Unveiling the Art of Piano Performance:
A Study of Pianist Identification through Statistical and Hierarchical Models

Syed Rifat Mahmud Rafee

A thesis submitted in partial fulfilment of the requirements of the
Degree of Doctor of Philosophy

Centre for Digital Music
School of Electronic Engineering and Computer Science
Queen Mary University of London
United Kingdom

February 2023

Statement of Originality

I, Syed Rifat Mahmud Rafee, hereby confirm that the research contained within this thesis is entirely my own original work. Any ideas or quotations sourced from the works of others, whether published or otherwise, have been fully acknowledged in accordance with the standard referencing practices of the discipline.

I attest that I have exercised reasonable care to ensure that the work is original and does not, to the best of my knowledge, violate any laws of the United Kingdom or infringe upon any third party's copyright or other intellectual property rights, or contain any confidential material.

I also acknowledge that the College has the right to utilize plagiarism detection software to verify the electronic version of the thesis.

Furthermore, I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university. The copyright of this thesis belongs to the author, and no portion of it or information derived from it may be published without the prior written consent of the author.

Abstract

Automatic Performer Identification from the symbolic representation of music has been a challenging topic in the field of Music Information Retrieval (MIR). This thesis proposes various approaches for modeling and identifying musical instrumentalists, with a specific focus on pianist identification, an exceptionally challenging task that often requires trained/expert musicians.

Performers' continuously modifying important parameters like tempo and dynamics to stress specific notes or 'shape' certain passages in the metrically-notated music are what makes them distinctive in their performances. By comparing a performance to its notated score and the performance norm (defined as a quasi-performance calculated by taking the average of all the performances of the same piece), a set of note-level expressive features related to timing, dynamics, and articulation are proposed that are capable of capturing an individual performer's performance traits. To validate the utility of these characteristic features, several statistical models are used to model their distributions, followed by a similarity metric that compares the distribution similarity of a candidate pianist with that of the pianists in the dataset. The identification is done considering the distribution of each individual feature as well as a feature fusion technique. Results show that features related to expressive timing and loudness are the most informative about performers' styles when fused together, followed by note duration.

Hierarchical modelling of music can be useful for performer identification, as it allows to capture the structure and organization of the music. Specifically, Western classical music demonstrates a distinct hierarchical organization of elements (note, beat, measure, phrase level etc.). Utilizing a convolutional neural network (CNN) for learning hierarchical representations of this data is a suitable approach. In this study, a pianist identification model is proposed that employs a multichannel 1D CNN, designed to exploit the hierarchical nature of Western classical music through the utilisation of a beat-specific kernel in the first layer of the CNN, optimised to extract musically salient features. Although the proposed model achieves good precision, it does not incorporate recurrence and, as such, is not aware of the context of the music, which is highly dependent on context.

Central to this research is the creation of the Automatically Transcribed Expressive Piano Performance (ATEPP) dataset. This extensive dataset, comprising 11,742 virtuoso piano recordings spanning over 1,007 hours, serves as a valuable resource. It facilitates the study of performer-specific expressiveness and diverse playing styles in Western classical piano music, providing a substantial foundation for further investigation and analysis.

Finally, to address the limitation of CNNs, a more complex and musically motivated model is proposed that utilizes Recurrent Neural Networks (RNNs) and a multi-head attention mechanism over different hierarchical levels to incorporate recurrence and attention. This facilitates the learning of both the local and global dependencies of the music structure and expressive performance. Results from experimental evaluations reveal that the suggested method outperforms the baseline models, demonstrating the model's discriminative power and ability to learn performer-specific styles.

In summary, this thesis aims to advance performer identification in symbolic music data by uncovering key expressive features, proposing innovative modeling techniques, and introducing a comprehensive dataset. These contributions provide valuable insights and tools for the field of Music Information Retrieval, enhancing our understanding of performer-specific musical styles.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to God, who has guided me throughout this journey and has provided me with the strength and determination to see this Thesis through to completion.

My heartfelt thanks go to my primary supervisor, Dr. George Fazekas, whose unwavering support and expertise in all technical matters have been a constant source of guidance and inspiration throughout my PhD journey. His patience in answering all of my trivial questions and his constant kindness were a beacon of hope during the many challenges I faced. Dr. Fazekas was always there for me, whether I struggled with an experiment and didn't know how to solve it, or when I finished a paper draft and didn't know how to revise it. Over the past four years, through his invaluable guidance, continuous encouragement and support, I have not only strengthened myself personally and professionally, but also grown as a researcher.

I am forever grateful to my Co-Supervisor, Prof. Geraint Wiggins, for opening a door to the field of music technology for me. Despite my background in Electronic Engineering and Computing, he gave me the opportunity to be part of the Centre for Digital Music (C4DM) and to meet people in this fantastic group. He always supported my research with his sage advice and patient instruction. I have significantly benefited from his scientific rigour, broad knowledge, and vast research experience. I could not have undertaken this journey without his help. He has always supported me in my difficult times and inspired me greatly whenever I needed. I would also like to extend my profound gratitude to my independent assessor, Dr. Mathieu Barthet for his meticulous feedbacks in each stage of the PhD. His invaluable insights have served as a foundation for my further proceedings. I couldn't have asked for a more exceptional panel of supervisors.

I would like to express my profound gratitude to all the individuals who have supported my research in a myriad of ways. My sincerest thanks to Professor Mark Sandler, who introduced me to my then supervisor, Professor Geraint Wiggins, and to Professor Simon Dixon for his valuable reviews of my conference paper. My gratitude extends to Dr. Yudong Zhao, for sharing his expertise on violinist identification, and to Dr. Beici Liang, for imparting her knowledge on symbolic music data. I am also indebted to Dr. Nikesh Bajaj for his assistance in setting up servers for my deep learning models, and to Dr. Keunwoo Choi for his invaluable tutorials on machine learning and deep learning in Music Information Retrieval. My appreciation extends to Dr. Emmanuouil Benetos for his reviews on my conference paper and Dr. Lakhal for sharing his knowledge on deep learning models. My heartfelt thanks to the anonymous reviewers of the papers I have submitted for their invaluable feedback.

I gratefully acknowledge the financial support for my research, publications and conference presentations from the Department of Electronic Engineering and Computing at Queen Mary University of London, and from the Centre for Digital Music (C4DM) for funding my travel to conferences.

Special thanks to the members of the C4DM, including Ali, Adrien, Andrew, Charis, Dalia, Dan, Delia, Emir, Elona, Franco, Illaria, Jeff, Jianing, Jiawen, Johan, Lele, Maksud, Pedro, Sami, Saurjya, Soumya, Sebastian, Simon, Tim, Vinod, Wahab, Yukun and so many others. My appreciation extends to Huan and Xingxing, for sharing their ideas on musical expression analysis. I am grateful for the opportunity to collaborate with them, and for the insights I gained on expressive music performance. I would also like to express my deepest gratitude to Nafi Ahmed, who has been a constant source of support and encouragement throughout the years, always there to lend a listening ear or a helping hand. His unwavering loyalty and kindness have made a profound impact on my life, and I am forever grateful for his friendship.

Lastly, my deepest gratitude goes to my parents, for their unwavering love, support, and encouragement throughout my academic journey. Their guidance and sacrifices have been instrumental in my success, and I am forever grateful for the foundation they have provided me. I am so proud to be their son. My thanks also extend to my sisters for their constant love and support, and for being a source of inspiration and motivation throughout my life. I would also like to express my love and appreciation to my wife, Elma

Haque Tory, for her love, patience, and unwavering support. She has been my rock, my confidant, and my best friend, and her love and guidance have been invaluable and have made this journey possible. I would like to express my deepest gratitude to my newborn daughter, Rehnum Syed Amayra, for coming into my life and inspiring me to strive for greatness. Your innocent wonder and boundless energy have been a constant source of motivation for me, and your arrival has made my world a brighter and more beautiful place.

