MusicJSON* and the MEI2JSON converters upon each metric. It is valuable to mention that the mandatory elements in this experiment concern the note element and its attributes, including the eastern music score ones.

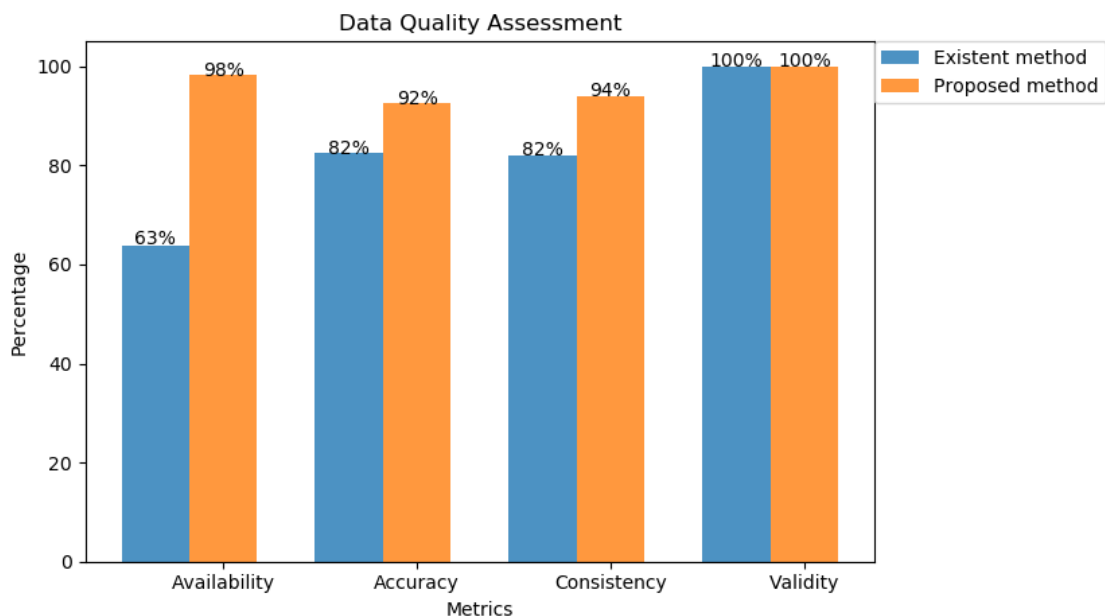


Figure 4.7: Histogram - Data Quality Metrics

Below we present the assessment metrics to compare both approaches:

- *Availability* is a metric used to measure whether all the necessary elements of a music score are present in a specific dataset. The dataset in this matter concerns the output generated out of both converters. We measure the availability of both approaches by calculating the percentage of music score fields that have values entered into them. The MEI2JSON resulted in an availability percentage of 98.2% and the *Meico+MusicJSON* of 63.9%. The gap between both results is mainly due



to the lack of support of the modified MEI schema in the *Meico+MusicJSON* approach. This lack is responsible for discarding undefined elements and preserving only the standard occidental music score elements. In this case, the undefined elements are the ones related to eastern music scores.

- *Accuracy* is a metric used to evaluate the correctness of the music score in question. We measure the accuracy of both approaches by calculating the percentage of the correctly converted music score elements compared to the initial values. Since the future usage of the converted music scores is in the AI field, we chose to estimate accuracy using the *accuracy\_score* function provided by Pedregosa et al (2011) [47]. Considering the usage of several essential elements for encoding music scores, we estimate the accuracy as a multilabel approach. We calculate the accuracy at the level of multiple elements, each element containing multiple labels. As shown in the equation below, the accuracy is the sum of accuracies at the level of each music element divided by  $N$  - the number of music score elements used. The *true\_label* are the labels of the initial music elements before conversion and the *output\_label* are the labels of elements after JSON or *MusicJSON* conversion. It is important to mention that since music score elements have labels encoded as characters in MEI, we had to use the *LabelEncoder* function provided by Pedregosa et al. (2011) [47] to convert characters to numeric. This way the element's labels would be compatible for usage in the *accuracy\_score* function.

For example, and before using the *accuracy\_score* function, we use the *LabelEncoder* to transform the labels of the PitchName element from [a, b, c, d, e, f, g] to [0, 1, 2, 3, 4, 5, 6] so that we can calculate the accuracy of the PitchName using *accuracy\_score* at first, calculate the accuracy of the remaining elements, sum all the accuracies and divide them by the number of elements used. Finally, we multiply the resulting accuracy by 100 to obtain the accuracy value as a percentage.

$$Accuracy = \frac{\sum accuracy\_score(true\_label, output\_label)}{N} \times 100$$

The *Meico+MusicJSON* resulted in an accuracy of 82.4%, and the MEI2JSON an accuracy of 92.5%.

- *Consistency* is a metric used to evaluate the synchronicity of music score in terms of data types and schema structure. We measure this metric by calculating the percentage of data types that match across different records. We use the schema structure of both approaches to detect the structure and data types changes while passing from a component/converter to another.

Similar to accuracy calculation, we took the same approach to calculate the con-

sistency at the level of data types per music score element. Therefore, we use the *LabelEncoder* function to transform data type values to numeric labels and use the *v\_measure\_score* function provided in Pedregosa et al. (2011) [47] to estimate consistency of music score elements.

By definition the *v\_measure\_score* clusters the labels given a ground truth. In our case, it clusters the element's data type labels reflecting the consistency of the latter elements. The *v\_measure\_score* function takes as parameters the following: The *true\_label* which stands for the element's data type labels before using the MEI to JSON or *MusicJSON* converters. The *output\_label* which stands for the element's data type labels after MEI to JSON or *MusicJSON* conversion. The last parameter,  $\beta$ , is the ratio of weight attributed to homogeneity and completeness. We will leave this value to its default meaning that the resultant score should have the same weight regarding homogeneity and completeness. The *v\_measure\_score* results in a score between 0.0 and 1.0. The greater the result, the better is the consistency.

Once the *v\_measure\_score* is calculated against all the essential elements of a music score, we sum all the resultant scores, divide them by  $N$  - the number of music score element used. Finally, we multiply the whole result by 100 to obtain the final consistency value as a percentage.

$$\text{Consistency} = \frac{\sum v\_measure\_score(true\_label, output\_label, \beta)}{N} \times 100$$

As shown in Figure 4.7, the consistency percentage of *Meico+MusicJSON* is equal to 82% and the MEI2JSON equal to 94%. This slight improvement of the MEI2JSON over the combined approach is due to the existence of the *MusicPatternOWL* ontology present in the first and last converter, to structure and filter the music score elements in question.

- *Validity* is a metric used to measure how well data conforms to the required value attributes. We measured the validity by calculating the percentage of music score elements and attributes that have values within the domain of acceptable values. We used the JSON schema of both approaches, in addition to the syntax definition proposed in Pezoa et al. (2016) [94] to calculate the validity metric. Both approaches resulted in a validity percentage of 100%. This high percentage is due to *MusicJSON's* built-in validator in the first approach and the presence of the *MusicPatternOWL* ontology in the second.

## 4.5/ STORAGE ASSESSMENT

The previous assessment reflected the outperforming of the MEI2JSON converter over the *Meico+MusicJSON* in terms of data quality metrics. The current assessment presents the storage reduction of both approaches at different input size scales.

Numbers of Inputs (Files)		10	20	30	40	50	60		
Initial input size (in kB)		58.9	115.3	208.1	267.5	546.2	861.4		
<i>Meico+MusicJSON</i> output size (in kB)		58.5	113.9	191.4	249.5	398.0	542.6		
MEI2JSON output size (in kB)		55.9	100.7	146.9	181.4	224.7	259.1		
70	80	90	100	110	120	130	140	150	Reduction(%)
1100	1200	1500	1900	2200	2500	2800	3100	3300	—
658.7	704.3	829.9	1000	1200	1300	1400	1500	1700	45.3
305.2	325.4	346.5	414.3	450.2	497.0	542.6	595.6	645.8	74.1

Table 4.1: Storage Overview Table.

In Table 4.1, we demonstrate the storage allocation of both conversion approaches using the same input size scale as the complexity study. The first row of the table presents the different scales of input sizes. The second corresponds to the initial size of the MEI music scores before conversion. The third and the last concern the output sizes of the *Meico+MusicJSON* and the MEI2JSON converter. The input and output sizes expressed in kiloBytes (kB). While assessing the storage, it was clear that the *Meico+MusicJSON* reduced the storage allocation depending on the input size. Therefore, we calculated the average reduction percentage that resulted in a decrease of 45.3%. On the other side, the MEI2JSON was able to reduce the storage with an average of 74.1%. Thus, the MEI2JSON is capable of reducing the storage by 28.8% more than the *Meico+MusicJSON* approach. This improvement is beneficial for database systems where it can ensure the integrity of music score using one of the most optimal format possible, the JSON format. Also, it has a positive influence on our pre-processing target, since ingesting smaller inputs makes the neural network handle bigger datasets while keeping the hardware in healthy conditions.

To briefly summarise our experiments, we calculated the time and space complexity, the data quality metrics, and the storage assessment of both converters using an eastern music dataset presented at the beginning of Section 5. The *Meico+MusicJSON* approach outperformed MEI2JSON in terms of time and space complexity. The latter outperformed *Meico+MusicJSON* in terms of data quality assessment and storage assessment. Thus, the MEI2JSON proved its role as a converter for eastern music scores from MEI to JSON format where other converters focused primarily on supporting occidental music scores.

On the other hand, the comparison between the previously mentioned converters will normally differ using occidental music scores. The existing solution should outperform the MEI2JSON in terms of data quality metrics since occidental music scores are in continuous development in the MEI community. These continuous updates are supported by the MEI converters like *Meico* (Berndt et al., 2018) [121] since they use the latest versions of the MEI encoding tools (<https://github.com/musicencoding/music-encoding>). However, the MEI2JSON relies on the *MusicPatternOWL* [165] that supports eastern music scores more than the occidental because of the latter's continuous updates.

## 4.6/ CONCLUSION

In this chapter, we proposed the MEI2JSON converter that covers the transformation of music scores encoded in MEI to JSON format for pre-processing purposes. As explained, the proposed converter consists of three components. The components rely on the *MusicPatternOWL* ontology to achieve information retrieval and structure music score content throughout the conversion process. We compared the MEI2JSON with a combined approach composed of two existent converters, *Meico* and *MusicJSON*. We used a dataset of 150 eastern music scores encoded in MEI to obtain the needed results. The experiment results were promising by the fact that our converter was able to outperform the combined converters in terms of data quality and storage assessment. The converter proved its capability of preserving the quality of the data while reducing the allocated storage space. However, the combined approach still outperforms the MEI2JSON in terms of analysis performance. The outperformance was mainly due to the behaviour of the last component, the RDF2JSON component.



# IV

## COMPUTER VISION IN THE MUSIC INDUSTRY



# MUSIC GENRE CLASSIFICATION USING COMPUTER VISION

Embedding music genre classifiers in music recommendation systems offers a satisfying user experience. It predicts music tracks depending on the user's taste in music. In this chapter, we propose a preprocessing approach for generating STFT spectrograms and upgrades to a CNN-based music classifier named Bottom-up Broadcast Neural Network (BBNN). These upgrades concern the expansion of the number of inception and dense blocks, as well as the enhancement of the inception block through reduction block implementation. The proposed approach is able to outperform state-of-the-art music genre classifiers in terms of accuracy scores. It achieves an accuracy of 97.51% and 74.39% over the GTZAN and the FMA dataset respectively. Code is available at <https://github.com/elachkarcharbel/music-genre-classifier>.

## 5.1/ INTRODUCTION

Modern studies found interest in building robust music classifiers to automate genre classification of unlabeled music tracks. There were diverse approaches in their feature engineering process as well as the neural network selection [105, 168, 128, 78, 172]. In this chapter, we propose a custom approach for music genre classification. STFT spectrograms are generated and diversified by slicing each spectrogram into multiple slices to ensure a variety of visual representations among the same music track. Furthermore, upgrades to a state-of-the-art Convolutional Neural Network (CNN) network for music genre classification named BBNN [168] are proposed. The contribution of this chapter relies on two main improvements: expanding the number of inception and dense blocks of the network and enhancing the inception block by implementing the reduction block B proposed in [116] instead of the existing block inspired by [81]. The proposition is evaluated through its application using the GTZAN [6] and the FMA [85] music datasets.



The remainder of this chapter is organized as follows: in Section 2, we present the pre-processing process in addition to the contributed upgrades. Section 3 explores the experimental results of the proposed upgrades over competitive CNN networks, followed by a conclusion in Section 4.

## 5.2/ PROPOSED APPROACH

In this section, the BBNN network proposed in [168] is briefly introduced. Later, the proposed approach is elaborated while mentioning the proposed upgrades to achieve higher accuracy results against the GTZAN and the FMA dataset. As mentioned in the related work, the Bottom-up Broadcast Neural Network (BBNN) is a recent CNN architecture that fully exploits the low-level features of a spectrogram. It takes the multi scale time-frequency information transferring suitable semantic features for the decision-making layers [168]. The BBNN network consists of inception blocks interconnected through dense blocks. The inception block is inspired by the inception v1 module proposed in [81] while adding a Batch Normalization (BN) operation and a Rectified Linear Unit activation (ReLU) before each convolution. This approach relied on generating coloured Mel spectrograms from the music tracks while providing the latter as input features to the CNN network. The spectrograms had the size of  $647 \times 128$  and were used as-is for training purposes. This network was able to achieve the second-best accuracy over the GTZAN dataset (93.90%) by stacking three inception blocks with their corresponding dense connections.

### 5.2.1/ PREPROCESSING

Spectrograms are the key to successful music genre classification using CNN-based networks. Based on the approaches mentioned in the Related Work chapter, greyscale STFT spectrograms are adopted instead of coloured Mel spectrograms. The majority of CNN based music genre classifiers relied on Mel spectrograms, since STFT spectrograms required greater GPU memory for their increased quantity of embedded features. Thus, we use STFT spectrograms in our experiments to leverage the latter increase on accuracy scores, in addition to the availability of efficient GPUs for experimental testing. Using the Sound eXchange (SOX) package, the greyscale spectrograms are generated with a size of  $600 \times 128$ . As expressed in the Related Work, the GTZAN dataset has several faults [70]. For instance, three audio tracks were discarded while recursively generating the spectrograms using the SOX package. Each music track of the discarded ones was associated with a separate genre of the dataset. Therefore, we randomly removed a single audio track from the remaining genres to normalize the number of music tracks per genre.

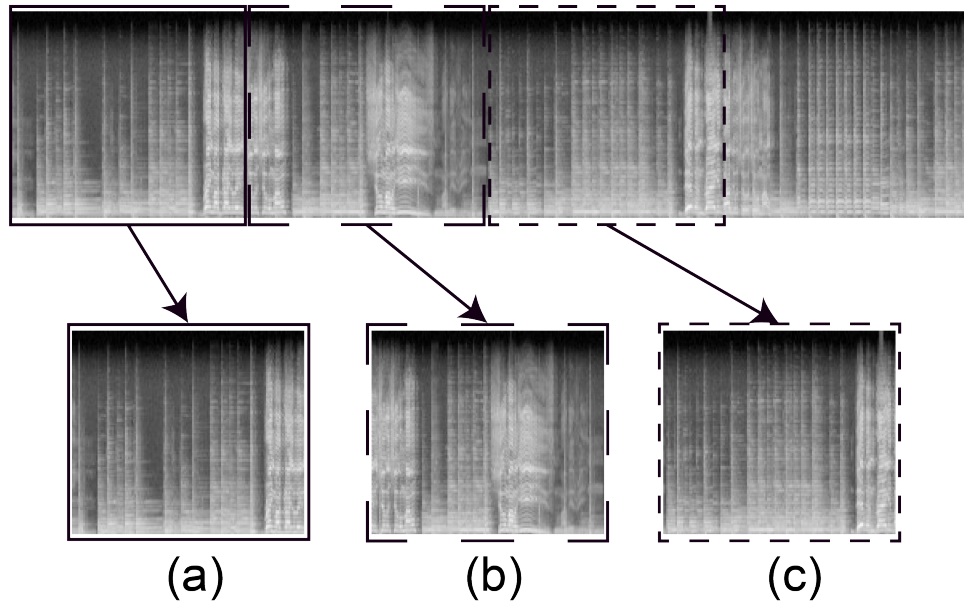


Figure 5.1: Spectrogram slicing approach

Subsequently, the Python Imaging Library (PIL) is used to slice the STFT spectrograms into multiple images. The spectrogram is divided into three to four separated slices. Each slice is a normalized  $128 \times 128$  slice that represents a 6.4 seconds (s) track out of the initial 30 s music tracks. Therefore, the last one and a half slices of the spectrogram are discarded, keeping only the first three slices (a, b and c in Figure 5.1). This approach is mainly used for better data preparation for CNNs by normalizing the spectrogram's width and height. It also increases the diversity of the music genres, since spectrograms variate dependently on the time axis. Thus, this normalization does not accentuate overfitting due to the variety in every spectrogram's slices. It is important to mention that the discarded slices may hold useful data for our classification. However, we adopted this approach to limit the number of training/testing images as well as ensuring the obtention of the same number of slices per music track (music tracks length is not always consistent to 30 s).