Licence

Contents

List of Figures

List of Tables

List of abbreviations

1D	One-Dimensional
2D	Two-Dimensional
AI	Artificial Intelligence
AAP	Adaptive Average Pooling
AMT	Automatic Music Transcription
ATEPP	Automatically Transcribed Expressive Piano Performance
BPM	Beats per Minute
CV	Cross-Validation
CAM	Class Activation Map
CER	Composition Entity Resolution
CNN	Convolutional Neural Network
CRNN	Convolutional Recurrent Neural Network
C4DM	Centre for Digital Music
DL	Dynamic Level
DNN	Deep Neural Network
EM	Expectation Maximization
FF	Fused Feature
FC	Fully-Connected
FN	False Negative
FP	False Positive
FFN	Feed-Forward Neural Network
FCN	Fully Convolutional Neural Network
GAP	Global Pooling Layer
GMM	Gaussian Mixture Model
GASF	Gramian Angular Summation Field
HN	Hierarchical Network
HE	Hermonic Error

HMM	Hidden Markov Model
HIPI	Hierarchical Performer Identifier
HPCP	Harmonic Pitch Class Profile
IOI	Inter Onset Interval
JS	Jensen-Shannon
KL	Kullbeck Leibler
KDE	Kernel Density Estimation
KNN	K-Nearest Neighbour
LSTM	Long Short-Term Memory
LOGOCV	Leave One Group Out Cross-Validation
MS	Mis-touched Short notes
MOS	Mean Opinion Score
MIR	Music Information Retrieval
MIDI	Musical Instrument Digital Interface
ND	Note Duration
OT	Onset Time
OTD	Off Time Duration
PDF	Probability Density Function
RNN	Recurrent Neural Network
ResCNN	Residual Convolutional Neural Network
ResNet	Residual Network
SF	Stacked Feature
SN	Segmented Note
SVM	Support Vector Machine
TP	True Positive
VAE	Variational Auto Encoder

Chapter 1

Introduction

1.1 Motivation

Music is a medium comprising various musical ideas arranged within a composition in a manner that elicits contrast and surprise, or a sense of familiarity through repetition or alteration. These ideas are closely interconnected and may be decomposed into shorter ideas due to their hierarchical organization. This hierarchical structure, known as musical form, reflects the composer's perspective on the piece and is manifest in the nested organization of sounds. In Western classical music, the lowest level of the hierarchy consists of notes and chords, which are linked together to form higher structural constructs such as measures, motives, and phrases, which define the sections of the composition. This hierarchical structure can be represented visually using a tree representation [3], and it plays a crucial role in the ease with which a piece of music may be classified based on its emotional content.

Music, however, also needs the human performance for rendering the musical ideas into acoustic realisations [4]. As defined by Kendall and Carterette [5], music performance operates as a communicative system wherein composers encode their artistic vision into notation, which is subsequently re-coded by performers through the execution of acoustical signals, and ultimately decoded by listeners as a rendition of the original musical ideas. The realization of these acoustic manifestations is achieved through the interplay of three key elements of performance: interpretation, planning, and movement, which often reflects the cognitive processes engaged by the performer during the execution of a piece of notated music [6]. Interpretation, in particular, pertains to the process of conceptually deciding how to perform precise notations in a way that captures the emotional essence of the music. It is a delicate balance of technical skill and artistic expression, as the performer must navigate the structure of the piece while also adding their own personal touch. However, music research has traditionally provided limited insight into the processes through which performers generate their desired interpretations [7]. Nevertheless, several music theorists hold the perspective that interpretation comprises essential components, including structure, emotion, and physical movement [8, 9]. These elements are considered fundamental contributors to how performers shape their interpretations, shedding light on the complexities of musical expression.

Planning, on the other hand, pertains to the process by which a performer strategises the utilisation of musical structure to convey expression through their unique style. For instance, the utilisation of specific techniques such as variations in dynamics and tempo can serve to accentuate the phrasing structure, thereby emphasizing the way in which the music is divided into phrases, as demonstrated by studies such as [10, 11], and to evoke a specific emotional response. The final aspect of performance, movement, represents the concrete execution of the musical composition by the performer, reflecting their interpretation and the plan. These movements can be considered as embodied interactions between the performer and the music, which have a significant impact on the manner in which the music is performed and perceived [12].

All three above aspects is what constitutes a musical performance through which performers express affective content in the music. These elements, when combined, create a unique and dynamic interpretation of the piece that greatly impacts the listener's perception and enjoyment. The affective content conveyed through a performance is subjective and can vary greatly between different renditions of the same piece, leading to listener preferences and appreciations for diverse interpretations. Virtuoso performers often distinguish themselves in their performances by constantly modifying parameters such as tempo, timing, dynamics, and articulation, which are not prescribed in the notated score but are used to produce an expressive rendition of the composition which is known as expressive performance. The manipulation of these parameters are clearly distinguishable by the listeners and often brings out the dramatic, affective and artistic qualities of performers which in most cases, may affect and connect the listeners emotionally. Thus, a comprehensive analysis of these parameters is imperative for the study of musical expression, providing insight into the performer's distinct stylistic characteristics.

The complexity inherent in the interplay of structural and interpretative factors renders the formalization of expressive piano performance a highly challenging task in the field of Music Information Retrieval (MIR). Despite this, the analysis of musical expression through quantitative methods is vital for various MIR endeavors, including automatic music generation, music transcription, and music recommendation. In addition, this analysis holds immense value in the realm of musical pedagogy and instruction, as it enables the refinement of more efficacious methodologies for teaching, and enables students to learn and measure the similarity of their performance styles with those of renowned pianists. Moreover, quantifying expressive piano performance enables musicologists to delve deeper into the history of piano performance and the impact of various pianistic styles. Furthermore, it facilitates music producers in creating piano pieces that exemplify a wide array of styles by quantifying the characteristic styles of pianists.

The advent of deep generative models has sparked interest among researchers in the realm of music generation, particularly with regards to style specific music generation, where the focus is on emulating the styles of specific composers, genres, or performers. Another area of interest is style specific music rendition, where a raw, non-expressive composition is transformed into an expressive interpretation through the application of learned performance styles derived from a corpus of music. Traditional methods of evaluation for such models have relied on subjective listening tests. While human experience is crucial in evaluating synthesized music, it is acknowledged that individual responses may vary as music evokes different emotions for different individuals. Therefore, to supplement and enhance the evaluation of synthesized style specific music, a quantitative approach that measures the similarity between the generated style and an existing style is deemed necessary.

To address the above problems, the proposed thesis presents a comprehensive methodology for the study of expressive performance modeling and pianist identification. Initially, a dataset of various interpretations of the same composition by different performers is constructed to analyze similarities and differences among performers. Various expressive features are extracted from this dataset based on the domain knowledge of piano playing and statistical models are employed to estimate the distribution of each feature, providing a compact representation of pianists' playing styles. The similarity of these distributions is calculated to identify pianists. In addition, the use of Convolutional Neural Networks (CNNs) for learning hierarchical representations of data is proposed as a musically motivated model for pianist identification. Furthermore, a large-scale, performer-oriented dataset of automatically transcribed expressive piano performances is proposed, allowing for the exploration of performer-specific expressiveness and different schools of playing. Finally, a hierarchical performer identification model is proposed, integrating hierarchical RNNs with hierarchical multi-head attention, representing a novel approach in its ability to model the hierarchical structure of Western musical compositions and the relationship between structural and expressive performance parameters.

In the following section, a comprehensive review of the existing methods for musical performer identification will be undertaken. The organization of the thesis and the primary contributions of the research will be outlined in Section 1.3, and any associated publications will be presented in Section 1.4.

1.2 Prior Research on Performer Identification

In this section, we provide an overview of existing research in the field of performer identification. We start by reviewing the various applications of computational models for automatic identification of music performers. Through a review of previous studies, we aim to identify the challenges and limitations in the current state of research and highlight the potential avenues for future work.