### 5.2.2/ NETWORK CONTRIBUTION

Inspired by the BBNN network [168], custom modifications are proposed to achieve higher accuracy results. Even though the BBNN stacks three inception blocks connected with dense blocks, the trained model possessed a tiny size (only 0.18 M). Using a small sample of both datasets, we performed a hyperparameter search taking the number of inception and dense blocks as the hyperparameter in question. The search result showed that the optimal number of blocks is equal to 6 for achieving the greatest accuracy.

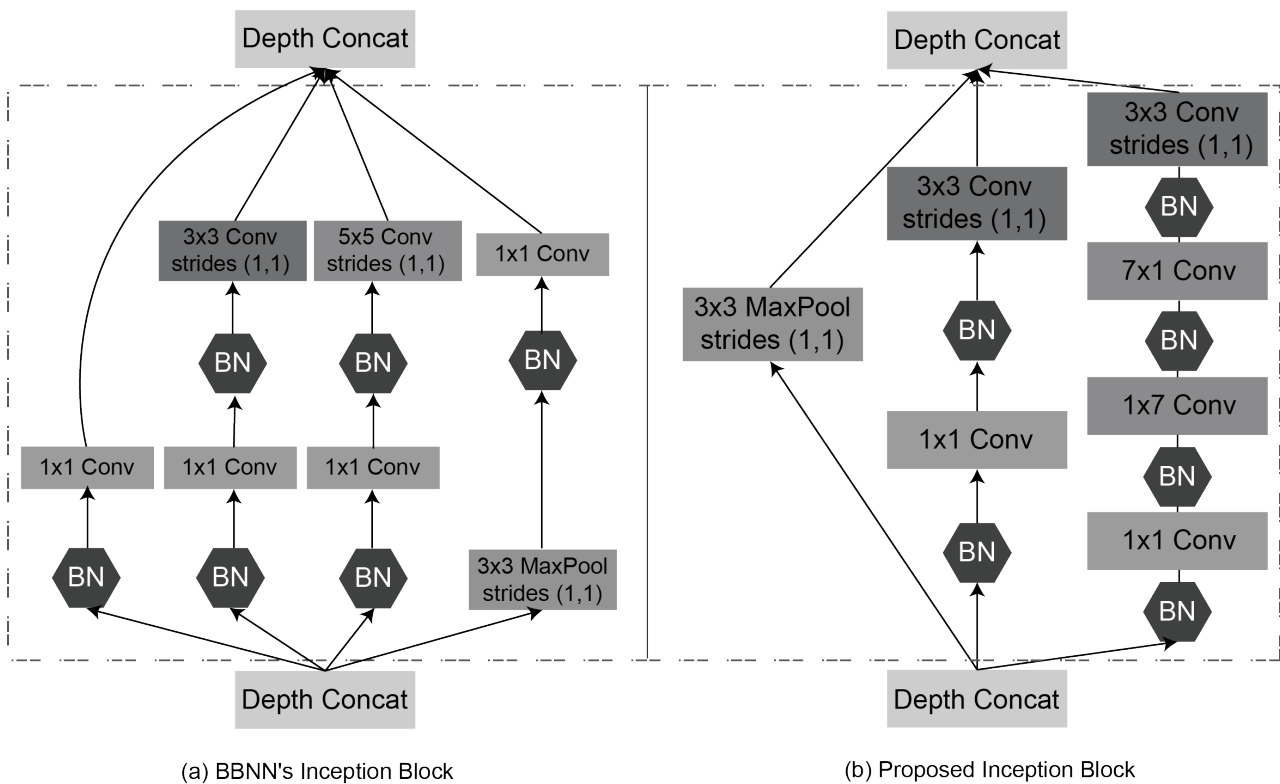


Figure 5.2: Proposed inception block modifications over the BBNN network

At this stage, the proposed network consisted of doubling the number of inception and dense blocks in the Broadcast Module (BM) of the BBNN, leaving the remaining layers (Shallow, Transition, and Decision) as proposed in [168].

Increasing the number of blocks reflected an increase in accuracy scores. On the other hand, it expanded the size of the training model and slowed the training process. Consequently, the architecture of the BBNN network was modified to reduce significant drawbacks due to overfitting and computation problems in the inception v1 block [116]. Many CNN related studies, in particular a music-related study in [141], proved that dense blocks are better than residual blocks. Thus, it was decided to keep the dense connection of the BBNN network intact. Moreover, the BBNN network relied on the inception v1 proposed in [81] while adding BN and ReLU operations before each convolution. The original inception v1 was found computationally expensive as well as prone to overfitting in many cases. At this stage, the next contribution was to replace the modified inception v1 blocks with modified inception v4 blocks in order to improve the computation efficiency and most importantly to increase the accuracy. As mentioned in [116], the earlier inception modules (v1, v2, v3) were found more complicated than necessary. They proposed specialized "Reduction Blocks" A and B to change the width and height of the grid. This change produces a performance boost by applying uniform and simplified operations to the network. Figure 5.2 presents the modified inception blocks in detail. The block on the left

concerns the custom inception v1 block of BBNN, and the block on the right concerns our proposed inception v4 block. As previously mentioned, the left block is inspired by the inception v1 block in [81], while adding BN and ReLU operation before each convolution. On the other hand, the proposed inception block is inspired by the “Reduction Block B” introduced in [116]. Compared with BBNN’s inception block in [168], the “Reduction Block B” of inception v4 [116] reduces the network complexity by mainly removing unnecessary  $1 \times 1$  convolution operations and replacing the  $5 \times 5$  convolution with a stack of  $1 \times 7$ ,  $7 \times 1$ , and  $3 \times 3$  convolution operations. Also, it accentuates memory optimization to back-propagation by implementing the factorization technique of inception v3. This technique is responsible to reduce the dimensionality of convolution layers, which reduce overfitting problems. In this matter, it was proposed to use the same architecture as the “Reduction Block B”, while implementing BN and ReLU operations before each convolution.

## 5.3/ EXPERIMENTAL EVALUATION

In this section, the training hyperparameters are presented while evaluating the proposed contribution against state-of-the-art music genre classifiers. The training operations are performed using an NVIDIA Tesla V100 SXM2 GPU with 32 GB of memory.

### 5.3.1/ HYPERPARAMETERS AND TRAINING DETAILS

As mentioned in Section 2, the input images were prepared by generating a STFT spectrogram out of each music track of the GTZAN and the FMA dataset. Each spectrogram ( $600 \times 128$ ) was sliced into  $128 \times 128$  slices, taking only the first three slices as a visual representation of each music track. At this stage, the input images for GTZAN classification were 297 slices of spectrograms per genre (99 music tracks per genre), and the input images for FMA classification were 3000 per genre (1000 music tracks per genre).

Inspired by BBNN [168], the proposed network upgrades were added as well as the hyperparameters to start the training. Considering that the BBNN network was initially tested against the GTZAN dataset [6], the same hyperparameters as the BBNN network were used for this case. The ADAM optimizer was selected to minimize the categorical cross-entropy between music genre labels, a batch size of 8 and an epoch size equal to 100. An initial learning rate of 0.01 was configured, while automatically decreasing its value by a factor of 0.5 once the loss stops improving after 3 epochs. The early stopping mechanism was implemented to prevent overfitting, and the GTZAN input spectrograms were fed to the classifier through 10-folds cross-validation training. Since all related publications used different dataset split ratios, the same ratio as BBNN’s [168] is adopted to compare our results with BBNN in particular and with other publications in general. Thus,

<i>GTZAN Classification</i>		
<b>Methods</b>	<b>Preprocessing</b>	<b>Accuracy</b>
AuDeep[105]	Mel Spectrogram	85.40
NNet2[97]	STFT	87.40
Hybrid model[128]	MFCC, SSD, etc.	88.30
Transform learning[102]	MFCC	89.80
DenseNet+Data augmentation[141]	STFT Spectrogram	90.20
Multi-DNN[78]	MFCC	93.40
TPNTF[35]	MFCC	93.70
BBNN[168]	Mel Spectrogram	93.90
DNN+Transfer learning[124]	Mel Spectrogram	94.20
GIF generation Framework[172]	MFCC Spectrogram	94.70
<b>Our approach</b>	<b>STFT Spectrogram</b>	<b>97.51</b>

Table 5.1: Comparative table for GTZAN classification methods in terms of accuracy (%)

the training, testing and validation sets were randomly divided following an 8/1/1 proportion (80% for training, 10% for testing, and 10% for validation). The resulting training and testing accuracies were calculated by averaging all the accuracies concluded in the cross-validation folds.

Concerning the FMA dataset, the increase in the batch size revealed an accuracy increase. However, the same hyperparameters as GTZAN were used, in addition to keeping the same value of the batch size (8), to align our results with the existing ones. Before initiating the training, the inception block's training parameters were calculated for both, the BBNN network and the proposed approach. This calculation showed that the proposed inception block uses less than 26.78 percentage points (*pp*) of BBNN's inception block parameters.

### 5.3.2/ TESTING RESULTS

In the tables below (Table 5.1 and Table 5.2), the proposed approach is compared to the most recent and accurate methods. These methods either rely on deep learning models or hand-crafted feature descriptors to provide an efficient classification of the GTZAN and the FMA datasets.

Table 5.1 compares the music genre classifiers used on the GTZAN dataset. It shows the different methods used over this dataset, including its preprocessing features and the resulted accuracies. As mentioned in the Related Work chapter, each method relied on a different preprocessing and training approach to achieve the highest accuracy possible. The classification methods are enumerated in ascending order based on the accuracy score. As for the proposed approach, its related fields are displayed in bold in the table.

<i>FMA Classification (fma-small subset)</i>		
<b>Methods</b>	<b>Preprocessing</b>	<b>Accuracy</b>
Representation learning[114]	Mel Spectrogram	57.91
BBNN[168]	Mel Spectrogram	61.11
SongNet[147]	Raw audio	65.23
DenseNet+Data augmentation[141]	STFT Spectrogram	68.20
<b>Our approach</b>	<b>STFT Spectrogram</b>	<b>74.39</b>

Table 5.2: Comparative table for FMA classification methods in terms of accuracy (%)

The results show that the proposed method can outperform the accuracy of the BBNN network [168] specifically by 3.61 *pp*, and outperform the highest accuracy mentioned [172] by 2.81 *pp*.

As for the small subset of FMA, Table 5.2 presents the methods applied over the latter to provide accurate music genre classification. Similar to Table 5.1, this table shows the different methods used over this dataset, in addition to the preprocessing features used and the resulted accuracies. As for the proposed approach, it outperformed the highest accuracy over the FMA small subset [85] by 6.19 *pp*. Since the proposed approach was inspired by the BBNN network and the latter is not tested against the small subset of FMA, the BBNN Github code<sup>1</sup> was used as-is over this dataset for experimentation purposes. It resulted in an accuracy of 61.11%, found to be less than 13.28 *pp* of the proposed approach. It is important to note that the outperformance against the related publications is not limited to the proposed network contribution only. The proposed preprocessing process assisted in this outperformance, especially with the GTZAN faults, where we reduced the number of music tracks per genre. Furthermore, the idea of slicing the generated spectrograms to obtain a diversity of visual representations among the same music track.

## 5.4/ DISCUSSION

### 5.4.1/ LIBROSA VS SOX

In this study, we used the Sound eXchange (SoX) library to generate the greyscale STFT spectrograms. However, other studies have used the Librosa Python library to perform such audio-visual computations. Fig 5.3 and 5.4 presented below represent the same Blues track computed to a greyscale STFT spectrogram, Fig 5.3 is the Librosa-computed version and Fig 5.4 is the SoX-computed version that we adopt in this study. Below are the common differences between SoX and Librosa while adding our observations and motivation for adopting the SoX version. Librosa provides a high-level and user-friendly

interface to work with the data, while SoX is a command-line utility, which makes the scripting and programming part less user-friendly.

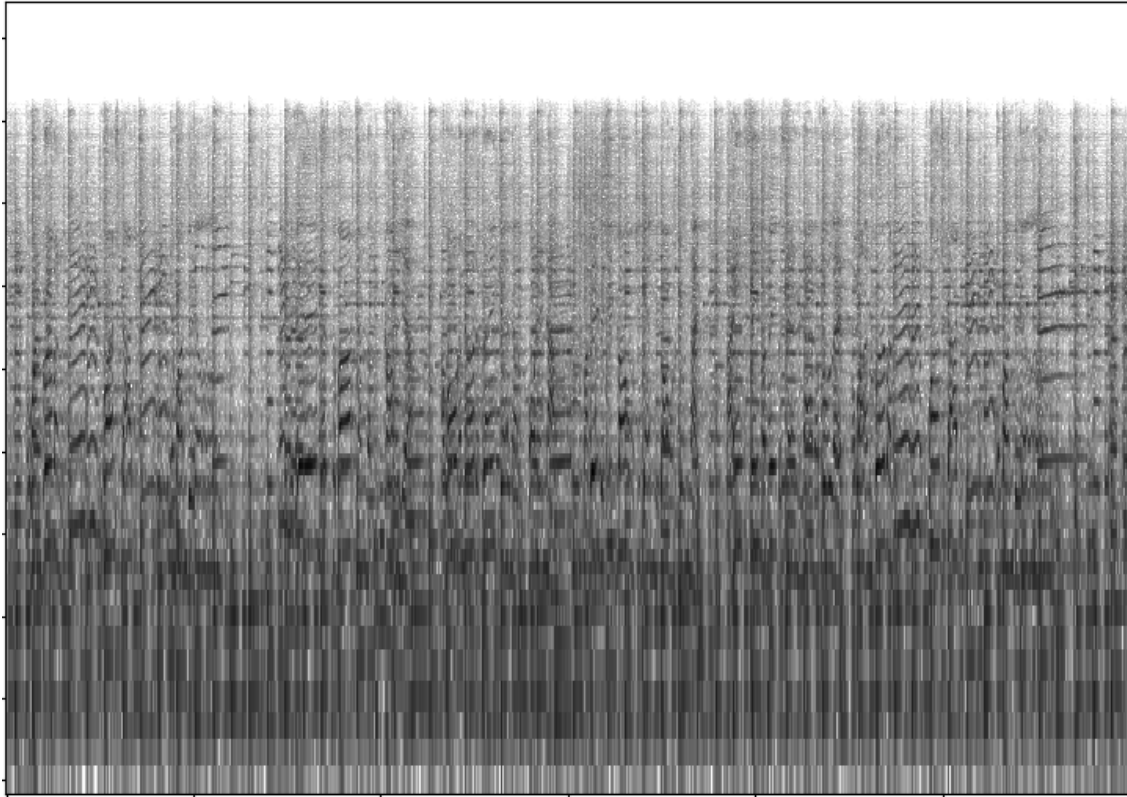


Figure 5.3: STFT greyscale spectrogram of a Blues track computed using Librosa

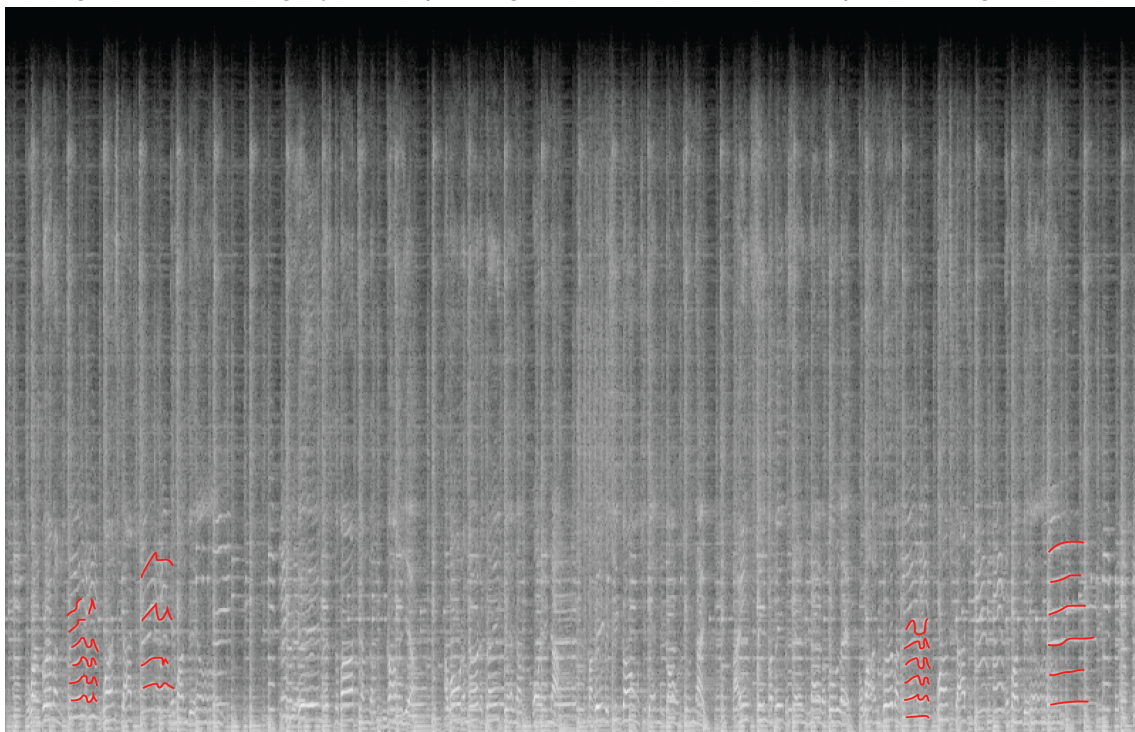


Figure 5.4: STFT greyscale spectrogram of a Blues track computed using SoX

In the STFT spectrogram generation part, the Librosa library allows for various parameters that provide more computational control and flexibility over the characteristics of the spectrogram. On the other hand, the SoX library has fewer parameters, requiring more manual configuration and scripting to achieve the flexibility of Librosa. The SoX library focuses on basic audio format conversion, editing and effects, while Librosa offers additional audio processing features such as beat tracking, pitch estimation and feature extraction for machine learning tasks. While the comparison showed better usability of Librosa over SoX, we present the observation that motivated us to adopt the SoX approach:

The spectrograms calculated with Librosa have a white background (subtractive colour synthesis), while the spectrograms calculated with SoX have a black background (additive colour synthesis). The additive synthesis approach helped to visualise a vivid and dynamic colour control of the greyscale spectrogram compared to the subtractive approach. This frequency representation, using a variation of white on a black background, emphasises the sharpness of the recorded frequencies, as well as the melodic features of the audio track, such as the harmonics, which we have marked in red (easier to detect by human observation). This feature, among others, encouraged us to adopt the SoX library, where the distinctive representation of melodic features helps to discriminate the visual features of audio tracks to achieve more accurate music genre classification. It is important to note that this observation was not our only reason for using SoX instead of Librosa. Nevertheless, it encouraged us to carry out a comparative approach, where a small sample of the GTZAN dataset was computed using both libraries. The empirical test results showed promising results of the SoX approach over Librosa.



### 5.4.2/ GENRE BY GENRE PRECISION PERCENTAGES OF THE PROPOSED APPROACH

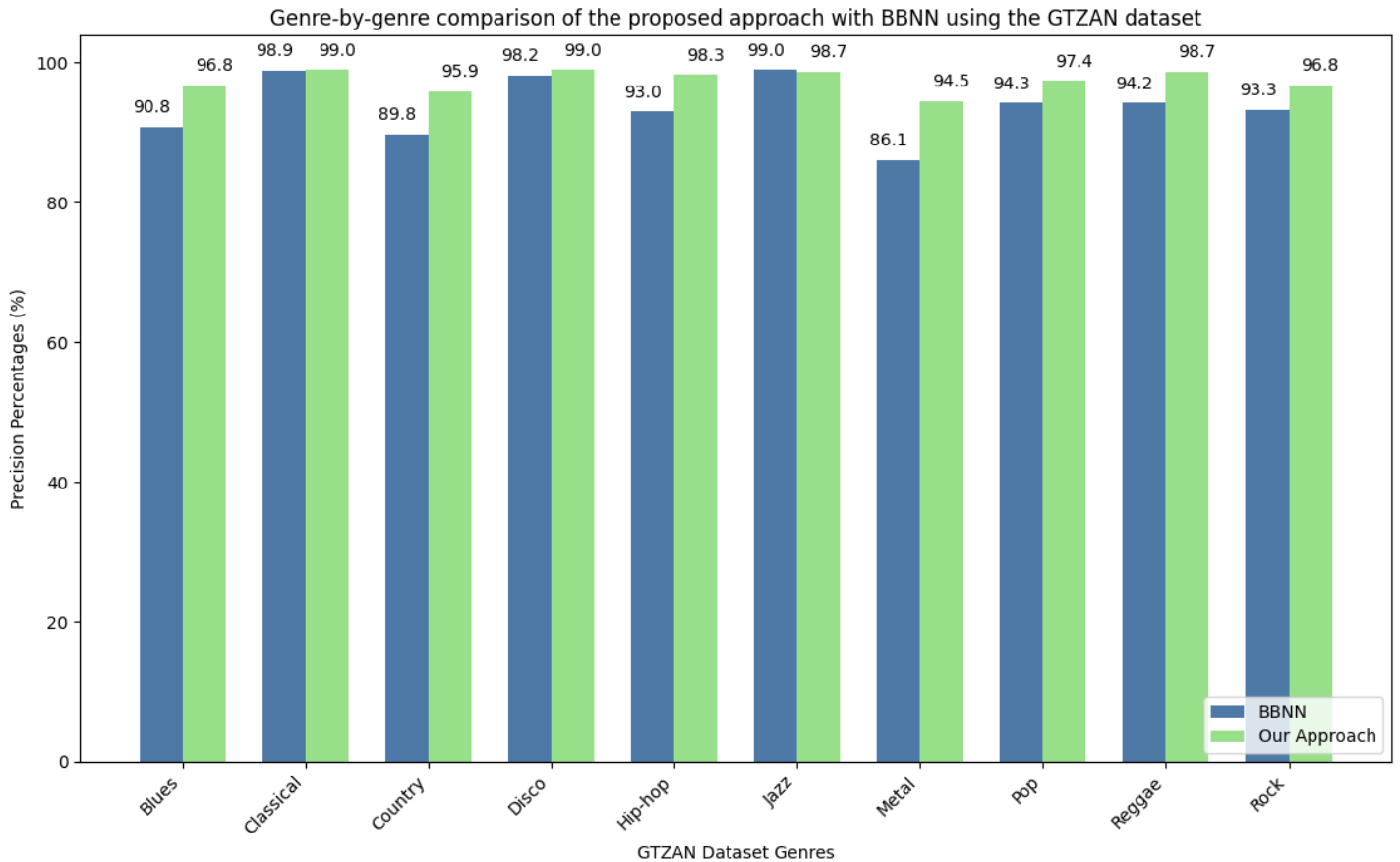


Figure 5.5: Comparison by genre of the proposed approach with BBNN using the GTZAN dataset

In the experiments section, we gathered state-of-the-art networks and evaluated their accuracy against either the GTZAN dataset or a smaller subset of the FMA dataset. Notably, the BBNN network, as described in [168], provided per-genre precision visualizations over the GTZAN dataset. We conducted a similar analysis to compare our proposed approach with BBNN. Figure 5.5 presents a comparative histogram displaying precision percentages for both networks. Precision measures the proportion of true positive predictions relative to all positive predictions made by the model, gauging its performance in correctly identifying positive cases. Mathematically, precision is calculated by dividing the number of true positive predictions by the sum of true positives and false positives, followed by multiplying the result by 100 to obtain a percentage. In Figure 5.5, the precision percentages for BBNN over the GTZAN dataset are displayed in blue, as per the original manuscript, while those for our proposed approach are shown in green. For clarity, we've included exact percentage scores above each histogram bar. Our analysis in Figure 5.5 reveals that, in the majority of genres, our proposed approach surpasses

BBNN in terms of precision percentages. Notably, it significantly improves precision for genres like Rock, Country, and Metal, which were challenging to distinguish in BBNN due to their closely related frequency distributions. This improvement can be attributed to our approach's expansion of inception and dense blocks, providing a more detailed perspective on spectrogram frequency variations. It emphasised a more detailed view of the frequency variations in the spectrograms, which allowed a better pattern recognition performance of each genre. Similarly, our proposed approach also enhances the classification of genres like Blues, Hip-Hop, and Pop, which exhibit a high correlation with Rock.

## 5.5/ CONCLUSION

In this chapter, we introduce significant enhancements to the CNN-based music genre classifier known as BBNN, coupled with a custom preprocessing procedure for generating STFT spectrograms from music tracks. These enhancements represent a significant improvement in music genre classification. They allow for a more accurate and robust solution to this task. We improve the user experience by predicting music tracks based on individual preferences by embedding music genre classifiers in music recommendation systems. Our approach introduces several improvements, including increasing the number of inception and dense blocks in the BBNN network, as well as enhancing the inception block with a novel reduction block B inspired by previous work. These enhancements have led to remarkable improvements in accuracy, consistently exceeding the performance of other music genre classifiers on both the GTZAN and FMA datasets. Specifically, our approach achieved superior accuracy scores of 97.51% on the GTZAN dataset and 74.39% on the FMA dataset. The preprocessing step plays a major role in our approach. We use greyscale STFT spectrograms instead of color Mel spectrograms, a choice validated through empirical tests. This preprocessing step, combined with the slicing of the spectrograms into multiple images, allows for a more diverse set of visual representations for each music track, thus contributing to improved accuracy. In particular, our approach demonstrates not only accuracy but also efficiency. It consists of an optimised inception block that uses fewer training parameters while achieving superior results. This reduction in training parameters highlights the innovative nature of our approach, simplifying the classification process for improved efficiency.



# AUTOMATIC MUSIC TRANSCRIPTION USING COMPUTER VISION

Generating music-related notations offers assistance for musicians in the path of replicating the music using a specific instrument. In this chapter, we evaluate the state-of-the-art guitar tablature transcription network named TabCNN against state-of-the-art computer vision networks. The evaluation is performed using the same dataset as well as the same evaluation metrics of TabCNN. Furthermore, we propose a new CNN-based network named TabInception to transcribe guitar-related notations, also called guitar tablatures. The network relies on a custom inception block converged by dense layers. The TabInception network outperforms the TabCNN in terms of multi-pitch precision (MP), tablature precision (TP), and tablature F-measure (TF). Moreover, the Swin Transformer achieves the best score in terms of multi-pitch recall (MR) and tablature recall (TR), while the Vision Transformer achieves the best score in terms of multi-pitch F-measure (MF). These results were acquired while training all the networks with 8 or 16 epochs. Motivated by the previous insights, we train the networks with more epochs and propose another network named Inception Transformer (InT) to surpass all the estimation metrics of TabCNN using a single network. The InT network relies on an inception block converged by a Transformer Encoder. The TabInception and the InT network outperformed all estimation metrics of TabCNN except the tablature disambiguation rate (TDR) when trained using a bigger epoch size. Code is available at <https://github.com/elachkarcharbel/Guitar-Tablature-Transcription>.

## 6.1/ INTRODUCTION

Over the last decade, researchers have been exploring the benefits of their innovations in music-related fields while producing tools that can facilitate musicians' daily tasks. One of the latter fields is automatic music transcription (AMT). The AMT is the task of generating a symbolic notation, and instructing a musician how to play a song using a specific

instrument. Several studies have been conducted in the AMT field, but only a few of them dealt with the guitar instrument, such as I. Barbancho et al. (2012) [50], Fuentes et al. (2012) [56], and A. M. Barbancho et al. (2012) [49]. As for automatic guitar transcription, the guitarist generally relies on both the music score and the tablature notation to play the song in question, as shown in Fig. 6.1. The music score represents the distribution of pitches in time, and the tablature notation defines the guitar strings and the position of the fingers along the fretboard to produce those pitches.

As described by Klapuri (2006) [17], the pitch is a perceptual property of sounds that allows their ordering on a frequency-related scale. It can be perceived as the property that measures the loudness of the sound. Contrarily, the tablature is a form of musical notation that indicates instrument fingering rather than pitches. This notation is mostly common for fretted stringed instruments like the guitar, where frets can be defined as thin strips of material inserted laterally at a specific position along the fretboard of the guitar.

The figure displays a musical score and its corresponding guitar tablature for the first four bars of the song "Radioactive" by Imagine Dragons. The music score is written in a 4/4 time signature with a key signature of one sharp (F#). It includes a dynamic marking of *f* (forte) and a "Capo 2" instruction. The tablature is organized into four measures, each corresponding to a bar of music. The strings are labeled T (Treble), A (Acoustic), and B (Bass). The tablature uses numbers 0-3 to indicate fret positions and 'o' for open strings. The first measure shows a power chord on strings 2, 3, and 4 at the 2nd fret, followed by a melody on strings 1, 2, and 3. The second measure continues the power chord and melody. The third measure features a power chord on strings 2, 3, and 4 at the 3rd fret, with a melody on strings 1, 2, and 3. The fourth measure shows a power chord on strings 2, 3, and 4 at the 2nd fret, with a melody on strings 1, 2, and 3.

Figure 6.1: Music score and tablature notation illustrating the first four bars of the song *Radioactive* by *Imagine Dragons*. The tablatures in the bottom represent the string to be played by the guitarist, in addition to the number of the fret to press.

This chapter explores several computer vision techniques for automatic guitar transcription. Inspired by the TabCNN model published in Wiggins and Kim (2019) [146], Constant-Q spectrograms are generated from each audio track and computed through Computer Vision approaches as visual representations of the audio data. Furthermore, we propose a new CNN-based network named TabInception that relies on Inception and Dense Blocks for automatic guitar transcription. Moreover, we propose another network named Inception Transformer (InT) to attempt to improve the results of TabInception and other featured networks. The InT network relies on an Inception Block converged by the Transformer Encoder Block proposed in [118].

Thus, the leading purpose of this study is to evaluate the TabCNN network against state-of-the-art computer vision networks while proposing new networks that might be capable of outperforming the latter network in the field of guitar tablature transcription. All the aforementioned networks, in addition to the TabInception and the InT network, are evaluated using the GuitarSet dataset published in Xi et al. (2019) [139], by the fact that TabCNN was assessed earlier against this dataset in Wiggins and Kim (2019) [146]. At a broader level, the aim is to explore which of the shallow networks like TabCNN or the

deeper networks, such as the proposed ones, can perform better on music transcription use cases. The remainder of this chapter is organized as follows: Section 2 presents the selected dataset in addition to the adopted preprocessing procedure. Section 3 interprets the proposed networks for automatic guitar transcription, while Section 4 compares the proposed networks with state-of-the-art CNNs and Transformer-based networks in terms of multi-pitch and tablature estimation metrics. Section 5 concludes the work and gives some directions for future work.

## 6.2/ DATA SELECTION AND PREPARATION

The TabCNN model proposed in Wiggins and Kim (2019) [146] holds the state-of-the-art record for guitar tablature transcription using CNNs. In this study, the same dataset chosen in TabCNN is used, in addition to the preprocessing procedure for computing audio features to images.

### 6.2.1/ THE GUITARSET DATASET

The GuitarSet dataset proposed in Xi et al. (2019) [139] consists of 360 solo guitar recordings, with a length of approximately 30 seconds for each one. The guitar solos were recorded using a hexaphonic pickup and a condenser microphone inside a soundproof recording studio. The authors used the JAMS file format of Humphrey et al. (2014) [73] to annotate the recorded guitar performances. Thus, the GuitarSet consists of 360 guitar recordings encoded in WAV format and annotated with 360 JAMS files separately. Each JAMS file contains various musical features such as tempo, key, beats and downbeats, note-level transcription (including string and fret position), and many more. Similar to TabCNN, the TabInception and InT networks use only the monophonic microphone signal to estimate the tablature.

### 6.2.2/ DATA PREPROCESSING

Similar to the TabCNN approach of Wiggins and Kim (2019) [146], the audio recordings were downsampled from 44100 to 22050 Hz to reduce the input signals' dimension. The input signals were normalized to obtain an identical range of amplitudes among all the recordings. This normalization is essential to achieve the next step: computing the convenient audio signal feature out of each recording.

Inspired by previous experiences in guitar tablature transcription, the Constant-Q Transform (CQT) is adopted as the feature to compute. For this reason, and to directly compare all studied networks with TabCNN, similar CQT parameters are adopted.

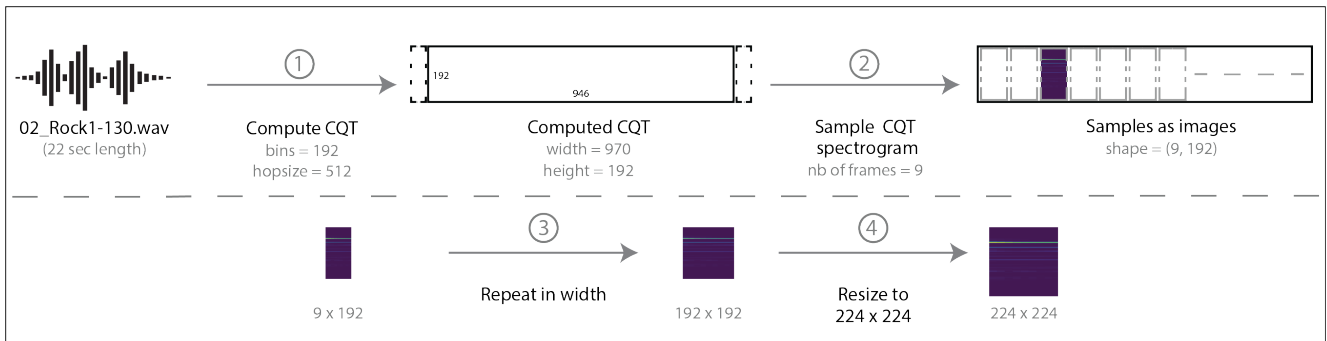


Figure 6.2: Audio to Image transformation through Constant-Q Transform computation

As shown in Fig. 6.2, and using the *Librosa* Python library, the CQT is computed over the audio recording in the first place. A value of 192 is selected for the bins and 512 for the hopsize parameter. The bins parameter consists of the intervals between samples in the frequency domain. It is estimated by dividing the sampling rate by the Fast Fourier Transform (FFT) size. On the other hand, the hopsize is the number of samples between each successive FFT window. It is processed by dividing the FFT size by an integer defining the overlap factor of FFT windows. As for this parameter selection validation, we plotted the `chroma_cqt` features using the selected bins and hopsize parameters. The chroma features captures harmonic and melodic characteristics of music while being robust to changes in timbre and instrumentation. In this case, the `chroma_cqt` analyzes these musical features following the CQT parameter already computed. While visualizing the plots, it was found that the produced chroma features were slightly noisy and unclear. Thus, the number of bins per octave parameter was scaled from its default value (12) to 24 to clarify the computed CQT by increasing its resolution. The CQT is then computed using the new parameter values: `hopsize = 512`, `number of bins = 192`, and `number of bins per octave = 24`.

At this stage, the computed CQT can be obtained as a visual representation of size 970x192, while adding zero padding on both sides of the CQT to achieve the sampling step (the initial size is 946x192 since the audio used in this example has a 22 seconds length and the hopsize used corresponds to 43 frames per seconds approximately). In addition, the sampling step (second in Fig. 6.2) is where the sliding context window of 9 frames takes place to generate multiple images of size 9x192 out of the initial computed CQT. The entire process results in multiple CQT images out of the same audio recording. Each image concerns nine successive frames of the initially computed CQT.

It was essential to resize the sampled CQT images into square-shaped images to compare the proposed and the existing approaches with state-of-the-art computer vision net-





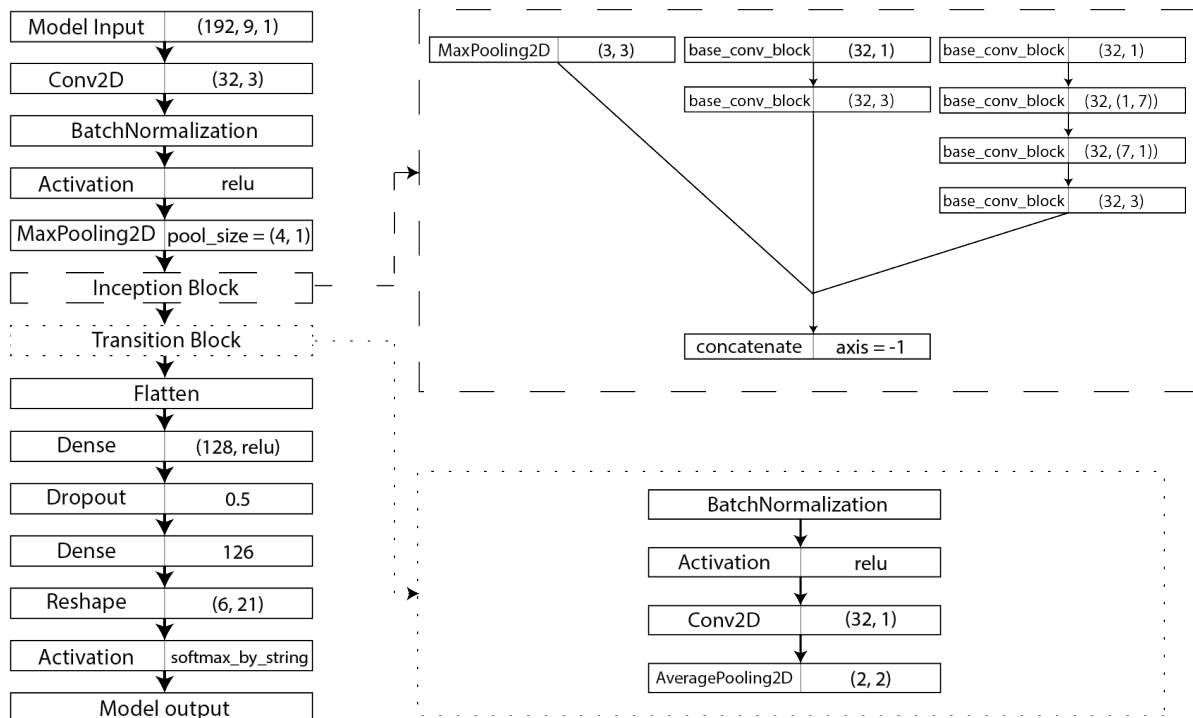


Figure 6.4: Architecture of the TabInception network

As shown in Fig. 6.4, the TabInception starts with an input layer taking images of shape (192, 9, 1). Thus, it involves swapping the axes of the computed images in the preprocessing steps to provide a proper data fitting. Consequently, we propose adding a two-dimensional convolutional layer of 32 filters adjacent to a Batch Normalization, a Relu activation, and a Max Pooling layer with a pool size equal to (4,1). The output of the latter bundle is fed into an Inception block that can be described as follows:

The proposed Inception block uses a similar architecture to the Inception v4 architecture implemented in Szegedy et al. (2017) [116], while adding Batch Normalization and Relu activation layers among adjacent Conv2D layers. Fig. 6.4 shows a high-level visualization of the Inception block, where several base convolutional blocks (`base_conv_block`) are interconnected together and are concatenated at the end with a MaxPooling2D layer.

Each `base_conv_block` consists of a Batch Normalization, a Relu activation, and a Conv2D layer with 32 filters. This technique ensures that the adopted inception approach will be less likely to over-fit. Also, the Batch Normalization improves memory optimization to backpropagation while reducing the intensive computations caused by convolutional layers. After concatenating the Inception block's calculations, the output is fed to the Transition Block. As presented in Fig. 6.4, the Transition block is the same as a `base_conv_block` with the addition of an AveragePooling2D layer after the Conv2D one. This approach is essential to downsample the huge spatial dimensions caused by the Inception block, and to converge the network into its decisive and final layers. Since TabInception concerns

guitar tablature transcription, the network should be able to compute multidimensional calculations. Hence, the use of the Flatten layer to convert the sixth channelled output to a single channelled one for Dense layer calculations. Each of the sixth channels consists of a guitar string having 21 frets. The Dense calculations are dropped out with a value of 0.5 while re-iterating the Dense computations using a number of units equal to the multiplication of the number of strings and frets ( $6 \times 21 = 126$  units). The output of the concluding layer is reshaped back to (6, 21) to compute the activation of each guitar string separately. Finally, the `softmax_by_string` activation function proposed in Wiggins and Kim (2019) [146] is used to concatenate the separately computed six softmax calculations and to unify the output.

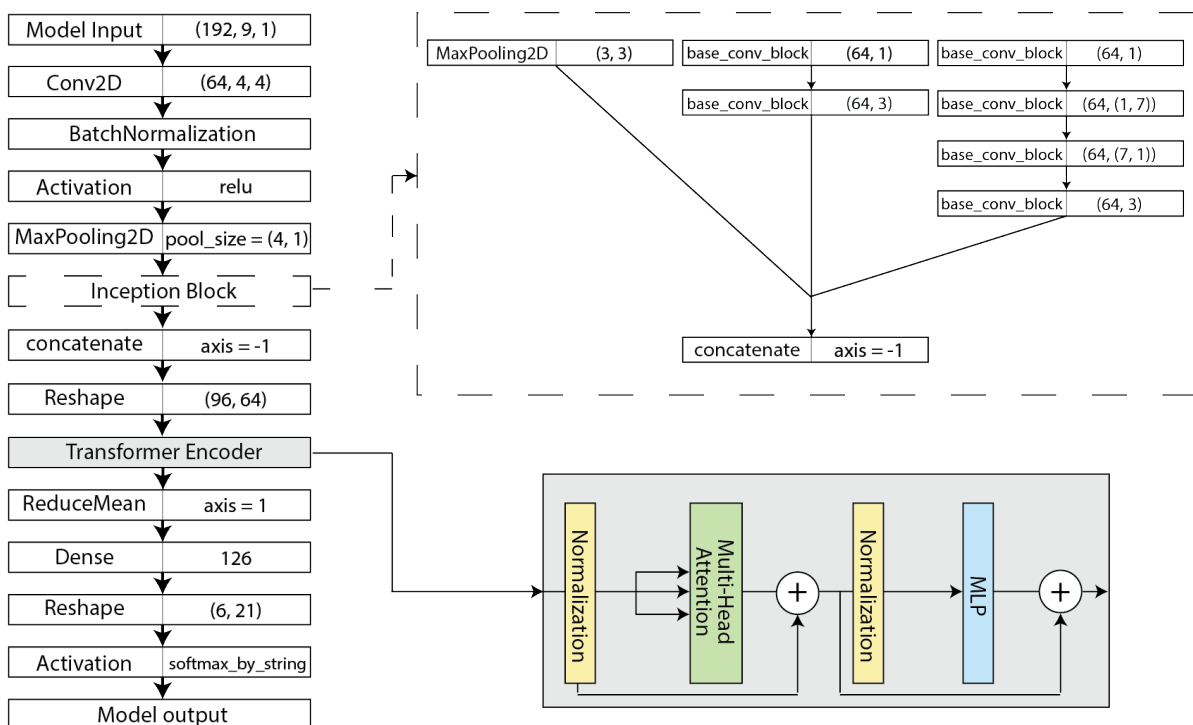


Figure 6.5: Architecture of the Inception Transformer (InT) network

### 6.3.2/ THE INCEPTION TRANSFORMER NETWORK

The TabInception network was trained over the computed CQT images besides other CNN-based and Transformer-based models. The latter networks could not surpass all the estimation metrics results of TabCNN (check detailed insights in the experiments section). Thus, a new network named Inception Transformer (InT) is proposed to attempt to exceed all the results of TabCNN.

Inspired by the precision of the TabInception network and the recall and the F-measure

of the Transformer-based models, the InT proposes a fusion between the Inception block of the TabInception network and the Transformer Encoder proposed in [118] and adopted in the Vision Transformer (ViT) model [149].

Similar to TabInception, The InT network relies on similar Input layers except using a number of filters equal to 64 instead of 32, as well as adding 4 strides to the initial Conv2D layer. The increased number of filters is adapted into the Inception Block of the InT network. The latter Block is identical to the one used in TabInception except for the number of filters. Furthermore, the computed calculations are concatenated and reshaped to (96, 64) to match the input shape needed for the Transformer Encoder. The Transformer Encoder adopted in [149] expects a sequence of embeddings vector that serves as input. These vectors consist of positional embeddings in addition to those of previously generated patches. As for the Transformer Encoder of the InT network, it expects a sequence of positional embedding along with the reshaped tensors produced out of the previously mentioned Inception block. Thus, the idea of generating patches and feeding them to the encoder is replaced by loading the encoder with convolutional-based tensors. The Transformer Encoder is responsible for alternating mutlihead self-attention blocks with MLP blocks. A LayerNormalization layer is applied before every block in addition to a residual connection after every block inside the encoder [144, 120]. The Transformer Encoder used in the proposed InT network relies on six transformer layers, an MLP dimension of 128, and a patch size equal to 4. The output of the latter encoder is fed to a ReduceMean layer to reduce the dimension of the tensor for the succeeding Dense layer. The Dense layer of shape 126 is then fed to the same concluding layers as TabInception to compute the activation of each guitar string separately. The same activation function and optimizer are adopted for both, the TabInception and the Inception Transformer networks.

It is important to note that the choice of the hyperparameters for both proposed networks (the number of filters, the number of transformer layers, and the MLP dimension...) is based on various trainings previously performed to find the most optimal value.

## 6.4/ EXPERIMENTS

In this section, the TabCNN network is compared to the TabInception and InT networks, in addition to state-of-the-art computer vision (CV) networks such as Liu et al. (2021) [169], Tan and Quoc (2019) [143], and Dosovitskiy et al. (2020) [149]. The CV networks were modified slightly by reshaping their decisive layers to provide a unified output across all networks (shape of (6, 21) as a matrix of 6 strings with 21 frets each). The implementation took place using the official version of the CV networks or its equivalent in *Keras*.

Image Size	Epochs	Network	MP	MR	MF	TP	TR	TF	TDR
192x9	8	TabCNN	0.9 ±0.016	0.764 ±0.043	0.826 ±0.025	0.809 ±0.029	0.696 ±0.061	0.748 ±0.047	0.899 ±0.033
192x9	16	TabCNN	0.927 ±0.008	0.7805 ±0.003	0.8474 ±0.0176	0.8101 ±0.0115	0.711 ±0.0268	0.757 ±0.0151	0.873 ±0.006
192x9	300 with ES	TabCNN	0.8549 ±0.013	0.722 ±0.021	0.782 ±0.0172	0.815 ±0.0185	0.697 ±0.0346	0.751 ±0.0237	<b>0.953</b> ±0.0136
192x9	8	TabInception	0.941 ±0.008	0.7189 ±0.031	0.815 ±0.0176	0.7973 ±0.0115	0.6455 ±0.0268	0.7134 ±0.0151	0.8473 ±0.006
192x9	16	TabInception	<b>0.9688</b> ±0.0192	0.7454 ±0.0518	0.8425 ±0.0317	<b>0.8519</b> ±0.0199	0.6911 ±0.0608	<b>0.7631</b> ±0.0443	0.8793 ±0.0244
192x9	300 with ES	TabInception	<b>0.9533</b> ±0.0147	0.7834 ±0.0339	0.86 ±0.0187	<b>0.8639</b> ±0.0158	<b>0.739</b> ±0.0356	<b>0.7965</b> ±0.0221	<b>0.906</b> ±0.0104
192x9	8	ViT	0.908 ±0.0165	0.8209 ±0.0373	0.8622 ±0.0204	0.7291 ±0.0329	0.7144 ±0.0444	0.7216 ±0.0349	0.802 ±0.0313
192x9	16	ViT	0.882 ±0.0066	0.8 ±0.0155	0.839 ±0.0056	0.7043 ±0.0074	0.6901 ±0.0183	0.6971 ±0.0099	0.798 ±0.0093
192x9	300 with ES	ViT	0.937 ±0.0115	<b>0.8524</b> ±0.0264	<b>0.8927</b> ±0.013	0.7586 ±0.0201	<b>0.7441</b> ±0.0313	0.7512 ±0.0224	0.8096 ±0.0203
192x9	8	InT	0.8785 ±0.0083	0.8213 ±0.0092	0.8489 ±0.0056	0.7202 ±0.0128	0.7206 ±0.0132	0.7203 ±0.0117	0.8198 ±0.0089
192x9	16	InT	0.891 ±0.0031	0.82 ±0.0062	0.854 ±0.0031	0.7134 ±0.0356	0.7019 ±0.0738	0.7076 ±0.0473	0.8 ±0.0034
192x9	300 with ES	InT	<b>0.9481</b> ±0.0057	<b>0.914</b> ±0.0077	<b>0.9307</b> ±0.0043	<b>0.8551</b> ±0.0242	<b>0.8041</b> ±0.0435	<b>0.828</b> ±0.0295	<b>0.901</b> ±0.00615
224x224	8	SwinTF	0.8875 ±0.0146	0.8034 ±0.0374	0.843 ±0.0146	0.709 ±0.0226	0.693 ±0.0212	0.7 ±0.041	0.798 ±0.031
224x224	16	SwinTF	0.9035 ±0.0075	0.8421 ±0.0034	0.8717 ±0.0024	0.7331 ±0.0019	0.726 ±0.0069	0.729 ±0.002	0.8114 ±0.007
224x224	300 with ES	SwinTF	0.9259 ±0.011	<b>0.8531</b> ±0.0204	<b>0.888</b> ±0.0085	0.7307 ±0.0122	0.7191 ±0.014	0.7248 ±0.0215	0.789 ±0.019
224x224	8	EfficientNetB0	0.839 ±0.0176	0.7691 ±0.0071	0.8025 ±0.016	0.6739 ±0.026	0.6406 ±0.048	0.656 ±0.034	0.803 ±0.031
224x224	16	EfficientNetB0	0.861 ±0.006	0.691 ±0.067	0.766 ±0.0386	0.733 ±0.0359	0.615 ±0.0695	0.668 ±0.0475	0.851 ±0.0401
224x224	300 with ES	EfficientNetB0	0.8947 ±0.0118	0.7747 ±0.037	0.83 ±0.0273	0.748 ±0.0309	0.6723 ±0.0587	0.708 ±0.0407	0.836 ±0.0355

Table 6.1: Comparative table for guitar tablature transcription using computer vision networks. The best score per metric is highlighted in **black**, the second best in **green**, and the third best in **red**.