The field of MIR has seen widespread use of computational models in various applications, yet the application of these models for automatic identification of music performers remains an under-investigated area of study. Despite this, prior research has focused on the use of computational models for performer identification. Repp [13] studied the statistical analysis of temporal commonality and diversity of a well-known piece that also demonstrates the distinctiveness of some well-known pianists. Stamatatos and

Widmer [1] developed of a dataset consisting of 22 performers, extracting expressive features, including timing, loudness, articulation and melody lead [14] and proposed a system for automatic performer identification, which employed an ensemble of simple classification algorithms to identify the most likely performer given a performance. Wang [15] and Saunders et al. [16] similarly employed expressive features extracted from audio recordings of piano performances to identify performers. Saunders et al. [17] presented a novel application of string kernel to identify famous pianists based on their unique style of playing. Grachten and Widmer [18] delved into the idea of automatically recognizing pianists based on their execution of ritardandi, utilizing deviations from the performance norm (average performance) to differentiate between pairs of pianists.

In addition to pianist identification, research has been conducted to identify other forms of instrumentalists, though these efforts are limited in scope, likely due to a lack of available large-scale datasets. For instance, Ramirez et al. [19] proposed a machine learning-based approach for identifying jazz saxophonists through the extraction of deviation features from monophonic audio recordings, specifically analyzing variations in pitch, timing, amplitude, and timbre of individual notes. They subsequently expanded upon this work by examining violin performances, utilizing features such as articulation, timing, and amplitude to capture both note-level characteristics and broader musical context central to violinist identification [20]. Additionally, Zhao et al. [21] introduced a transfer learning strategy for identifying violinists, leveraging pre-trained weights from music auto-tagging neural networks and singer identification models.

Apart from instrumentalist identification, there has been research on singer identification as well. Recently, the application of deep learning models has made it possible to identify singers automatically with greater accuracy, as opposed to utilizing hand-crafted features [22]. For example, Nasrullah and Zhao [23] proposed the use of a Convolutional Recurrent Neural Network (CRNN) based network for artist classification utilizing the artist20 dataset [24]. Subsequently, Zhang et al. [25] employed a deep CRNN model with an attention mechanism on the same dataset for singer identification, achieving superior f1-scores. The model learns the local timbre feature representation from the mixture of singer voice and background music, facilitating automatic singer identification. In a study by Kroher and Gómez [22], high-level features related to the performance character were extracted from the predominant fundamental frequency envelope and automatic symbolic transcriptions, in addition to timbre and vibrato descriptor, and a robust system for modelling the singer's typical performance was proposed, which facilitates the identification of singers.

The majority of studies pertaining to pianist identification have been compromised by the limited quantity of data available, thereby constraining the applicability and generalizability of their findings. Additionally, majority of these studies have employed the extraction of manually-engineered performance features for identification, as opposed to utilizing deep learning models, which have been demonstrated to be effective in identifying unique patterns within the input data automatically. Furthermore, there has been a lack of research examining the correlation between musical structural features and performance features through the utilization of mathematical descriptors or deep learning models. As highlighted in Section 1.3, this thesis attempts to address these shortcomings through the presentation of novel methodologies and the subsequent analysis of experimental results.

1.3 Thesis Structure and Contributions

Chapter 2 Background

This chapter provides the technical background of this thesis starting by introducing the topic of musical expression and the factors that influence it. It reviews various techniques and algorithms that have been developed to analyze and replicate the expressive elements of a musical performance using computational modeling approaches. It also focuses on the use of expressive features to quantitatively measure the playing style of a performer. The chapter also provides an overview of the use of statistical models and music similarity measurement algorithms in the analysis of musical expression. Additionally, it covers definitions of machine learning and deep learning algorithms and how they may be used in music information retrieval. Finally, it looks at various evaluation metrics and techniques developed to assess the reliability and performance of computational models.

Chapter 3 Pianist Identification via Probabilistic Density Estimation

This chapter proposes a pianist identification method that utilizes similarity calculation from note-level feature distribution. The method leverages the global distribution of proposed expressive features to characterize the performer's style, and calculates similarity between feature distributions of different performers using KL-divergence. It then performs identification of the performer using this similarity. Three distribution models - Histogram, Kernel Density Estimation (KDE), and Gaussian Mixture Model (GMM) - are evaluated and compared for their identification performance.

Chapter 4 Parametric Learning for Pianist Identification

This chapter proposes a novel approach for pianist identification using a multichannel 1D convolutional neural network (CNN). The model aims to capture the nuanced temporal contexts of piano performances by establishing the CNN's first layer with diverse filter shapes. The model exploits the hierarchical structure of Western classical music by incorporating a beat-specific kernel in the first layer of the CNN, experimenting with varying kernel sizes that align with each beat in the music. This allows the CNN to learn the micro-variations injected by performers within each beat, such as variations in timing, velocity, and articulation.

Chapter 5 Large Scale Dataset Construction

This chapter presents the Automatically Transcribed Expressive Piano Performance (ATEPP) dataset, a comprehensive corpus of 11742 virtuoso piano recordings, spanning a total duration of 1007 hours. The dataset was generated by applying state-of-the-art piano transcription models to audio recordings of performances, as opposed to MIDI files recorded from computer-controlled pianos, enabling the inclusion of a diverse set of performances and examination of performer-specific expressiveness and diverse playing styles. The validity of the transcribed performances was established through an error analysis and listening test of existing transcription models. The dataset serves as a valuable resource for researchers investigating expressiveness and styles in Western classical piano music, and can be utilized for a wide range of tasks including performance feature analysis, comparison of performances and styles, stylistic performance generation, and performance visualization. The compilation of the dataset was a joint endeavor in which I collaborated with two of my colleagues from the Center for Digital Music (C4DM), Huan Zhang and Jingjing Tang. Each party contributed equally to the construction of the dataset. The materials presented in this chapter have been utilized with the express consent of my aforementioned co-authors.

Chapter 6 Hierarchical Performance Modelling for Pianist Identification

This Chapter proposes a novel approach for pianist identification by using a recurrent neural network-based hierarchical performance encoder model. The model first employs a beat-level Long Short-Term Memory (LSTM) encoder that initially encodes performance information at the beat level. The outputs are then summarized by a multi-head attention mechanism, which serves as input to the measure-level LSTM encoder. The model is trained using note-level features derived from calculating the deviations of each performance from a mechanical performance produced by a reference score, with the goal of predicting the most likely pianist. The approach leverages the ability of LSTMs to learn short musical ideas and utilizes a multi-head attention mechanism to address known limitations of LSTMs in learning long-term dependencies.

Chapter 7 Conclusion

This chapter concludes this Thesis and identifies avenues for future research. The ideas and methods presented throughout this thesis have the potential to be expanded upon and further investigated, providing a foundation for future work.

1.4 Associated publications

Portions of the work detailed in this thesis have been presented in national and international scholarly publications, as follows:

- Chapter 3: Syed Rifat Mahmud Rafee, György Fazekas, and Geraint A. Wiggins. Performer identification from symbolic representation of music using statistical models. *International Computer Music Conference (ICMC 2021)*, 2021.

Syed Rifat Mahmud Rafee, György Fazekas, and Geraint A. Wiggins. Performer Identification using Note Level Expressive Features. *The Journal of New Music Research, 2022.* **(In Review**)

- Chapter 5: Huan Zhang, Jingjing Tang, Syed Rifat Mahmud Rafee, Simon Dixon and György Fazekas. ATEPP: A Dataset Of Automatically Transcribed Expressive Piano Performance. *In 23rd International Society for Music Information Retrieval Conference (ISMIR 2022).*, 2022.