The same parameters and hyperparameters are used across all the networks for better comparison with TabCNN. A batch size of 128 and a 6-fold cross-validation training method were selected while relying on the preprocessed CQT as input images to the network. The images were divided using an 85% training and 15% testing ratio for all networks.

The Swin Transformer (SwinTF) in Liu et al. (2021) [169] and the EfficientNetB0 in Tan and Quoc (2019) [143]<sup>1</sup> networks perform their best when trained using squared images since they rely on patch-based architectural structures. Therefore, it was favourable to experiment with both networks using a squared image format instead of performing architecture changes for fitting non-squared images. Hence, the resized 224x224 CQT images are used for these approaches. In contrast, the 192x9 CQT images are adopted to train the TabCNN, the TabInception, The InT, and the Vision Transformer (ViT) networks (Dosovitskiy et al. (2020) [149]) by the fact that they are not image size dependent. Thus, the proposed networks can be directly compared with the TabCNN network of Wiggins and Kim (2019) [146], while presenting other approaches where an image resizing may impact the training results.

It is important to mention that the ViT network relies on a patch-based structure. Nevertheless, it can be fed with non-squared images by its ability to transform each image into patches of equal width and height size. Thus, the input images are transformed into patches before being fed to the ViT encoder. In this experiment, and for the ViT network exclusively, we adopt a patch size of 4 and a hidden size of 64 after performing several empirical tests where both parameters were varied to maximize the evaluation results. Consequently, each of the 192x9 input images is transformed into 64 patches, having a size of 64x64 for each patch. The latter is conducted using the patch generation function proposed in the original code of the ViT [149] for rescaling and transforming the input images into patches.

Table 6.1 presents all the networks that we compare with TabCNN. The first training of TabCNN is written in *italic* to indicate that its results are shown as they appear in the official contribution. Contrarily, the remaining training is performed in our test environment.

The table header presents seven different multi-pitch and tablature estimation metrics. Each metric manifests an essential role already proposed in Wiggins and Kim (2019) [146]. The metrics referenced in Table 6.1 are the following:

**Multi-pitch Precision (MP)** measures the frequency of correctly detected pitches.

**Multi-pitch Recall (MR)** measures the frequency of existent pitch detection.

---

<sup>1</sup>The B0 base model of EfficientNet is the only selected model for this experiment since it is the only compatible model for computing 224x224-sized images. As for the SwinTF, we use the base architecture for this experiment, also known as swin.b.

**Multi-pitch F-measure (MF)** summarizes the overall multi-pitch estimation performance.

**Tablature Precision (TP)** measures the frequency of correctly detected tablatures.

**Tablature Recall (TR)** measures the frequency of existent tablature detection.

**Tablature F-measure (TF)** summarizes the overall tablature estimation performance.

**Tablature Disambiguation Rate (TDR)** measures the frequency of correctly detected pitches assigned to correctly detected tablatures.

As shown in Table 6.1, the networks were trained using two different epoch sizes. An epoch size of 8 is used to compare the results with the official TabCNN results. Furthermore, an epoch size of 16 is adopted to identify the behaviour of each network using longer iterations.

At an epoch size of 8, the TabInception outperformed the TabCNN by 4.1 percentage points (*pp* in terms of multi-pitch precision (MP)). On the other hand, the InT network and the transformer-based networks (ViT and SwinTF) can either outperform or obtain the same results as the TabCNN in terms of multi-pitch recall (MR) and F-measure (MF). The ViT exceeded the TabCNN by 3.62 *pp* in terms of MF and 5.69 *pp* in terms of MR. Also, the SwinTF exceeded TabCNN's MF by 1.7 *pp*, and TabCNN's MR by 3.94 *pp*. As for the InT network, it exceeded TabCNN's MF by 2.29 *pp* and TabCNN's MR by 5.73 *pp*. These results show that the TabInception network is a good solution for better pitch detection, while both, the proposed InT network and the transformer-based networks are better options when the comparison concerns the MR and MF metrics.

The TabInception network outperformed the TabCNN network in terms of MP, TP, and TF metrics when increasing the epoch size to 16. It achieved the greatest results concerning the multi-pitch precision metric. It outperformed TabCNN's MP by 4.18 *pp*, the TP by 4.17 *pp*, and the TF by 0.61 *pp*. As for the proposed InT network, it surpassed the TabCNN network in terms of MR and MF. Moreover, among the remaining networks, the SwinTF improved its results using an increased epoch size. Contrarily, the EfficientNetB0 could not exceed TabCNN's results in either epochs variations.

Motivated by the increase in metrics when raising the epochs size, we configured an epochs size of 300 while establishing the early stopping mechanism with a patience value equal to 5. Thus, the models will keep training until they reach a safe point to stop without overfitting. All the tests produced using the latter configuration are highlighted in a dashed outline in Table 6.1 to discriminate the latter from the legacy configuration ( 8-16 epochs without an early stopping mechanism). Also, we highlight the best score per metric with a **black** bold color, the second best with a **green** bold color, and the third best with a **red** bold color. The results show that the TabInception network achieved the best result in terms of TP, and the InT network achieved the best results for MR, MF, TR, and TF. Both proposed networks were able to surpass all of TabCNN's results except the TDR metric. The TabCNN preserved the best result in terms of TDR in that case. The significant

TDR value of TabCNN is due to the closer MP and TP values compared to the remaining networks. As for the SwinTF, the ViT, and the EfficientNetB0, some of their metrics' results increased but could not considerably surpass TabCNN's values at all times. It is essential to note the preference to choose the InT network over the TabInception for significantly exceeding TabCNN and for clustering the remarkable metrics' results among all the studied networks.

## 6.5/ CONCLUSION

In this chapter, two networks were proposed for guitar tablature transcription. The first network, TabInception, relies on a custom inception block converged by dense layers. The second network, Inception Transformer (InT), relies on a similar inception block of TabInception converged by a Transformer Encoder. Both networks were compared against the state-of-the-art guitar tablature transcription network named TabCNN and other recent computer vision networks. The experiment results showed that the proposed networks can outperform the TabCNN in terms of multi-pitch precision (MP), multi-pitch recall (MR), multi-pitch F-measure (MF), tablature precision (TP), tablature recall (TR), and tablature F-measure (TF). On the other hand, the latter networks could not outperform the TabCNN network in terms of tablature disambiguation rate (TDR) due to larger gaps between their MP and TP values. As for the Transformer-based networks (ViT and SwinTF), the increase in the epochs size reflected an increase in the majority of their estimation metrics. These networks achieved relevant values in terms of MR and MF. Nevertheless, they did not exceed TabCNN's results in the remaining metrics. Last but not least, the EfficientNetB0 also improved when increasing the epochs size but did not produce as promising results as the previously mentioned networks. Our future work should focus on exploring the performance and the usability of both proposed networks for transcribing tablatures of other string instruments such as the violin, cello, and harp. Furthermore, it would be essential to test the proposed networks on computer vision use cases beyond the tablature transcription or even the music field to better evaluate and explore the importance of such contribution.

# V

## CONCLUSION & PERSPECTIVES





## CONCLUSION & PERSPECTIVES

### 7.1/ CONCLUSION

This thesis proposes novel approaches to music encoding and deep learning techniques in the context of music applications. The primary motivation behind this research is to address the lack of support for oriental music genres within the music industry and to enhance the capabilities of deep learning models for tasks such as music genre classification and automatic music transcription. This dissertation is structured into three key parts. The first part provides a comprehensive overview of the foundational principles and related research studies on music encoding and deep learning in music applications. The subsequent parts present specific contributions made to the fields of music encoding and deep learning in music applications.

The first part encompasses a thorough review of related tools, software, and research literature that connect to either music encoding or deep learning for music applications. It introduces the concept of extracting knowledge from music scores and underscores the significance of such tools for music analysis and archival purposes. Additionally, it highlights cutting-edge technologies for converting music scores from XML-based formats to other formats, emphasising their utility for targeted analysis and research on music scores. Furthermore, this part focuses on the challenges encountered when employing these tools in the context of oriental music datasets, particularly Middle Eastern ones. Subsequently, following the discussion of music encoding, the focus shifts to audio signal processing techniques used to extract visual features from audio-related data. The processes involved the generation of spectrograms, which play a pivotal role in subsequent contributions. Various research techniques utilising these technologies for music analysis purposes are presented. The concept of computer vision is introduced, emphasising its relevance to music applications and the crucial role played by audio signal processing techniques as input data for computer vision networks. The broader concept of computer vision is further condensed into the main themes of the second and third parts, namely music genre classification and guitar tablature estimation. The latest and most effective

studies in music genre classification are detailed, with an emphasis on the achieved accuracies when applied to datasets such as GTZAN and FMA. Additionally, state-of-the-art tablature transcription techniques are outlined, along with key metrics used to assess their performance. Finally, the last section of this part provides an overview of commonly utilised datasets for music application studies, with a primary focus on the three datasets used in this research: GTZAN, FMA, and GuitarSet.

The second part focuses on knowledge extraction and format conversion of music scores for music encoding purposes. Challenged by the lack of support for the Eastern music dataset in the field of music encoding, an ontology called MusicPatternOWL is proposed to structure the pattern retrieval and analysis of music scores. The ontology is inspired by an algorithm for encoding traditional modal monodies of the Middle East, which makes it supportive of Eastern music scores. The structural aspects and motivation for creating the latter ontology are presented, while a proof of concept of its use through SPARQL queries is shown. The MusicPatternOWL is seen as an initiative to improve pattern retrieval analysis algorithms, especially those related to Eastern music datasets. Motivated by the positive impact of deep learning on music applications, and challenged by the lack of support for eastern music score conversion, a music score converter from MEI to JSON is proposed to ensure a lossless data process while converting the score from one format to another. The MEI2JSON converter is evaluated in terms of time and space complexity, quality assessment and storage evaluation. It showed great potential in both reducing the storage allocation size and ensuring high data quality throughout the conversion process.

The third and final part consists of exploiting two different topics in the use of computer vision networks for music applications. The first topic, music genre classification, proposes preprocessing approaches to better compute and prepare spectrograms from audio features, in addition to network upgrades to a well-known music genre classifier called BBNN that exploits low-level features of audio computations. The proposed contribution is evaluated against state-of-the-art music genre classifiers using the GTZAN and FMA datasets. The proposed approach can outperform the existing approaches on the two datasets in terms of accuracy score. A further discussion was elaborated to show our motivation for the use of one audio processing tool over the other, while a result visualisation was added to further compare the proposed approach with the benchmark network. The second topic, automatic music transcription, is where a CNN-based network called TabInception and a CNN-Transformer-based network called Inception Transformer (InT) took place in an attempt to outperform the state-of-the-art network for guitar tablature transcription called TabCNN. All previous networks, as well as state-of-the-art computer vision networks, were evaluated on multi-pitch and tablature metrics using the GuitarSet dataset. The audio data of GuitarSet was computed to visual Constant-Q Transform spectrograms, which were considered input data images to the networks evaluated in this study. Both proposed networks were able to achieve the best results, while the InT net-

work is considered the most convenient solution to adopt for its top score results in the majority of the presented metrics.

We believe that all the proposed solutions, both in the music encoding part and in the deep learning for music applications part, can serve as initiatives to suggest further phases of improvement for different music-related challenges.

## 7.2/ PERSPECTIVES

This thesis has presented innovative approaches to music encoding and deep learning techniques in the context of music applications, with a particular focus on addressing the under representation of oriental music genres and enhancing the capabilities of deep learning models. The contributions of this work have laid the foundation for several potential research and development areas to be explored in the future.

Concerning music encoding, an important avenue for further exploration is the extension of our initiative to Middle Eastern music scores. It is imperative to investigate the behavior of MusicPatternOWL on more extensive datasets, encompassing a diverse range of musical forms beyond traditional modal monodies. This exploration will facilitate a deeper understanding of the ontology's behavior and adaptability, enabling it to support enhanced knowledge extraction and structured pattern retrieval. Furthermore, the integration of machine learning techniques with music coding ontologies holds promise in providing automatic adaptation and support for multiple datasets, thereby promoting a generalisation effect and strengthening their overall credibility in diverse musical contexts.

Shifting our focus to deep learning applications in music, it is essential to subject the proposed music genre classifier to a more comprehensive evaluation. This requires testing the classifier on a larger and more diverse set of music genre datasets, including those referenced in the related work chapter. Furthermore, the practical deployment of these classifiers within music recommendation systems or realistic simulations of music streaming services is essential. Such implementations will not only demonstrate their usability but also evaluate their speed in delivering recommendations based on a user's playlist history. Additionally, these classifiers can play a key role in automating the categorisation of music databases by genre, facilitating content indexing and organisation. Motivated to explore new music application fields, we merged our focus from music genre classification to automatic music transcription. However, there might be still room for improving the proposed classifier. Thus, it would be interesting to test Transformer-based networks and diffusion networks to explore further the best network approach for music genre classification. In the field of guitar tablature transcription, there is an immediate opportunity for future work. It is imperative to assess the versatility of the proposed CNN-based and

CNN-transformer-based networks by applying them to a broader spectrum of music and audio-related use cases. The versatility test will help provide a more comprehensive understanding of the significance and adaptability of these contributions. Furthermore, conducting comparative studies to benchmark the behavior of our networks against other hybrid network solutions and emerging models, such as diffusion models, within the context of automatic music transcription will offer valuable insights into the strengths and weaknesses of various approaches.

Since the Transformer-based networks were generally created for NLP use cases, and since the music data can be represented in XML-based formats as a music score, it would be interesting to exploit NLP techniques for classification, transcription, and generation tasks and compare the resulting outcomes with the computer vision-based ones. The latter comparison will help us get insightful ideas for either pursuing music application studies using visual audio features or shifting to vectorised audio features.

Considering the close correlation between music generation and music transcription, a convincing avenue for future research is to remodel the tablature transcription networks for music generation tasks. Leveraging these networks to generate entirely new musical compositions, including novel guitar recordings, represents an exciting frontier. The audio generated in this manner can serve as synthetic data for broader research studies in music composition and generation, providing unique insights into the creative capabilities of deep learning algorithms when confronted with the complex nuances of musical expression.

Both deep learning contributions relied on generating spectrograms out of the audio signal for further purposes. A fascinating avenue for future exploration involves harnessing the power of Generative networks to reverse-engineer these spectrograms while retaining critical phase information. This could facilitate the generation of synthetic audio data, which can significantly contribute to the expansion of experimental datasets, enabling the execution of more extensive and robust tests in various applications.

In this research, we explored both visual audio characteristics and textual representations, specifically those in XML-based formats like MEI. Consequently, it is valuable to examine multi-modal strategies that merge these two representations to assess how their combined modelling can enhance various music-related tasks. Additionally, a significant endeavour is the creation of AI-based automated tools for encoding and annotating music datasets, particularly those that lack digital coverage, to help the manual workload for musicians.

# PUBLICATIONS

## JOURNAL PAPERS

- Charbel El Achkar and Talar Atéchan. “*MEI2JSON: a pre-processing music scores converter*”. In **International Journal of Intelligent Information and Database Systems**. 15, 1 (2022), 57–77. <https://doi.org/10.1504/ijids.2022.120130>

## CONFERENCE PAPERS

- Charbel El Achkar and Talar Atéchan. “*Supporting Music Pattern Retrieval and Analysis: An Ontology-Based Approach*”. In **Proceedings of the 10th International Conference on Web Intelligence, Mining and Semantics (WIMS 2020)**. Association for Computing Machinery, New York, NY, USA, 17–20. <https://doi.org/10.1145/3405962.3405973>
- Charbel El Achkar, Raphael Couturier, Talar Atéchan, and Abdallah Makhoul. “*Combining Reduction and Dense Blocks for Music Genre Classification*”. In: **Mantoro, T., Lee, M., Ayu, M.A., Wong, K.W., Hidayanto, A.N. (eds) Neural Information Processing. ICONIP 2021**. Communications in Computer and Information Science, vol 1517. Springer, Cham. [https://doi.org/10.1007/978-3-030-92310-5\\_87](https://doi.org/10.1007/978-3-030-92310-5_87)
- Charbel El Achkar, Raphael Couturier, Abdallah Makhoul, and Talar Atéchan. “*Leveraging Computer Vision Networks for Guitar Tablature Transcription*”. In: **Sheng, B., Bi, L., Kim, J., Magnenat-Thalmann, N., Thalmann, D. (eds) Advances in Computer Graphics. CGI 2023**. Lecture Notes in Computer Science, vol 14495. Springer, Cham. [https://doi.org/10.1007/978-3-031-50069-5\\_2](https://doi.org/10.1007/978-3-031-50069-5_2)



# BIBLIOGRAPHY

- [1] DAVIS, S., AND MERMELSTEIN, P. **Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences.** *IEEE transactions on acoustics, speech, and signal processing* 28, 4 (1980), 357–366.
- [2] COOK, P. R. **Physically informed sonic modeling (phism): Percussive synthesis.** In *Proceedings of the 1996 International Computer Music Conference* (1996), The International Computer Music Association, pp. 228–231.
- [3] MIDI MANUFACTURERS ASSOCIATION. **Complete midi 1.0 detailed specification,** March 1996.
- [4] LOGAN, B., AND OTHERS. **Mel frequency cepstral coefficients for music modeling.** In *Ismir* (2000), vol. 270, Plymouth, MA, p. 11.
- [5] BRANDSTEIN, M., AND WARD, D. **Microphone arrays: signal processing techniques and applications.** Springer Science & Business Media, 2001.
- [6] GEORGE, T., GEORG, E., AND PERRY, C. **Automatic musical genre classification of audio signals.** In *Proceedings of the 2nd international symposium on music information retrieval, Indiana* (2001), vol. 144.
- [7] GOOD, M. D. **Musicxml for notation and analysis.** In *in Hewlett, W.B. and Selfridge-Field, E. (Eds.): The Virtual Score: Representation, Retrieval, Restoration* (MIT Press, Cambridge (MA); London (UK), 2001), pp. 113–124.
- [8] ROLAND, P. **The music encoding initiative (mei).** In *Proceedings of the First International Conference on Musical Applications Using XML* (2002), pp. 55–59.
- [9] SIN, D., AND CHAN, C. B. H. **Lxml: lightweight xml for storing data in smart card wallets.** In *International Conference on Internet Computing* (2002).
- [10] TZANETAKIS, G., AND COOK, P. **Musical genre classification of audio signals.** *IEEE Transactions on speech and audio processing* 10, 5 (2002), 293–302.
- [11] **5th international conference on music information retrieval (ismir 2004) rhythm description contest,** 2004.
- [12] MH, K. **Saxon—the xslt and xquery processor,** 2004.



- [13] BELLO, J. P., DAUDET, L., ABDALLAH, S., DUXBURY, C., DAVIES, M., AND SANDLER, M. B. **A tutorial on onset detection in music signals.** *IEEE Transactions on speech and audio processing* 13, 5 (2005), 1035–1047.
- [14] HARTE, C., AND SANDLER, M. **Automatic chord identification using a quantised chromagram.** In *Audio Engineering Society Convention 118* (2005), Audio Engineering Society.
- [15] HOMBURG, H., MIERSWA, I., MÖLLER, B., MORIK, K., AND WURST, M. **A benchmark dataset for audio classification and clustering.** In *ISMIR* (2005), vol. 2005, pp. 528–31.
- [16] LIN, C.-Y., AND WANG, H.-C. **Language identification using pitch contour information.** In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.* (2005), vol. 1, IEEE, pp. 1–601.
- [17] KLAPURI, A., AND DAVY, M., Eds. **Introduction to Music Transcription.** Springer US, Boston, MA, 2006, pp. 3–20.
- [18] TUOHY, D. R., AND POTTER, W. D. **An evolved neural network/hc hybrid for tablature creation in ga-based guitar arranging.** In *International Conference on Mathematics and Computing* (2006).
- [19] LEE, C.-H., SHIH, J.-L., YU, K.-M., AND SU, J.-M. **Automatic music genre classification using modulation spectral contrast feature.** In *2007 IEEE International Conference on Multimedia and Expo* (2007), IEEE, pp. 204–207.
- [20] MÜLLER, M. **Information retrieval for music and motion**, vol. 2. Springer, 2007.
- [21] PROAKIS, J. G. **Digital signal processing: principles, algorithms, and applications, 4/E.** Pearson Education India, 2007.
- [22] RICHARDSON, L. **Beautiful soup documentation**, 2007.
- [23] STOWELL, D., AND PLUMBLEY, M. **Adaptive whitening for improved real-time audio onset detection.** In *Proceedings of the 2007 International Computer Music Conference, ICMC 2007* (2007), pp. 312–319.
- [24] BENESTY, J., CHEN, J., AND HUANG, Y. **Microphone array signal processing**, vol. 1. Springer Science & Business Media, 2008.
- [25] PANAGAKIS, I., BENETOS, E., AND KOTROPOULOS, C. **Music genre classification: A multilinear approach.** In *ISMIR* (2008), pp. 583–588.
- [26] RODRIGUES, T., ROSA, P., AND CARDOSO, J. **Moving from syntactic to semantic organizations using jxml2owl.** *Computers in Industry* 59, 8 (2008), 808–819.

- [27] SERRA, J., GÓMEZ, E., HERRERA, P., AND SERRA, X. **Chroma binary similarity and local alignment applied to cover song identification.** *IEEE Transactions on Audio, Speech, and Language Processing* 16, 6 (2008), 1138–1151.
- [28] BIKAKIS, N., GIOLDASIS, N., TSINARAKI, C., AND CHRISTODOULAKIS, S. **Querying xml data with sparql.** In *Database and Expert Systems Applications* (Berlin, Heidelberg, 2009), S. S. Bhowmick, J. Küng, and R. Wagner, Eds., Springer Berlin Heidelberg, pp. 372–381.
- [29] BREITLING, F. **A standard transformation from xml to rdf via xslt.** *Astronomische Nachrichten: Astronomical Notes* 330, 7 (2009), 755–760.
- [30] ALVARO, J. L., AND BARROS, B. **Musicjson: A representation for the computer music cloud.** In *Proceedings of the 7th Sound and Music Computer Conference, Barcelona* (2010).
- [31] CANNAM, C., LANDONE, C., AND SANDLER, M. **Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files.** In *Proceedings of the 18th ACM international conference on Multimedia* (2010), pp. 1467–1468.
- [32] CUTHBERT, M. S., AND ARIZA, C. **Music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data.** In *Proceedings of the 11th International Society for Music Information Retrieval Conference* (Utrecht, Netherlands, Aug. 2010), ISMIR, pp. 637–642.
- [33] KLAPURI, A., VIRTANEN, T., AND HEITTOLA, T. **Sound source separation in monaural music signals using excitation-filter model and em algorithm.** In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing* (2010), IEEE, pp. 5510–5513.
- [34] MENZIES, D. **Miller puckette, the theory and technique of electronic music. singapore: World scientific publishing, 2007. isbn 13 978-981-270-077-3.** *Organised Sound* 15, 2 (2010), 179–180.
- [35] PANAGAKIS, Y., AND KOTROPOULOS, C. **Music genre classification via topology preserving non-negative tensor factorization and sparse representations.** In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing* (2010), IEEE, pp. 249–252.
- [36] RABINER, L., AND SCHAFFER, R. **Theory and applications of digital speech processing.** Prentice Hall Press, 2010.
- [37] SMITH III, J. O. **Physical audio signal processing: For virtual musical instruments and audio effects.** (*No Title*) (2010).

- [38] VANROSSUM, G., AND DRAKE, F. L. **The python language reference**, vol. 561. Python Software Foundation Amsterdam, Netherlands, 2010.
- [39] YANG, B., AND LUGGER, M. **Emotion recognition from speech signals using new harmony features**. *Signal processing* 90, 5 (2010), 1415–1423.
- [40] BERTIN-MAHIEUX, T., ELLIS, D., WHITMAN, B., AND LAMERE, P. **The million song dataset**. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)* (01 2011), pp. 591–596.
- [41] DAHL, G. E., YU, D., DENG, L., AND ACERO, A. **Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition**. *IEEE Transactions on audio, speech, and language processing* 20, 1 (2011), 30–42.
- [42] EWERT, S. **Chroma toolbox: Matlab implementations for extracting variants of chroma-based audio features**. In *Proc. ISMIR* (2011).
- [43] HANKINSON, A., ROLAND, P., AND FUJINAGA, I. **The music encoding initiative as a document-encoding framework**. In *ISMIR* (2011), pp. 293–298.
- [44] LACOSTE, D., SAWANT, K. P., AND ROY, S. **An efficient xml to owl converter**. In *Proceedings of the 4th India Software Engineering Conference* (New York, NY, USA, 2011), ISEC '11, Association for Computing Machinery, p. 145–154.
- [45] LIBEKS, J., AND TURNBULL, D. **You can judge an artist by an album cover: Using images for music annotation**. *IEEE MultiMedia* 18, 4 (2011), 30–37.
- [46] MARSDEN, A. **Software for schenkerian analysis**. In *Proceedings of the 2011 International Computer Music Conference, ICMC 2011, Huddersfield, UK, July 31 - August 5, 2011* (2011), Michigan Publishing.
- [47] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., AND OTHERS. **Scikit-learn: Machine learning in python**. *the Journal of machine Learning research* 12 (2011), 2825–2830.
- [48] YING, D., YAN, Y., DANG, J., AND SOONG, F. K. **Voice activity detection based on an unsupervised learning framework**. *IEEE Transactions on Audio, Speech, and Language Processing* 19, 8 (2011), 2624–2633.
- [49] BARBANCHO, A. M., KLAPURI, A., TARDON, L. J., AND BARBANCHO, I. **Automatic transcription of guitar chords and fingering from audio**. *IEEE Transactions on Audio, Speech, and Language Processing* 20, 3 (2012), 915–921.

- [50] BARBANCHO, I., TARDON, L. J., SAMMARTINO, S., AND BARBANCHO, A. M. **Inharmonicity-based method for the automatic generation of guitar tablature.** *IEEE Transactions on Audio, Speech, and Language Processing* 20, 6 (2012), 1857–1868.
- [51] BENETOS, E., AND DIXON, S. **A shift-invariant latent variable model for automatic music transcription.** *Computer Music Journal* 36, 4 (2012), 81–94.
- [52] BÖCK, S., KREBS, F., AND SCHEDL, M. **Evaluating the online capabilities of onset detection methods.** In *ISMIR* (2012), pp. 49–54.
- [53] BOSCH, J. J., JANER, J., FUHRMANN, F., AND HERRERA, P. **A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals.** In *ISMIR* (2012), pp. 559–564.
- [54] BOULANGER-LEWANDOWSKI, N., BENGIO, Y., AND VINCENT, P. **Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription.** *arXiv preprint arXiv:1206.6392* (2012).
- [55] COLLINS, M. **Pro Tools for music production: recording, editing and mixing.** Taylor & Francis, 2012.
- [56] FUENTES, B., BADEAU, R., AND RICHARD, G. **Blind harmonic adaptive decomposition applied to supervised source separation.** In *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)* (2012), pp. 2654–2658.
- [57] HINTON, G., DENG, L., YU, D., DAHL, G. E., MOHAMED, A.-R., JAITLY, N., SENIOR, A., VANHOUCHE, V., NGUYEN, P., SAINATH, T. N., AND OTHERS. **Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups.** *IEEE Signal processing magazine* 29, 6 (2012), 82–97.
- [58] LEE, S., KIM, J., AND LEE, I. **Speech/audio signal classification using spectral flux pattern recognition.** In *2012 IEEE Workshop on Signal Processing Systems* (2012), pp. 232–236.
- [59] MCFEE, B., BARRINGTON, L., AND LANCKRIET, G. **Learning content similarity for music recommendation.** *IEEE transactions on audio, speech, and language processing* 20, 8 (2012), 2207–2218.
- [60] PUGIN, L., KEPPEL, J., ROLAND, P., HARTWIG, M., AND HANKINSON, A. **Separating presentation and content in mei.** In *ISMIR* (2012), Citeseer, pp. 505–510.
- [61] SALAMON, J., AND GÓMEZ, E. **Melody extraction from polyphonic music signals using pitch contour characteristics.** *IEEE transactions on audio, speech, and language processing* 20, 6 (2012), 1759–1770.

- [62] CHANDWADKAR, D., AND SUTAONE, M. **Selecting proper features and classifiers for accurate identification of musical instruments.** *International Journal of Machine Learning and Computing* 3, 2 (2013), 172.
- [63] DENG, L., LI, J., HUANG, J.-T., YAO, K., YU, D., SEIDE, F., SELTZER, M., ZWEIG, G., HE, X., WILLIAMS, J., AND OTHERS. **Recent advances in deep learning for speech research at microsoft.** In *2013 IEEE international conference on acoustics, speech and signal processing* (2013), IEEE, pp. 8604–8608.
- [64] DENG, L., AND LI, X. **Machine learning paradigms for speech recognition: An overview.** *IEEE Transactions on Audio, Speech, and Language Processing* 21, 5 (2013), 1060–1089.
- [65] GRAVES, A., MOHAMED, A.-R., AND HINTON, G. **Speech recognition with deep recurrent neural networks.** In *2013 IEEE international conference on acoustics, speech and signal processing* (2013), IEEE, pp. 6645–6649.
- [66] HUMPHREY, E. J., BELLO, J. P., AND LECUN, Y. **Feature learning and deep architectures: New directions for music informatics.** *Journal of Intelligent Information Systems* 41 (2013), 461–481.
- [67] KOS, M., KAČIČ, Z., AND VLAJ, D. **Acoustic classification and segmentation using modified spectral roll-off and variance-based features.** *Digital Signal Processing* 23, 2 (2013), 659–674.
- [68] LIU, F., XU, Y., PROM-ON, S., AND YU, A. C. **Morpheme-like prosodic functions: Evidence from acoustic analysis and computational modeling.** *Journal of Speech Sciences* 3, 1 (2013), 85–140.
- [69] SIG, T. M. **Tei with music notation.** Tech. rep., [Online], 2013.
- [70] STURM, B. L. **The gtzan dataset: Its contents, its faults, their effects on evaluation, and its future use.** *arXiv preprint arXiv:1306.1461* (2013).
- [71] VAN DEN OORD, A., DIELEMAN, S., AND SCHRAUWEN, B. **Deep content-based music recommendation.** *Advances in neural information processing systems* 26 (2013).
- [72] XU, Y. **Prosodypro — a tool for large-scale systematic prosody analysis.** In *in Proceedings of Tools and Resources for the Analysis of Speech Prosody* (01 2013).
- [73] HUMPHREY, E. J., AND BELLO, J. P. **From music audio to chord tablature: Teaching deep convolutional networks to play guitar.** In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2014), pp. 6974–6978.

- [74] HUMPHREY, E. J., SALAMON, J., NIETO, O., FORSYTH, J., BITTNER, R. M., AND BELLO, J. P. **Jams: A json annotated music specification for reproducible mir research.** In *ISMIR* (2014), pp. 591–596.
- [75] PANAGAKIS, Y., KOTROPOULOS, C. L., AND ARCE, G. R. **Music genre classification via joint sparse low-rank representation of audio features.** *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22, 12 (2014), 1905–1917.
- [76] PUGIN, L., ZITELLINI, R., AND ROLAND, P. **Verovio: A library for Engraving MEI Music Notation into SVG.** In *Proceedings of the 15th International Society for Music Information Retrieval Conference* (Taipei, Taiwan, Oct. 2014), ISMIR, pp. 107–112.
- [77] SU, L., AND YANG, Y.-H. **Power-scaled spectral flux and peak-valley group-delay methods for robust musical onset detection.** In *ICMC* (2014).
- [78] DAI, J., LIU, W., NI, C., DONG, L., AND YANG, H. **“multilingual” deep neural network for music genre classification.** In *Sixteenth annual conference of the international speech communication association* (2015).
- [79] IOFFE, S., AND SZEGEDY, C. **Batch normalization: Accelerating deep network training by reducing internal covariate shift.** In *International conference on machine learning* (2015), pmlr, pp. 448–456.
- [80] SCHEDL, M., KNEES, P., MCFEE, B., BOGDANOV, D., AND KAMINSKAS, M. **Music recommender systems.** *Recommender systems handbook* (2015), 453–492.
- [81] SZEGEDY, C., LIU, W., JIA, Y., SERMANET, P., REED, S., ANGUELOV, D., ERHAN, D., VANHOUCHE, V., AND RABINOVICH, A. **Going deeper with convolutions.** In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 1–9.
- [82] AMODEI, D., ANANTHANARAYANAN, S., ANUBHAI, R., BAI, J., BATTENBERG, E., CASE, C., CASPER, J., CATANZARO, B., CHENG, Q., CHEN, G., AND OTHERS. **Deep speech 2: End-to-end speech recognition in english and mandarin.** In *International conference on machine learning* (2016), PMLR, pp. 173–182.
- [83] BURGHARDT, M., LAMM, L., LECHLER, D., SCHNEIDER, M., AND SEMMELMANN, T. **Tool-based identification of melodic patterns in musicxml documents.** In *11th Annual International Conference of the Alliance of Digital Humanities Organizations, DH 2016, Krakow, Poland, July 11-16, 2016, Conference Abstracts* (2016), M. Eder and J. Rybicki, Eds., Alliance of Digital Humanities Organizations (ADHO), pp. 440–442.