- Chapter 6: Syed Rifat Mahmud Rafee, György Fazekas, and Geraint A. Wiggins. HIPI: A Hierarchical Performer Identification model based on Symbolic Representation of Music. *2023 IEEE International Conference on Acoustics, Speech and Signal Processing.*

  Syed Rifat Mahmud Rafee, György Fazekas, and Geraint A. Wiggins. A Hierarchical Modelling Approach of Expressive Performance for Virtuoso Pianist Identification. *Transactions of the International Society for Music Information Retrieval (TISMIR), 2023.* **(In Review**)

Chapter 2

Background

2.1 Introduction

This chapter presents the technical background of the thesis, beginning with an introduction to musical expression and the factors that influence it. Then we provide a detailed review of the various techniques and algorithms that have been developed to analyze and replicate the expressive elements of a musical performance using computational modelling approaches in Section 2.3. We then shift our focus on the use of expressive features to quantitatively measure the playing style of a performer in Section 2.4. In Section 2.5 we provide an overview of the use of statistical models and music similarity measurement algorithms in the analysis of musical expression. This section discusses the various techniques and approaches that have been developed in this area, and how they can be used to identify patterns and characteristics in a performer's playing style. The definitions of machine learning and deep learning algorithms are covered in Section 2.6, after which the discussion moves to how these models may be used in music information retrieval. Finally, in Section 2.7, we look at various evaluation metrics and techniques that have been developed to assess the reliability and performance of computational models.

2.2 Expressive Music Performance

A fundamental concept in the study of music and its performance is that listeners perceive music as expressive [26, 27, 28]. For many listeners, being “moved” by a performance is the pinnacle musical experience. The enjoyment of the music greatly relies upon the manner in which a piece of music has been performed. In addition to the emotional content of a composition that is introduced by a composer by incorporating numerous musical ideas into the composition’s structure to present contrast and surprise the listeners, performers frequently improvise and express the affective emotional content through their performances. They do not just play a piece of music with the prescribed durations and the pitches printed in the notated score. Instead, they speed up or slow down in certain places, emphasising certain notes or passages in different ways to enhance the emotional expressivity. The most important parameters that are available to a pianist are timing, tempo, dynamics and articulation (the connection between the successive notes and how each note is played). These subtle variations are not stated in the notated score, yet they are vital for the music to be affective and engaging. The art of continually shaping these important parameters throughout a musical performance is also known as Expressive Performance. Performers learn these performative rules through many years of dedicated and rigorous training as well as from intellectual involvement with music. Artists that can evoke strong feelings in their listeners tend to be more popular than just technically adept ones [29]. However, it goes without saying that technical proficiency is vital, but the ability to really express oneself is what sets great artists apart.

2.2.1 Different Aspects of Expressivity

The term ‘expression’ can be utilised in different ways depending on the musical context in which it is being used. For example, it has been broadly used to define the deliberate changes in acoustic features (such as tempo, dynamics, articulation and timing) that differentiate one performance of the same musical composition from another [6]. On the other hand, the word expression has also been used to describe how listeners perceive musical performance to be emotionally expressive [27]. Expressiveness in music, however, is not limited to the simple expression of emotions; rather, it is a phenomenon with several dimensions [30] that also includes the musical sensitivity (knowing exactly how to perform a particular phrase or section) of a performer and their ability to convey the emotion and meaning of a piece of music through their performance [31].

Studies of expressive music performances have generally taken one of three approaches. The first is the philosophical point of view that has existed since ancient times and has been used to examine the question of how music may be described as “expressive” [26, 32, 27, 33]. The musicological perspective, which incorporates both music psychology and cognition, is a second way of looking at expressive performance [34, 35, 36, 37, 38, 39, 40]. Since the advent of more precise data collection enabled by computers and electronic devices in the late 1980s [41], a third approach has emerged, which considers the study of expressive performance as an empirically tractable issue that can be investigated through performance analysis [42, 28, 6] or computational models [43, 44]. This study does not have the time or space to provide a comprehensive analysis of all three approaches; rather, it focuses on the third method, computational models of expressive piano performance, as its primary emphasis.

2.2.2 Music Structure

The emotional content in music is often linked to two interdependent factors: the structure laid down in the score by the composer and the interpretation that the performers makes of it [45]. Musical structure is the arrangement of many musical ideas inside a composition, most of which are offered to present contrast and surprise the listener, while others are repeated or even altered to create a feeling of familiarity. These musical ideas are closely connected and may be decomposed into shorter ideas due to their hierarchical organisation. The interaction between familiarity, novelty and hierarchical decomposition of melodic patterns is coherently structured to reflect the composer’s view of the piece of music.

This perspective of music as a whole is sometimes referred to as musical form, and it is the most obvious manifestation of the hidden hierarchical structures embedded in music. In Western classical music, for instance, the lowest level of the hierarchy comprises of the notes and chords that make up a piece. Further, by linking them together in a synchronous or sequential fashion, we get higher structural constructs like measures, motives, and phrases, which in turn help define sections. As depicted in Figure 2.1, these high-level structural features dictate a composition’s overall layout, and the resulting nested organisation of sounds makes it feasible to visualise the hierarchical structure of a piece using tree representation of music [3].

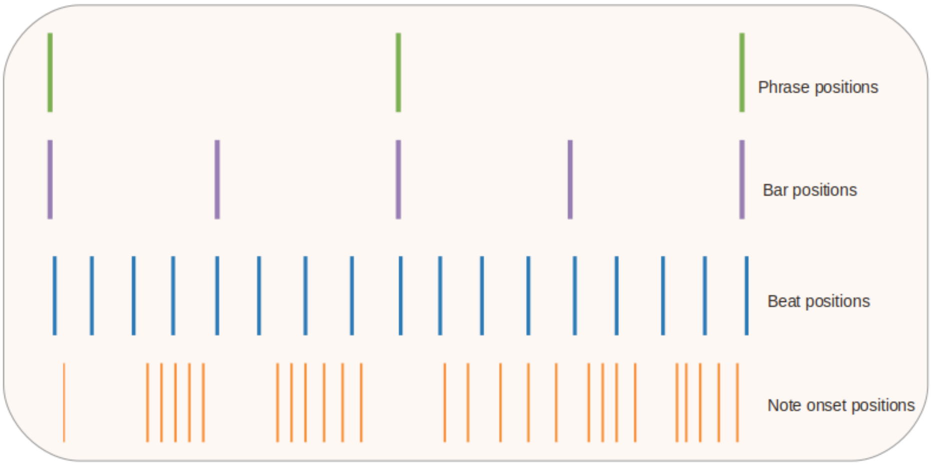


Figure 2.1: Visual representation of the metrical hierarchy of Western classical music.

2.2.3 Individual Styles of Performer

The inherent emotional content in the music flows from the composer to the performer to the listener [5]. Performers play a vital role in communicating the composer's intent to the audience by conceptualising the music in terms of a wide variety of abstract structures (motifs, phrases, sections etc.) and then continuously "shaping" these structures via the use of tempo, dynamics, and articulation. This often adds the emotional richness to the composition and Palmer [6] refers to this as performer's interpretation. To put it more simply, it is the process for performers to decide how they will perform the precise notated music to make it come colourfully and expressively alive. Performance that conveys emotion and feeling can be seen as a combination of the performer's interpretation and the physical execution of the piece. This is influenced greatly by the performer's personal style, as they infuse their own emotions, interpretation, and personality into the piece, resulting in a wide range of expressive styles within a specific genre. For instance, Some performers may be more dramatic and expressive, while others may be more reserved in their musical expression. Some may focus on technical precision and clarity, while others may prioritize emotional depth and intensity.

Virtuoso performers typically convey their own musical styles by adding a plethora of subtle variations to the most crucial expressive factors. However, these subtle variations are not stated in the notated score, but they convey and reflect the performer's comprehension of the structure and affective content inherent in a composition, thereby highlighting the dramatic, affective, and emotional qualities and potentially engaging and affecting the listeners emotionally. This explains why the "Ode to Joy" performed by the Berliner Philharmoniker is so much more lively and expressive than the one on your cellular phone.