- [84] CAPELLE, I., AND RICHTS, K. **Die welt des detmolder hoftheaters—erschlossen mit mei und tei.** *Bibliotheksdienst* 50, 2 (2016), 199–209.
- [85] DEFFERRARD, M., BENZI, K., VANDERGHEYNST, P., AND BRESSON, X. **Fma: A dataset for music analysis.** *arXiv preprint arXiv:1612.01840* (2016).
- [86] GUIDO, R. C. **Zcr-aided neurocomputing: A study with applications.** *Knowledge-Based Systems* 105 (2016), 248–269.
- [87] HAMANAKA, M., HIRATA, K., AND TOJO, S. **Implementing methods for analysing music based on Ierdahl and Jackendoff’s generative theory of tonal music.** *Computational music analysis* (2016), 221–249.
- [88] HE, K., ZHANG, X., REN, S., AND SUN, J. **Deep residual learning for image recognition.** In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 770–778.
- [89] LOSTANLEN, V., AND CELLA, C.-E. **Deep convolutional networks on the pitch spiral for musical instrument recognition.** *arXiv preprint arXiv:1605.06644* (2016).
- [90] MARCHAND, U., AND PEETERS, G. **The extended ballroom dataset.** Tech. rep., [Online], 2016.
- [91] MRAD, N. A. **Éléments de sémiotique modale. Essai d’une grammaire musicale pour les traditions monodiques.** Éditions Geuthner et Éditions de l’Université Antonine, 2016.
- [92] NANNI, L., COSTA, Y. M., LUMINI, A., KIM, M. Y., AND BAEK, S. R. **Combining visual and acoustic features for music genre classification.** *Expert Systems with Applications* 45 (2016), 108–117.
- [93] OORD, A. V. D., DIELEMAN, S., ZEN, H., SIMONYAN, K., VINYALS, O., GRAVES, A., KALCHBRENNER, N., SENIOR, A., AND KAVUKCUOGLU, K. **Wavenet: A generative model for raw audio.** *arXiv preprint arXiv:1609.03499* (2016).
- [94] PEZOA, F., REUTTER, J. L., SUAREZ, F., UGARTE, M., AND VRGOČ, D. **Foundations of json schema.** In *Proceedings of the 25th international conference on World Wide Web* (2016), pp. 263–273.
- [95] SIGTIA, S., BENETOS, E., AND DIXON, S. **An end-to-end neural network for polyphonic piano music transcription.** *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 24, 5 (2016), 927–939.

- [96] SZEGEDY, C., VANHOUCKE, V., IOFFE, S., SHLENS, J., AND WOJNA, Z. **Rethinking the inception architecture for computer vision**. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 2818–2826.
- [97] ZHANG, W., LEI, W., XU, X., AND XING, X. **Improved music genre classification with convolutional neural networks**. In *Interspeech* (2016), pp. 3304–3308.
- [98] ÇANO, E., MORISIO, M., AND OTHERS. **Music mood dataset creation based on last.fm tags**. In *2017 International Conference on Artificial Intelligence and Applications, Vienna, Austria* (2017), pp. 15–26.
- [99] CHEN, Z., LUO, Y., AND MESGARANI, N. **Deep attractor network for single-microphone speaker separation**. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2017), IEEE, pp. 246–250.
- [100] CHERFI, S. S.-S., GUILLOTTEL, C., HAMDI, F., RIGAUX, P., AND TRAVERS, N. **Ontology-based annotation of music scores**. In *Proceedings of the Knowledge Capture Conference* (New York, NY, USA, 2017), K-CAP 2017, Association for Computing Machinery.
- [101] CHOI, K., FAZEKAS, G., CHO, K., AND SANDLER, M. **A tutorial on deep learning for music information retrieval**. *arXiv preprint arXiv:1709.04396* (2017).
- [102] CHOI, K., FAZEKAS, G., SANDLER, M., AND CHO, K. **Transfer learning for music classification and regression tasks**. *arXiv preprint arXiv:1703.09179* (2017).
- [103] DAMMANN, T., AND HAUGH, K. **Genre classification of spotify songs using lyrics, audio previews, and album artwork**. *CS229 Final Project, Stanford University, Fall* (2017).
- [104] ENGEL, J., RESNICK, C., ROBERTS, A., DIELEMAN, S., NOROUZI, M., ECK, D., AND SIMONYAN, K. **Neural audio synthesis of musical notes with wavenet autoencoders**. In *International Conference on Machine Learning* (2017), PMLR, pp. 1068–1077.
- [105] FREITAG, M., AMIRIPARIAN, S., PUGACHEVSKIY, S., CUMMINS, N., AND SCHULLER, B. **audeep: Unsupervised learning of representations from audio with deep recurrent neural networks**. *The Journal of Machine Learning Research* 18, 1 (2017), 6340–6344.
- [106] GEMMEKE, J. F., ELLIS, D. P., FREEDMAN, D., JANSEN, A., LAWRENCE, W., MOORE, R. C., PLAKAL, M., AND RITTER, M. **Audio set: An ontology and human-labeled dataset for audio events**. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (2017), IEEE, pp. 776–780.



- [107] HAN, Y., KIM, J., AND LEE, K. **Deep convolutional neural networks for predominant instrument recognition in polyphonic music.** *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25, 1 (2017), 208–221.
- [108] HERSHEY, S., CHAUDHURI, S., ELLIS, D. P., GEMMEKE, J. F., JANSEN, A., MOORE, R. C., PLAKAL, M., PLATT, D., SAUROUS, R. A., SEYBOLD, B., AND OTHERS. **Cnn architectures for large-scale audio classification.** In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2017), IEEE, pp. 131–135.
- [109] HUANG, G., LIU, Z., VAN DER MAATEN, L., AND WEINBERGER, K. Q. **Densely connected convolutional networks.** In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 4700–4708.
- [110] JONES, J., DE SIQUEIRA BRAGA, D., TERTULIANO, K., AND KAUPPINEN, T. **Musical: The music score ontology.** In *Proceedings of the International Conference on Web Intelligence* (New York, NY, USA, 2017), WI '17, Association for Computing Machinery, p. 1222–1229.
- [111] LIN, T.-Y., DOLLÁR, P., GIRSHICK, R., HE, K., HARIHARAN, B., AND BELONGIE, S. **Feature pyramid networks for object detection.** In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 2117–2125.
- [112] O'HANLON, K., EWERT, S., PAUWELS, J., AND SANDLER, M. B. **Improved template based chord recognition using the crp feature.** In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2017), IEEE, pp. 306–310.
- [113] ORAMAS, S., NIETO, O., BARBIERI, F., AND SERRA, X. **Multi-label music genre classification from audio, text, and images using deep features.** *arXiv preprint arXiv:1707.04916* (2017).
- [114] PARK, J., LEE, J., PARK, J., HA, J.-W., AND NAM, J. **Representation learning of music using artist labels.** *arXiv preprint arXiv:1710.06648* (2017).
- [115] SIMON, I., AND OORE, S. **Performance rnn: Generating music with expressive timing and dynamics.** *Magenta Blog* (2017), 16.
- [116] SZEGEDY, C., IOFFE, S., VANHOUCHE, V., AND ALEMI, A. **Inception-v4, inception-resnet and the impact of residual connections on learning.** In *Proceedings of the AAAI conference on artificial intelligence* (2017), vol. 31.
- [117] THAMMASAN, N., FUKUI, K.-I., AND NUMAO, M. **Multimodal fusion of eeg and musical features in music-emotion recognition.** In *Proceedings of the AAAI Conference on Artificial Intelligence* (2017), vol. 31.

- [118] VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, Ł., AND POLOSUKHIN, I. **Attention is all you need**. *Advances in neural information processing systems* 30 (2017).
- [119] ASMAR, M., ATÉCHIAN, T., MRAD, N. A., AND MARTIN, S. L. **Traditional Modal Monodies Generative Grammar Encoding in the Music Encoding Initiative**. In *Proceedings of the International Conference on Technologies for Music Notation and Representation* (May 2018), Concordia University, pp. 95–103.
- [120] BAEVSKI, A., AND AULI, M. **Adaptive input representations for neural language modeling**. *arXiv preprint arXiv:1809.10853* (2018).
- [121] BERNDT, A., WALOSCHEK, S., AND HADJAKOS, A. **Meico: A converter framework for bridging the gap between digital music editions and its applications**. *Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion* (2018).
- [122] DONG, H.-W., HSIAO, W.-Y., YANG, L.-C., AND YANG, Y.-H. **Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment**. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2018), vol. 32.
- [123] FONSECA, E., PLAKAL, M., FONT, F., ELLIS, D. P., FAVORY, X., PONS, J., AND SERRA, X. **General-purpose tagging of freesound audio with audioset labels: Task description, dataset, and baseline**. *arXiv preprint arXiv:1807.09902* (2018).
- [124] GHOSAL, D., AND KOLEKAR, M. H. **Music genre recognition using deep neural networks and transfer learning**. In *Interspeech* (2018), pp. 2087–2091.
- [125] GÓMEZ, J. S., ABESSER, J., AND CANO, E. **Jazz Solo Instrument Classification with Convolutional Neural Networks, Source Separation, and Transfer Learning**. In *Proceedings of the 19th International Society for Music Information Retrieval Conference* (Paris, France, Sept. 2018), ISMIR, pp. 577–584.
- [126] HUANG, C.-Z. A., VASWANI, A., USZKOREIT, J., SHAZEER, N., SIMON, I., HAWTHORNE, C., DAI, A. M., HOFFMAN, M. D., DINCULESCU, M., AND ECK, D. **Music transformer**. *arXiv preprint arXiv:1809.04281* (2018).
- [127] HUNG, Y.-N., AND YANG, Y.-H. **Frame-level instrument recognition by timbre and pitch**. *arXiv preprint arXiv:1806.09587* (2018).
- [128] KARUNAKARAN, N., AND ARYA, A. **A scalable hybrid classifier for music genre classification using machine learning concepts and spark**. In *2018 International Conference on Intelligent Autonomous Systems (ICoIAS)* (2018), IEEE, pp. 128–135.

- [129] KIM, D., SUNG, T. T., CHO, S. Y., LEE, G., AND SOHN, C. B. **A single predominant instrument recognition of polyphonic music using cnn-based timbre analysis.** *International Journal of Engineering & Technology* 7, 3.34 (2018), 590.
- [130] KIM, J., WON, M., SERRA, X., AND LIEM, C. C. **Transfer learning of artist group factors to musical genre classification.** In *Companion Proceedings of the The Web Conference 2018* (2018), pp. 1929–1934.
- [131] ORAMAS, S., BARBIERI, F., NIETO CABALLERO, O., AND SERRA, X. **Multimodal deep learning for music genre classification.** *Transactions of the International Society for Music Information Retrieval*. 2018; 1 (1): 4-21. (2018).
- [132] OUSSIDI, A., AND ELHASSOUNY, A. **Deep generative models: Survey.** In *2018 International conference on intelligent systems and computer vision (ISCV)* (2018), IEEE, pp. 1–8.
- [133] ROBERTS, A., ENGEL, J., RAFFEL, C., HAWTHORNE, C., AND ECK, D. **A hierarchical latent vector model for learning long-term structure in music.** In *International conference on machine learning* (2018), PMLR, pp. 4364–4373.
- [134] SCHLÜTER, J., AND LEHNER, B. **Zero-mean convolutions for level-invariant singing voice detection.** In *International Society for Music Information Retrieval Conference* (2018).
- [135] SCHREIBER, H., AND MÜLLER, M. **A single-step approach to musical tempo estimation using a convolutional neural network.** In *Ismir* (2018), pp. 98–105.
- [136] SERIZEL, R., TURPAULT, N., EGHBAL-ZADEH, H., AND SHAH, A. P. **Large-scale weakly labeled semi-supervised sound event detection in domestic environments.** *arXiv preprint arXiv:1807.10501* (2018).
- [137] STOLAR, M. N., LECH, M., STOLAR, S. J., AND ALLEN, N. B. **Detection of adolescent depression from speech using optimised spectral roll-off parameters.** *Biomedical Journal* 2 (2018), 10.
- [138] VALERIO, V. D., PEREIRA, R. M., COSTA, Y. M., BERTOINI, D., AND SILLA JR, C. N. **A resampling approach for imbalanceness on music genre classification using spectrograms.** In *The Thirty-First International Flairs Conference* (2018).
- [139] XI, Q., BITTNER, R. M., PAUWELS, J., YE, X., AND BELLO, J. P. **Guitarset: A dataset for guitar transcription.** In *International Society for Music Information Retrieval Conference* (2018).

- [140] ZHANG, W., CHEN, Z., AND YIN, F. **Melody extraction using chroma-level note tracking and pitch mapping**. *Applied Sciences* 8, 9 (2018), 1618.
- [141] BIAN, W., WANG, J., ZHUANG, B., YANG, J., WANG, S., AND XIAO, J. **Audio-based music classification with densenet and data augmentation**. In *PRICAI 2019: Trends in Artificial Intelligence: 16th Pacific Rim International Conference on Artificial Intelligence, Cuvu, Yanuca Island, Fiji, August 26-30, 2019, Proceedings, Part III 16* (2019), Springer, pp. 56–65.
- [142] HAWTHORNE, C., STASYUK, A., ROBERTS, A., SIMON, I., HUANG, C.-Z. A., DIELEMAN, S., ELSEEN, E., ENGEL, J., AND ECK, D. **Enabling factorized piano music modeling and generation with the MAESTRO dataset**. In *International Conference on Learning Representations* (2019).
- [143] TAN, M., AND LE, Q. **Efficientnet: Rethinking model scaling for convolutional neural networks**. In *International conference on machine learning* (2019), PMLR, pp. 6105–6114.
- [144] WANG, Q., LI, B., XIAO, T., ZHU, J., LI, C., WONG, D. F., AND CHAO, L. S. **Learning deep transformer models for machine translation**. *arXiv preprint arXiv:1906.01787* (2019).
- [145] WANG, Y., LI, J., AND METZE, F. **A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling**. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2019), IEEE, pp. 31–35.
- [146] WIGGINS, A., AND KIM, Y. **Guitar Tablature Estimation with a Convolutional Neural Network**. In *Proceedings of the 20th International Society for Music Information Retrieval Conference* (Delft, The Netherlands, Nov. 2019), ISMIR, pp. 284–291.
- [147] ZHANG, C., ZHANG, Y., AND CHEN, C. **Songnet: Real-time music classification**, 2019.
- [148] BYAMBATSOGT, G., CHOIMAA, L., AND KOUTAKI, G. **Guitar chord sensing and recognition using multi-task learning and physical data augmentation with robotics**. *Sensors* 20, 21 (2020), 6077.
- [149] DOSOVITSKIY, A., BEYER, L., KOLESNIKOV, A., WEISSENBORN, D., ZHAI, X., UNTERTHINER, T., DEGHANI, M., MINDERER, M., HEIGOLD, G., GELLY, S., AND OTHERS. **An image is worth 16x16 words: Transformers for image recognition at scale**. *arXiv preprint arXiv:2010.11929* (2020).

- [150] EL ACHKAR, C., AND ATÉCHIAN, T. **Supporting music pattern retrieval and analysis: An ontology-based approach.** In *Proceedings of the 10th International Conference on Web Intelligence, Mining and Semantics* (New York, NY, USA, 2020), WIMS 2020, Association for Computing Machinery, p. 17–20.
- [151] ELENA SCHILKE, IRMLIND CAPELLE, K. R. **Das modell — detmolder hoftheater.** Tech. rep., [Online], 2020.
- [152] GAN, C., HUANG, D., ZHAO, H., TENENBAUM, J. B., AND TORRALBA, A. **Music gesture for visual sound separation.** In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 10478–10487.
- [153] GHILDIAL, A., SINGH, K., AND SHARMA, S. **Music genre classification using machine learning.** In *2020 4th international conference on electronics, communication and aerospace technology (ICECA)* (2020), IEEE, pp. 1368–1372.
- [154] HUANG, Y.-S., AND YANG, Y.-H. **Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions.** In *Proceedings of the 28th ACM international conference on multimedia* (2020), pp. 1180–1188.
- [155] MANILOW, E., SEETHARAMAN, P., AND PARDO, B. **Simultaneous separation and transcription of mixtures with multiple polyphonic and percussive instruments.** In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2020), IEEE, pp. 771–775.
- [156] PALANISAMY, K., SINGHANIA, D., AND YAO, A. **Rethinking cnn models for audio classification.** *arXiv preprint arXiv:2007.11154* (2020).
- [157] RACHARLA, K., KUMAR, V., JAYANT, C. B., KHAIRKAR, A., AND HARISH, P. **Pre-dominant musical instrument classification based on spectral features.** In *2020 7th International Conference on Signal Processing and Integrated Networks (SPIN)* (2020), IEEE, pp. 617–622.
- [158] WIGGINS, A., AND KIM, Y. **Towards unsupervised acoustic guitar transcription.** *Journal*, vol 7, 2 (2020), 43–55.
- [159] YU, Y., LUO, S., LIU, S., QIAO, H., LIU, Y., AND FENG, L. **Deep attention based music genre classification.** *Neurocomputing* 372 (2020), 84–91.
- [160] YU, Y., LUO, S., LIU, S., QIAO, H., LIU, Y., AND FENG, L. **Deep attention based music genre classification.** *Neurocomputing* 372 (2020), 84–91.
- [161] ZHANG, B., SUN, L., SONG, Y., SHAO, W., GUO, Y., AND YUAN, F. **Deepfirenet: A real-time video fire detection method based on multi-feature fusion.** *Mathematical biosciences and engineering* 17, 6 (2020), 7804–7818.