A performer's expressive style to music may be shaped by their education, upbringing, life experiences, and the historical and cultural context in which they are performing. In addition to the ways in which individuals approach expressivity, it's possible that the ways in which various musical traditions or genres approach expressivity varies from one another. For example, classical music may place a greater emphasis on technical mastery and formal structure, while jazz or folk music may place a greater emphasis on improvisation and personal expression. Overall, the combination of these and other factors can contribute to the uniqueness of a performer's style, making them stand out from others in their genre. While it is possible to conceptualize a music performer's style theoretically, the use of computational models allows for a more systematic and precise examination of expressive music performance. In the following section, we will explore the application of these models to analyze and model musical expressivity.

2.3 Computational Modelling of Expressive Performance

Computational modelling of expressive performance has attracted increasing attention in recent years. The goal of computational modelling is to formulate hypotheses regarding expressive performance in the form of computer programmes that can be validated empirically using real measured performance data. Hence, it has been the subject of research and investigation in a wide range of scientific and creative fields [46], including music psychology and musicology, as well as computer science.

The motivations behind modelling expressive performance using computational modelling can be grouped into two wider categories. Firstly, Computational models may be used to examine how human perform music [47, 48]. In addition to this, it enables us to study the relationship that exists between the composer, the performer and the listener [5, 49]. Secondly, they can be used to generate new performances of musical pieces in a variety of contexts [50, 51, 52, 53]. Since it is beyond the scope of this chapter to provide a comprehensive analysis of both categories, we will instead concentrate on the former kind and explore computational models as a way of understanding how performers perform music. In particular, we try to quantify the stylistic features that characterises performers, which then can be modelled by computational models for the purpose of performer identification.

2.3.1 Modelling Relationship of Music Structure and Performance

Computational models, as analytical tools, allow us to study how humans perform music by investigating the relationship between musical features like phrase structure and performance features like timing and dynamics. A first notable trend in recent research is the growing importance of data-driven techniques that depend on machine learning algorithms to learn the relationship from a large corpus of real-world data. The research conducted by Kosta et al. [54, 55, 56], for instance, which focuses on learning the relationship between expressive dynamics and dynamic markings, is an example of such approaches. Grachten and Widmer [51] proposes a modelling framework that attempts to quantify the effect of annotated score descriptors (pitch, dynamic markings and metrical positions) on expressive performance parameters which includes expressive dynamics [57] and timing [58]. To represent the dynamics, articulation, and timbral aspects of expressive ensemble performances,[59] investigate the use of score features characterising horizontal (i.e. melodic) and vertical (i.e. harmonic) contexts. Giraldo and Ramirez [60] presented a data-driven computational approach including machine learning and feature selection to induce expressive performance rule models for note duration, onset, energy, and ornamentation transformations in jazz guitar music.

The quantitative studies on how different structural features contribute to the expressive performance not only demonstrate some well known relationship but also suggest that aspects of performance may be connected to the structure of the music in many ways; for instance, phrasing has been shown to be related to dynamics [61], and timing [62]. Leman et al. [63] developed a computational model to examine the timing and phrasing of piano performances to learn how these factors contribute to the performance's expressiveness. Todd has proposed several structure level models for modelling expressive timing [64, 65, 66] and dynamics [10]. Todd [64]'s findings support his claim that performers tend to increase tempo and dynamics towards the middle of the phrases and decreases towards the end so that listeners perceive the hierarchical structure of the music. Another study by [67, 68] tried to build a mathematical model of musical structure and expression based on enormous theoretical background, namely, the "mathematical music theory" [69]. Recent research by Gingras et al. [49] has provided more evidence of the correlation between musical structure and performance, which in turn serves as a reliable predictor of affective perception(listener).

While previous research has explored the connections between music structure and expressive performance parameters, there is an opportunity for a more comprehensive approach. Existing studies have identified correlations between specific elements like dynamics and timing and their impact on musical performance. However, these studies have not fully exploited on these findings, particularly in the context of performer identification. Additionally, the potential of deep learning techniques remains underutilized for gaining a deeper understanding of these complex relationships. This study aims to address these issues by proposing a hierarchical modelling approach in Chapter 6 that concurrently models both the structural and performance characteristics of music.

2.3.2 Comparing Expressive Performances

Comparing multiple performances of the same piece by different performers to its notated score has been a popular method for modelling expressive performance. This approach provides an opportunity to compare and analyse the similarities and differences in how different performers interpret a piece of music. However, obtaining a digital version of the notated score can be challenging. Furthermore, directly comparing the dynamics (loudness) of a digital score with those of an actual performance can be problematic, as scores often lack precise information regarding dynamics.

Hence, some of the works follow an unsupervised approach, where the average of multiple aligned performances of the same piece is compared with the individual performances[1, 70, 71]. This stands in contrast to supervised approach where each individual performance is compared with its respective score. Repp [13] examined the statistical analysis of temporal similarity and variety of a well-known composition that illustrates the individuality of a number of renowned pianists. Another study by Sloboda [72] shows that it is possible to deduce which notes were emphasised by measuring the differences in the performers' expressive variations, dynamic, and articulation.

A more recent study by Stamatatos [73] reveals that deviation from the norm performance (average of timing and dynamics) is stronger in portraying performer individuality than deviation from the printed score. Bernays and Traube [74] investigated pianist's individuality for the same piece performed by different performers through the study of different performance features measured from their gesture application on the keyboard. Their study suggests that individual performance technique correlates to the abstract concept of timbre and that pianists may express distinctive style through specific timbral intentions. Another approach to compare expressive timing is presented by Liem and Hanjalic [75] and Liem et al. [76], who look at alignment patterns across different expressive performances of the same work using standard deviations and entropy. Despite the fact that computational models for performance comparison have yielded some fascinating findings, very little progress has been made in really understanding the way humans perform music expressively.

2.4 Expressive Features for Quantifying Performer's playing Style

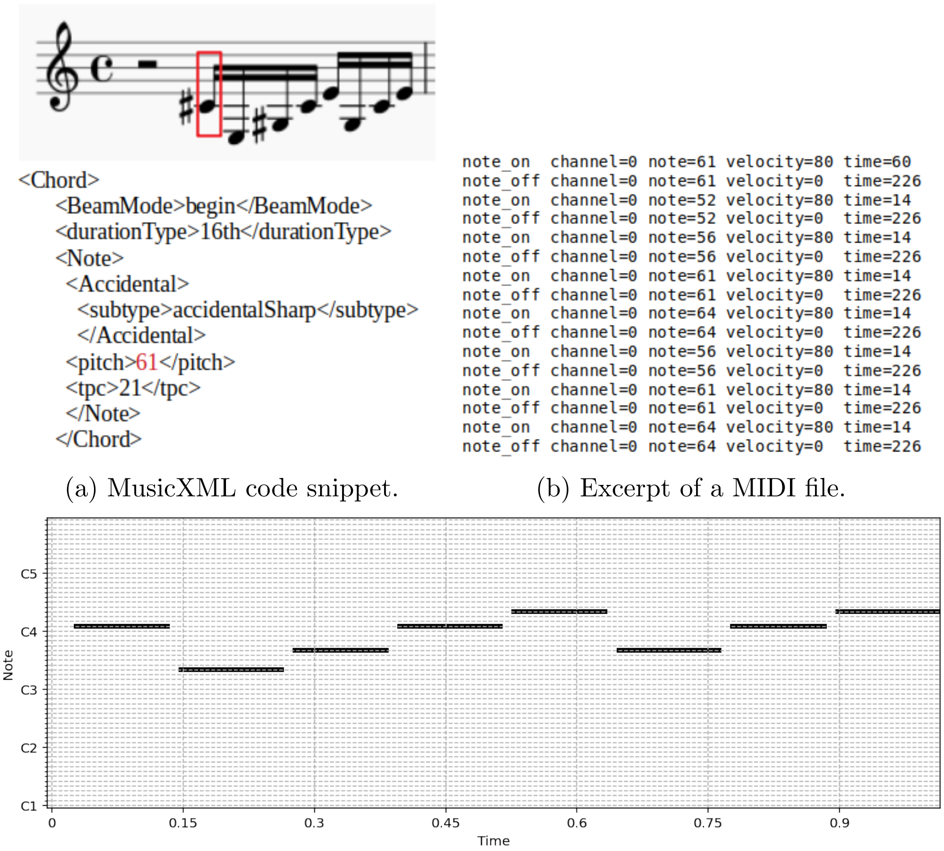
One of the most fascinating aspects of music performance is the way in which performers bring their own unique style and interpretation to a piece of music. This expressive quality, also known as "performer's playing style," can be influenced by a variety of factors, including the performer's training, technique, and artistic vision. To better comprehend and assess a performer's playing style, it is important to investigate the expressive elements that contribute to this quality. In this dissertation, we aim to develop such features that can be represented by a variety of numerical descriptors extracted from the recorded piano performances. These features are further used to discriminate and identify virtuoso piano performers. This section gives an overview of the expressive features that have been used in the past and are still used to analyse musical expressions and model different playing styles of performers.

Expressive features may be categorised according to the structural level of music from which they were derived. The low-level features for example tempo, timing and dynamic variations in a note level capture the small-scale expressive elements. The mid-level features demonstrate the medium-scale expressive elements that summarises the low level features, such as phrasing patterns, note duration patterns and articulation patterns. The high-level features, which are sometimes also referred to as semantic features, are what capture the large-scale expressive elements that reveal aspects that are typically close to how humans perceive music [77]. These aspects include the overall tempo and the average loudness of the track, for example. The most important expressive parameters available to piano performers are tempo/timing, dynamics and articulation; thus, we will only discuss these parameters, excluding additional aspects such as timbre, vibrato, and intonation. In addition, we begin with a brief introduction of the basics of symbolic representation.

2.4.1 Symbolic Representations

In general, there are two very common types of representations that may be used to represent music: audio and symbolic. Audio representations, such as WAV or MP3 files, encode sound as an electrical signal by digitizing and sometimes compressing the acoustic waves that are generated by a sound source. These waves are the result of the vibrational motion of a sound source, such as a string on a musical instrument or the vocal cords of a singer, which produce changes in air pressure that travel through the air and are detected by the human ear [78]. In contrast, symbolic music representations, such as MIDI, contain score information with an explicit encoding of musical events, such as notes, and instrumentation, rather than simply encoding the acoustic waves of the sound [78]. This allows for greater control and manipulation of the music, as well as the ability to more accurately represent complex musical structures. That being said, we shall review the fundamental characteristics of symbolic representations, with an emphasis on MIDI, the de facto standard for controlling music synthesisers.

While notated scores, such as sheet music, are also considered symbolic music, our focus here is on the digital representation of symbolic music.



```
<Chord>
   <BeamMode>begin</BeamMode>
   <durationType>16th</durationType>
   <Note>
    <Accidental>
      <subtype>accidentalSharp</subtype>
      </Accidental>
    <pitch>61</pitch>
    <tpc>21</tpc>
    </Note>
   </Chord>
```

(a) MusicXML code snippet.

```
note_on  channel=0 note=61 velocity=80 time=60
note_off channel=0 note=61 velocity=0  time=226
note_on  channel=0 note=52 velocity=80 time=14
note_off channel=0 note=52 velocity=0  time=226
note_on  channel=0 note=56 velocity=80 time=14
note_off channel=0 note=56 velocity=0  time=226
note_on  channel=0 note=61 velocity=80 time=14
note_off channel=0 note=61 velocity=0  time=226
note_on  channel=0 note=64 velocity=80 time=14
note_off channel=0 note=64 velocity=0  time=226
note_on  channel=0 note=56 velocity=80 time=14
note_off channel=0 note=56 velocity=0  time=226
note_on  channel=0 note=61 velocity=80 time=14
note_off channel=0 note=61 velocity=0  time=226
note_on  channel=0 note=64 velocity=80 time=14
note_off channel=0 note=64 velocity=0  time=226
```

(b) Excerpt of a MIDI file.

(c) Piano roll representation.

Figure 2.2: The different digital representations of symbolic music.

Hence, in Figure 2.2, we provide three very common digital representations of symbolic music. The MusicXML markup language shown in Figure 2.2a is a text-based structured representation tailored specifically to the needs of musical applications [79]. It offers a standard format for exchanging and storing musical data among music notation software, but it is too cumbersome to be used directly by humans. The same applies for their usage as representation for any machine learning task, despite the fact that they can be converted to more readable formats like MIDI. Figure 2.2c, shows a piano roll representation of the notated score presented in Figure 2.2a. It is a graphical representation of the music piece in which the horizontal axis represents time and the vertical axis represents pitch. Since each note on a piano roll is shown as a discrete event rather than as a part of a chord or other musical structure, it is much simpler to manipulate and rearrange musical passages in a digital setting. Despite its popularity as a standard representation for many machine learning algorithms, the absence of note-off information makes it difficult to tell the difference between a sustained note and a series of shorter ones [80].

Finally, Figure 2.2b displays the MIDI representation of the score shown in Figure 2.2a, after being processed using a python library called *mido* [81]. The Musical Instrument Digital Interface (MIDI) standard specifies a protocol for the digital representation and exchange of musical and other audio-related data [82]. In real time, they communicate note performance data and control data through a series of event messages. As shown in Figure 2.2b, the note_on event indicate that a note is played and the note_off event indicate the end of the note. Additionally, each MIDI event has a channel number between 0 and 15, which specifies the instrument or track, a MIDI note number, which specifies the note pitch and can take on integer values between 0 and 127, a velocity, which specifies the loudness of a MIDI note also represented by integer values between 0 and 127, and a time attribute, which specifies the delta-time value in ticks. Ticks and beats are the basis of MIDI file timing. You may think of a beat as a quarter note. The smallest time measurement in MIDI is called a tick, and it is used to divide beats. The ticks of each MIDI message indicates how much time has elapsed since the previous message was received. One of the primary benefits of MIDI is that it enables multiple devices to communicate and exchange music and other data. As a result, MIDI has become a key technology for the music production and distribution infrastructure, facilitating the use of a broad variety of software and hardware for music composition, arrangement, and editing.

2.4.2 Expressive Timing and Tempo

Expressive timing and tempo are vital components in music that add to the piece's overall emotional effect and character. They allow the musician to create a sense of story within the music and express a broad variety of feelings. The temporal position of musical events is typically conveyed via the use of expressive timing and tempo. The tempo of a piece of music refers to the speed at which it is played. It is indicated in beats per minute (bpm) or a particular note value (such as quarter note, half note, etc.) and can often refer to the global tempo of a performance or the local tempo. Global tempo refers to the overall speed at which a piece is performed and typically indicated in the music notation using a metronome marking. Unless changed by the composer or the performer, global tempo normally remains constant throughout the whole composition. Local tempo, on the other hand, refers to the speed at which a specific section or passage is performed and may be viewed as local deviations from the global tempo [48].

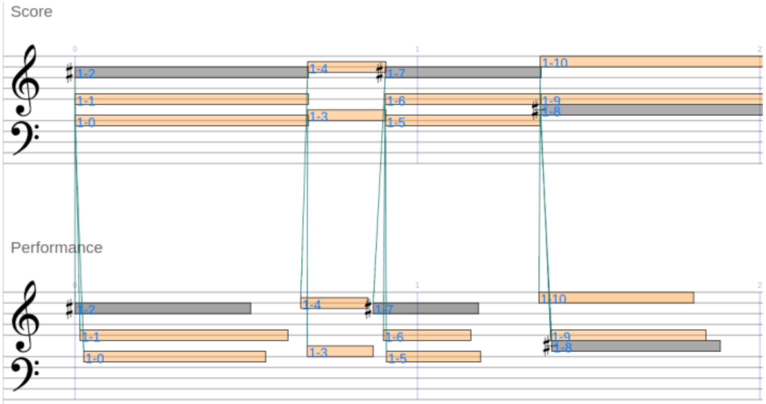


Figure 2.3: Note Onset locations for a performance and its corresponding score.