- [162] ZHOU, M., NG, M., CAI, Z., AND CHEUNG, K. C. **Self-attention-based fully-inception networks for continuous sign language recognition.** In *ECAI 2020*. IOS Press, 2020, pp. 2832–2839.
- [163] CHAKHTOUNA, A., SEKKATE, S., AND ADIB, A. **Improving speech emotion recognition system using spectral and prosodic features.** In *International Conference on Intelligent Systems Design and Applications (2021)*, Springer, pp. 399–409.
- [164] DAIR, Z., DONOVAN, R., AND O'REILLY, R. **Linguistic and gender variation in speech emotion recognition using spectral features.** *arXiv preprint arXiv:2112.09596* (2021).
- [165] EL ACHKAR, C., COUTURIER, R., ATÉCHIAN, T., AND MAKHOUL, A. **Combining reduction and dense blocks for music genre classification.** In *Neural Information Processing (Cham, 2021)*, T. Mantoro, M. Lee, M. A. Ayu, K. W. Wong, and A. N. Hidayanto, Eds., Springer International Publishing, pp. 752–760.
- [166] HEIN, E. **Ableton live 11.** *Journal of the American Musicological Society* 74, 1 (2021), 214–225.
- [167] KHASGIWALA, Y., AND TAILOR, J. **Vision transformer for music genre classification using mel-frequency cepstrum coefficient.** In *2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GU-CON) (2021)*, IEEE, pp. 1–5.
- [168] LIU, C., FENG, L., LIU, G., WANG, H., AND LIU, S. **Bottom-up broadcast neural network for music genre classification.** *Multimedia Tools and Applications* 80 (2021), 7313–7331.
- [169] LIU, Z., LIN, Y., CAO, Y., HU, H., WEI, Y., ZHANG, Z., LIN, S., AND GUO, B. **Swin transformer: Hierarchical vision transformer using shifted windows.** In *Proceedings of the IEEE/CVF international conference on computer vision (2021)*, pp. 10012–10022.
- [170] MAAIVELD, T., DRIEDGER, J., YELA, D., AND MEROÑO-PEÑUELA, A. **Automatic tablature estimation with convolutional neural networks: Approaches and limitations.** *Department of Computer Science, Faculty of Sciences Vrije Universiteit Amsterdam Netherlands* (04 2021).
- [171] MEHTA, J., GANDHI, D., THAKUR, G., AND KANANI, P. **Music genre classification using transfer learning on log-based mel spectrogram.** In *2021 5th International Conference on Computing Methodologies and Communication (ICCMC) (2021)*, IEEE, pp. 1101–1107.

- [172] MUJTABA, G., LEE, S., KIM, J., AND RYU, E.-S. **Client-driven animated gif generation framework using an acoustic feature**. *Multimedia Tools and Applications* (2021), 1–18.
- [173] PANDEYA, Y. R., AND LEE, J. **Deep learning-based late fusion of multimodal information for emotion classification of music video**. *Multimedia Tools and Applications* 80 (2021), 2887–2905.
- [174] SINGH, Y., KUMAR, R., AND BISWAS, A. **Swaragram: Shruti-based chromagram for indian classical music**. In *Advances in Speech and Music Technology: Proceedings of FRSM 2020* (2021), Springer, pp. 109–118.
- [175] SONG, X., QIAO, X., HAO, D., YANG, L., ZHOU, X., XU, Y., AND ZHENG, D. **Automatic recognition of uterine contractions with electrohysterogram signals based on the zero-crossing rate**. *Scientific Reports* 11, 1 (2021), 1956.
- [176] ALNUAIM, A. A., ZAKARIAH, M., SHASHIDHAR, C., HATAMLEH, W. A., TARAIZI, H., SHUKLA, P. K., AND RATNA, R. **Speaker gender recognition based on deep neural networks and resnet50**. *Wireless Communications and Mobile Computing* 2022 (2022), 1–13.
- [177] ALY, M., RAHOUMA, K. H., AND RAMZY, S. M. **Pay attention to the speech: Covid-19 diagnosis using machine learning and crowdsourced respiratory and speech recordings**. *Alexandria Engineering Journal* 61, 5 (2022), 3487–3500.
- [178] BAKHTYARI, M., DAVOUDI, S., AND MIRZAEI, S. **Evaluating various feature extraction methods and classification algorithms for music genres classification**. In *2022 27th International Computer Conference, Computer Society of Iran (CSICC)* (2022), pp. 1–6.
- [179] CHEN, K., DU, X., ZHU, B., MA, Z., BERG-KIRKPATRICK, T., AND DUBNOV, S. **Hts-at: A hierarchical token-semantic audio transformer for sound classification and detection**. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2022), IEEE, pp. 646–650.
- [180] CWITKOWITZ, F., HIRVONEN, T., AND KLAUPURI, A. **Fretnet: Continuous-valued pitch contour streaming for polyphonic guitar tablature transcription**. *arXiv preprint arXiv:2212.03023* (2022).
- [181] DAR, J. A., SRIVASTAVA, K. K., AND LONE, S. A. **Spectral features and optimal hierarchical attention networks for pulmonary abnormality detection from the respiratory sound signals**. *Biomedical Signal Processing and Control* 78 (2022), 103905.

- [182] EL ACHKAR, C., AND ATECHIAN, T. **Mei2json: a pre-processing music scores converter**. *International Journal of Intelligent Information and Database Systems* 15 (01 2022), 57.
- [183] FANG, J., LIN, H., CHEN, X., AND ZENG, K. **A hybrid network of cnn and transformer for lightweight image super-resolution**. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022), pp. 1103–1112.
- [184] HEAKL, A., ABDELGAWAD, A., AND PARQUE, V. **A study on broadcast networks for music genre classification**. In *2022 International Joint Conference on Neural Networks (IJCNN)* (2022), IEEE, pp. 1–8.
- [185] JOTHIMANI, S., AND PREMALATHA, K. **Mff-saug: Multi feature fusion with spectrogram augmentation of speech emotion recognition using convolution neural network**. *Chaos, Solitons & Fractals* 162 (2022), 112512.
- [186] KAVITHA, S., AND MANIKANDAN, J. **Improved methodology of svm to classify acoustic signal by spectral centroid**. *Journal of Trends in Computer Science and Smart Technology* 3, 4 (2022), 294–304.
- [187] KHAN, S., NASEER, M., HAYAT, M., ZAMIR, S. W., KHAN, F. S., AND SHAH, M. **Transformers in vision: A survey**. *ACM computing surveys (CSUR)* 54, 10s (2022), 1–41.
- [188] KIM, S., HAYASHI, T., AND TODA, T. **Note-level automatic guitar transcription using attention mechanism**. In *2022 30th European Signal Processing Conference (EUSIPCO)* (2022), IEEE, pp. 229–233.
- [189] LIAO, Y.-J., WANG, W.-C., RUAN, S.-J., LEE, Y.-H., AND CHEN, S.-C. **A music playback algorithm based on residual-inception blocks for music emotion classification and physiological information**. *Sensors* 22, 3 (2022).
- [190] MA, Y., SONG, Y., AND GAO, F. **A novel hybrid cnn-transformer model for eeg motor imagery classification**. In *2022 International Joint Conference on Neural Networks (IJCNN)* (2022), pp. 1–8.
- [191] REGHUNATH, L. C., AND RAJAN, R. **Transformer-based ensemble method for multiple predominant instruments recognition in polyphonic music**. *EURASIP Journal on Audio, Speech, and Music Processing* 2022, 1 (2022), 11.
- [192] SOROUSH, P., HERFF, C., RIES, S., SHIH, J., SCHULTZ, T., AND KRUSIENSKI, D. **Contributions of stereotactic eeg electrodes in grey and white matter to speech activity detection**. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)* (2022), IEEE, pp. 4789–4792.



- [193] SUMAN, S., SAHOO, K. S., DAS, C., JHANJHI, N., AND MITRA, A. **Visualization of audio files using librosa**. In *Proceedings of 2nd International Conference on Mathematical Modeling and Computational Science: ICMACS 2021* (2022), Springer, pp. 409–418.
- [194] TIPLE, B., AND PATWARDHAN, M. **Multi-label emotion recognition from indian classical music using gradient descent snn model**. *Multimedia Tools and Applications* 81, 6 (2022), 8853–8870.
- [195] YANG, Y.-Y., HIRA, M., NI, Z., ASTAFUROV, A., CHEN, C., PUHRSCHE, C., POLLACK, D., GENZEL, D., GREENBERG, D., YANG, E. Z., AND OTHERS. **TorchAudio: Building blocks for audio and speech processing**. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2022), IEEE, pp. 6982–6986.
- [196] CORREYA, A. A., BOGDANOV, D., ALONSO JIMÉNEZ, P., AND SERRA, X. **Essentia api: a web api for music audio analysis**. Tech. rep., [Online], 2023.
- [197] HAN, X., CHEN, F., AND BAN, J. **Music emotion recognition based on a neural network with an inception-gru residual structure**. *Electronics* 12, 4 (2023).
- [198] LU, W.-T., WANG, J.-C., AND HUNG, Y.-N. **Multitrack music transcription with a time-frequency perceiver**. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2023), IEEE, pp. 1–5.
- [199] MAAZ, M., SHAKER, A., CHOLAKKAL, H., KHAN, S., ZAMIR, S. W., ANWER, R. M., AND SHAHBAZ KHAN, F. **Edgenext: Efficiently amalgamated cnn-transformer architecture for mobile vision applications**. In *Computer Vision – ECCV 2022 Workshops* (Cham, 2023), L. Karlinsky, T. Michaeli, and K. Nishino, Eds., Springer Nature Switzerland, pp. 3–20.
- [200] SCHMID, F., KOUTINI, K., AND WIDMER, G. **Efficient large-scale audio tagging via transformer-to-cnn knowledge distillation**. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2023), IEEE, pp. 1–5.
- [201] YANG, H., AND YANG, D. **Cswin-pnet: A cnn-swin transformer combined pyramid network for breast lesion segmentation in ultrasound images**. *Expert Systems with Applications* 213 (2023), 119024.
- [202] YUAN, F., ZHANG, Z., AND FANG, Z. **An effective cnn and transformer complementary network for medical image segmentation**. *Pattern Recognition* 136 (2023), 109228.

- [203] ZENG, C., AND KWONG, S. **Combining cnn and transformers for full-reference and no-reference image quality assessment.** *Neurocomputing* 549 (2023), 126437.



# LIST OF FIGURES

2.1	Waveform Visualization of a Blues Track in the GTZAN Dataset . . . . .	20
2.2	Mel-Frequency Cepstral Coefficients Visualization of a Blues Track in the GTZAN Dataset . . . . .	21
2.3	STFT Spectrogram Visualization of a Blues Track in the GTZAN Dataset . .	22
2.4	CQT Spectrogram Visualization of a Blues Track in the GTZAN Dataset . .	23
2.5	Chromagram Visualization of a Blues Track in the GTZAN Dataset . . . . .	24
2.6	Common Audio Features Visualization of a Blues Track in the GTZAN Dataset . . . . .	25
3.1	MusicPatternOWL - General Overview. . . . .	45
3.2	A sample of a music score encoded in MEI (a) then rendered to SVG (b) throughout the analysis. . . . .	46
3.3	SVG output from MM analyzer. . . . .	48
4.1	MEI2JSON Main Components Overview . . . . .	54
4.2	MEI2JSON Activity Diagram . . . . .	59
4.3	Meico + MusicJSON Activity Diagram . . . . .	61
4.4	Time Complexity Chart . . . . .	64
4.5	Space Complexity Chart - Meico+MusicJSON . . . . .	66
4.6	Space Complexity Chart - MEI2JSON . . . . .	66
4.7	Histogram - Data Quality Metrics . . . . .	67
5.1	Spectrogram slicing approach . . . . .	77
5.2	Proposed inception block modifications over the BBNN network . . . . .	78
5.3	STFT greyscale spectrogram of a Blues track computed using Librosa . . .	82
5.4	STFT greyscale spectrogram of a Blues track computed using SoX . . . . .	82

5.5	Comparison by genre of the proposed approach with BBNN using the GTZAN dataset . . . . .	84
6.1	Music score and tablature notation illustrating the first four bars of the song <i>Radioactive</i> by <i>Imagine Dragons</i> . The tablatures in the bottom represent the string to be played by the guitarist, in addition to the number of the fret to press. . . . .	88
6.2	Audio to Image transformation through Constant-Q Transform computation	90
6.3	Label associated to the 512th frame of the 02_Rock_1_130.wav recording .	91
6.4	Architecture of the TabInception network . . . . .	92
6.5	Architecture of the Inception Transformer (InT) network . . . . .	93

# LIST OF TABLES

4.1	Storage Overview Table. . . . .	70
5.1	Comparative table for GTZAN classification methods in terms of accuracy (%) . . . . .	80
5.2	Comparative table for FMA classification methods in terms of accuracy (%)	81
6.1	Comparative table for guitar tablature transcription using computer vision networks. The best score per metric is highlighted in <b>black</b> , the second best in <b>green</b> , and the third best in <b>red</b> . . . . .	95







**Title:** Music Encoding and Deep Learning for Music Transcription and Classification based on Visually Represented Audio Features

**Keywords:** Ontology, Music Score Converter, Music Genre Classification, Spectrogram, Deep Learning, Computer Vision, Automatic Music Transcription

**Abstract:**

In the past decade, musicians have composed new occidental-style music scores, often encoded in XML-based formats for analysis. However, the oriental genre lacks XML support due to limited digitization interest, resulting in encoding challenges. The rapid growth of deep learning has stimulated the exploration of music applications. This thesis focuses on music genre classification and transcription using Deep Learning. Our efforts include: (1) Introducing MusicPatternOWL, an ontology for structuring knowledge extraction in Eastern music pattern analysis. (2) Developing the MEI2JSON converter to transform MEI-encoded scores into simplified JSON for Artificial Intelligence pre-training. (3) Proposing a pre-processing method for Short Time Fourier Transform (STFT) spectrograms and enhancing a CNN-based music genre classifier to examine deep learning's impact on music streaming services. (4) Presenting CNN-based and CNN-Transformer-based networks for guitar tablature transcription, evaluated against the state-of-the-art networks in the field.

**Titre :** Encodage de musique et apprentissage en profondeur pour la transcription et la classification de la musique basées sur des caractéristiques audio représentées visuellement

**Mots-clés :** Ontologie, Convertisseur de partition musicale, Classification des genres musicaux, Spectrogramme, L'apprentissage en profondeur, Vision par ordinateur, Transcription automatique de la musique

**Résumé :**

Au cours de la dernière décennie, de nouvelles partitions occidentales ont été régulièrement composées et encodées en XML pour l'analyse. Pour les partitions orientales, l'absence de support XML est due au manque d'intérêt pour la numérisation, entraînant une difficulté d'encodage précis. L'expansion rapide de l'apprentissage profond motive les développeurs et musiciens à explorer ses avantages en musique. Cette thèse se concentre sur la classification musicale et la transcription automatique grâce à l'apprentissage profond. Nos efforts incluent : (1) La proposition de MusicPatternOWL, une ontologie structurant l'extraction de connaissances pour l'analyse de partitions orientales. (2) Le développement du convertisseur MEI2JSON pour transformer les partitions MEI en JSON simplifié en vue du prétraitement de l'Intelligence Artificielle. (3) L'introduction d'une méthode de prétraitement pour les spectrogrammes STFT et l'amélioration d'un classificateur musical CNN, pour évaluer l'impact de l'apprentissage profond sur les services de streaming musical. (4) La présentation de réseaux basés sur CNN et CNN-Transformateur pour la transcription de tablatures de guitare, évalués par rapport aux réseaux de pointe dans le domaine.