In music, timing refers to the exact location of musical events like the placement of notes and rhythms inside a measure, and the music is often perfectly quantised, meaning that note onsets are properly matched onto the tempo. Expressive timing, however, refers to the intentional deviations of those individual events from the local tempo [48]. Figure 2.3 provides an illustration, with the top staffs depicting the exact onset positions of each note in a score and the bottom staffs depicting the locations of the same notes during a performance. Timing plays a significant role in expressive performance, allowing the artist to shape the music and evoke the desired feeling in the listener [83]. For example, a performer may inject those micro-variations in timing in their performance to build up a sense of tension by rushing ahead of the beat or to create a sense of anticipation by holding back beat or two before resolving a musical phrase.

Expressive tempo and timing have been shown to have strong connections to musical expressiveness, music emotion cognition, and performers' playing styles [50, 84, 65, 85, 83, 86, 47, 58, 87]. It is therefore assumed that the expressive timing features can characterise performers' unique playing style and may be used to differentiate across pianists. As discussed in section 2.3.2, it is usual practise to compare a performance with its score to determine the precise time at which each note occurs in both the score and the performance [47, 88]. In addition, a quasi performance by taking the average of several performances of the same piece can be used as a reference point for comparison [70, 71, 1].

2.4.3 Expressive Dynamics

The term "expressive dynamics" is used to describe how a pianist changes the loudness and intensity of a composition during a performance. For example, they may accentuate or soften a certain musical idea to effectively convey the structure and emotion to the listener [89]. While the term "dynamics" may refer to a wide variety of aspects of music [90], our focus here is limited to those that relate to actual musical performance. Expressive dynamics, or the variations in loudness and intensity in a musical performance, are controlled differently on different instruments. On the violin, expressive dynamics are achieved through the use of bow velocity, bow pressure, and bow position, whereas, on the piano, expressive dynamics are controlled by the velocity of the hammer as it strikes the string [90].

Most studies of the expressive piano performance relies on the MIDI velocity as a substitute for the exact volume of the sound produced during a performance. Some previous research, such as the NAIST model [91], the early versions of the Basis function model [51, 92], and the unsupervised method developed by Van Herwaarden et al. [93], all made use of MIDI velocity as an expressive objective for each individual note in the score. Other research that focuses on polyphonic music use sequential models to disentangle the melodic lines of each voice in a track and predict their individual MIDI velocities of each voice [94, 95, 96]. These studies have shown promising result in analysing expressive performance using the midi velocity, despite the fact that, MIDI velocities do not represent the actual measured or perceived loudness and may change from instrument to instrument [48].

2.4.4 Articulation

The term "articulation" is used to describe the manner in which a musician plays or produces a certain note or succession of notes on an instrument.

When it comes to playing the piano, articulation refers to the ratio between the duration of a note as it is played and its written duration on the score, which also determines the amount of overlap between consecutive notes [48]. Articulation is a key component of musical performance that enables artists to transmit a broad variety of emotions and nuances to the audience, similar to the way we articulate words while speaking [97]. There are many traditional articulation techniques that have been standardized in Western music [97], but composers can also invent new ones as needed [98], and performers have the freedom to interpret and apply articulation in their own way based on their interpretation of the music and the context of their performance [97]. However, the most prevalent articulation techniques in piano performance include staccato (the note is brief or detached) and legato (a note to be performed smoothly and connected with its successor).

A common approach of analysing articulation is to compare real performances with their corresponding scores. For example, Bresin and Umberto Battel [99] analyzed how pianists used different articulation strategies in expressive performances of a specific score, collecting measurements of key overlap time and its relation to the inter-onset-interval for legato and staccato notes and examining the resulting articulation applied by the right hand through statistical analysis, with the aim of potentially creating articulation rules for automatic piano performance. In contrast, other studies use linear [100] or logarithmic [52, 96] scaling of parameters to describe articulation quantitatively. More recently, [101] has made an effort to describe articulation by first modelling the pedal information that has complex consequences for note duration with the goal of automatic performance rendering.

2.5 Statistical Distribution Models and Music Similarity

Distributions of expressive features can be a powerful tool for modelling and understanding a piano performance. More importantly, we may use well-established statistical models to capture and quantify a performer's unique style by modelling the distributions of different expressive variables over all of his performances, under the assumption that distribution gives the compact representation of a pianist's style [102, 71]. There are various statistical models that can be used to model distributions of expressive features in piano performance. In this thesis, three well-known distribution models, namely Histogram, Kernel Density Estimation (KDE), and the Gaussian Mixture Model, are used, and then their modelling capabilities on pianist's style are compared. The definitions and applications of these models that are used throughout this thesis are outlined in section 2.5. Next, music similarity is thoroughly reviewed in section 2.5.2, since our proposed method uses a combination of statistical models and music similarity measures for pianist identification.

2.5.1 Statistical Distribution Models

Statistical modelling is the process of developing mathematical models that characterise the interrelationships between different variables in order to evaluate and interpret data [103]. This facilitates data analysts to easily detect patterns and trends in data, make future predictions, and create visually appealing representations of that data in an intuitive way. In the context of music performance, statistical models may be used to examine the feature distribution of hundreds or thousands of notes to provide insight into the performer's playing characteristics and style. This may be especially beneficial in music information retrieval studies, where it can be difficult to determine the link between a performer's style and expressive elements at the note level. Three specific statistical models that are commonly used in this context are histograms, kernel density estimation, and Gaussian mixture models. This section discusses the definition and calculation techniques of these models, in addition to their application to the MIR research.

2.5.1.1 Histogram

A histogram provides the approximate estimation of how the numerical values in a dataset are distributed. It is a way to visualise the frequency or occurrence of different values within certain ranges. Pearson [104] initially proposed the concept in 1894, and since then it has been used as one of the common approaches of density estimation, or more specifically, of estimating the Probability Density Function (PDF) that provides an approximate estimation of the density of the underlying distribution of the data. The estimated PDF may then be used to compute the likelihood that the random variable will take on a certain value or fall within a specified range of values

[105].

In order to construct a histogram, one must first divide the possible values into a series of intervals (or "bins"), with the width of each bin being referred to as the "bin size," and then determine how many values fall into each interval. Typically, the bins will be defined as a series of discrete, non-overlapping intervals of a variable and the intervals (or bins) must be adjacent and are typically (but not always) the same size [106]. In the case of equal size, the height of the bar over each bin is proportional to the frequency of values in that bin. Alternatively, the bins can be of unequal size, in which case the area of the rectangle over each bin is proportional to the frequency of values in that bin [107]. There are no gaps between adjacent bins in a histogram, this indicates that the original value is continuous [108].

In MIR, histograms are often used to illustrate the distribution of different musical properties or aspects. Using a histogram, one may see the distribution of pitch or tempo in a music recording, for instance. This can be useful for tasks such as genre classification [109], where the distribution of certain features may be indicative of the style of music. Histograms can also be used to identify patterns in the distribution of musical events over time [110], and to visualize the distribution of timbral features such as spectral content [111]. Overall, histograms are a useful tool for understanding and summarizing the characteristics of music performance and for extracting meaningful information from them.

2.5.1.2 Kernel Density Estimation

Kernel density estimation is a statistical method for estimating the probability density function (PDF) of a random variable based on a data sample [112]. It is a non-parametric approach, meaning that no assumptions regarding the shape of the underlying distribution are required. Instead, it operates by assigning a "kernel" function to each data point and then summing these functions to estimate the PDF of the variable. The kernel function is a probability density function itself, which is typically a bell curve (normal distribution) [113]. The resulting estimate is smooth and continuous, unlike a histogram, which is discrete and represented by vertical bars. Here is the formula for calculating kernel density estimation:

Let's say we have a set of n data points $\{X1, X2, ..., X_n\}$. The kernel density estimation for a point x is calculated as follows:

$$\hat{f}(x) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{X_i - x}{h}) \tag{2.1}$$

where h is the bandwidth that controls the smoothness of the estimate. A smaller h results in a smoother estimate, while a larger h results in a more jagged estimate. K is the kernel function evaluated at x for the $i^t h$ data point, and the sum is taken over all n data points to obtain the final PDF curve. The kernel function (indicates a Gaussian kernel) itself is defined as follows:

$$k_i(x) = \frac{1}{\sqrt{2\pi}} * \exp^{\frac{-x^2}{2}} \tag{2.2}$$

Choosing the optimal bandwidth can be a challenge, and there are several methods for doing so such as including Mean Integrated Squared Error [114]and rule of thumb [112].

Previous studies have made use of KDE for the purpose of music classification [115, 116], emotion recognition [117, 118], or music similarity analysis [119] due to its robustness in modelling the distribution of data and its visualisation. In light of this, we assume that KDE may be used to describe the distributions of a performer's expressive characteristics, therefore providing a compact representation of their unique style.

2.5.1.3 Gaussian Mixture Model

The Gaussian mixture model is a parametric method for estimating the probability density that is expressed as a weighted sum of a number of different distributions that follow the Gaussian distribution (often known as the normal distribution) [120]. Each Gaussian distribution in the mixture may be thought of as a separate component of the model, with its weight representing the proportion of the likelihood that a given data point was produced by that component. The GMM can be used for clustering, density estimation, and classification. In clustering, the model is used to find groups of similar data points within the data. In density estimation, the model is used to estimate the probability density function of the data. In classification, the model is used to assign class labels to data points based on the component of the mixture model that they are most likely to belong to.

The probability density function of the uni-variate or one-dimensional Gaussian distribution can be formulated as below:

$$\mathcal{N}(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{2.3}$$

Where the μ refers to the distribution mean and σ^2 refers to the variance. However, for multivariate GMM with k components, the parameters can be described as below:

The mean vectors of the component Gaussians, $\{\mu_1, \mu_2, \ldots, \mu_k\}$. The covariance matrices of the component Gaussians, $\{\Sigma_1, \Sigma_2, \ldots, \Sigma_k\}$. The mixing weights of the component Gaussians, $\{\omega_1, \omega_2, \ldots, \omega_k\}$. These weights sum to 1 and represent the probability that a data point was generated by each component. Given a set of n data points $\{x_1, x_2, \ldots, x_n\}$, the likelihood of the data given the GMM is given by:

$$p(X|\omega, \mu, \Sigma) = \prod_{i=1}^{n} \sum_{j=1}^{k} \omega_j \mathcal{N}(x_i|\mu_j, \Sigma_j) \tag{2.4}$$

with,

$$\mathcal{N}(x|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right) \tag{2.5}$$

where $\mathcal{N}(\cdot|\mu, \Sigma)$ is the Gaussian probability density function with mean μ and covariance matrix Σ.

To fit the GMM to a set of data, the model parameters can be estimated using the expectation-maximization (EM) algorithm. The EM algorithm iteratively refines estimates of the model parameters by alternating between the expectation step, in which the responsibility of each component for generating each data point is calculated, and the maximization step, in which the model parameters are updated based on the responsibilities.

GMMs have been widely used in MIR and since they can represent complex distributions over features extracted from music data, this makes them well-suited for tasks such as genre classification [121, 116], musical emotion modelling [122], and instrument classification [123, 124], where the goal is to identify patterns in the data that are indicative of certain categories or structures. Hence, we assume that GMMs can model the distribution of various expressive features that are indicative of the pianist's style.

2.5.2 Music Similarity

Music similarity refers to how closely two pieces of music are related in terms of their musical characteristics. A similarity score is used to numerically express the degree of resemblance; a higher number indicates a greater degree of similarity. Many different methods have been developed to determine how similar two musical works are. These include feature-based approaches, which extract and compare musical features like pitch [125] and rhythm [126]; content-based approaches [127, 128], which analyse the audio content of the music to identify similar characteristics; and structural approaches [129], which examine the structure of the music. The use of music similarity in this thesis for the task of performer identification is motivated by its widespread application to various Music Information Retrieval (MIR) tasks such as music genre classification [130], instrument classification [131], emotion detection [132], playlist generation [133] and music recommendation [134].

To this extent, our similarity estimation method can be described in three main steps. At first, given the presumption that performers continually modify the expressive parameters at a micro level, it is expected that the expressive characteristics are collected from the variations of each note. The next step is to conduct a statistical analysis of the expressive features at the note level. This is necessary since expressive aspects at the note level are not reflective of structural or global musical characteristics. Each feature's distribution may be modelled using different statistical models, and mean and variance can be computed if they follow a known distribution. However, more complicated or unknown distributions need the use of more sophisticated statistical models with parameters that are learnt or trained from the data in order to represent the properties of the distribution effectively. As a final step, a measure of how musically similar two pieces are is created by calculating the distance or divergence between their feature distributions.

Kullback-Leibler divergence [135] (also known as KL divergence or relative entropy), a measure of the difference between two probability distributions, is a popular method for calculating the divergence of two distributions. Unlike JS divergence [136], it is a non-symmetric measure, meaning that the KL divergence between distribution A and distribution B is not necessarily the same as the KL divergence between distribution B and distribution A.

The KL divergence between two distributions P and Q is defined as:

$$D_{KL}(P||Q) = \sum_{x} p(x) \log \frac{p(x)}{q(x)} \tag{2.6}$$

where P refers to the distribution of the observed data and p(x) is the probability of x under distribution P. Q refers to a model or approximation of P and q(x) is the probability of x under distribution Q. It's important to note that P and Q distributions are defined on the same probability space.

However, it might be challenging to determine the KL divergence analytically, when the distributions contain complicated, multivariate probability density functions (PDFs), such as in Gaussian mixture models. In such cases, the KL divergence must be computed using approximations or alternate methods, such as matching-based approximations [137]. Hence, this can be formulated as below for two Gaussians $N(\mu_1, \Sigma_1)$ and $N(\mu_2, \Sigma_2)$:

$$D_{KL} = \frac{1}{2}\left(\log \frac{|\Sigma_2|}{|\Sigma_1|} + \mathrm{Tr}\left(\Sigma_2^{-1}\Sigma_1\right) + (\mu_1 - \mu_2)^T \Sigma_2^{-1}(\mu_1 - \mu_2)\right) \tag{2.7}$$

Where μ refers to the mean, Σ is the covariance matrix and Tr refers to the trace of matrix.

2.6 Machine Learning

Machine learning is a subfield of artificial intelligence (AI) that involves the design and development of algorithms that can learn from and make predictions or decisions based on data. It's the method of training a computer to solve problems on its own by analysing large amounts of data and drawing conclusions from those analyses. It is commonly used for classification and regression tasks, in which the objective is to predict a discrete label or a continuous value based on input data, respectively.

There are several categories of machine learning, including Supervised learning, unsupervised learning and reinforcement learning. Supervised learning involves training a model on labelled data, where the correct output label is supplied for each example in the training set, also known as ground truth. For example, let's denote a dataset, X = $\{x_1, x_2..., x_k\}$ and $x_k \in \mathbb{R}^d$, where k is the number of training samples and x_i is d-dimensional feature vector and each corresponding output labels are denoted as Y = $\{y_1, y_2..., y_k\}$. During the training phase, data samples and their labels are fed into the machine learning model, and the model's output label, $\hat{Y}$ is then compared against the real label, Y using a loss function that measures how far off the model's prediction was. The trained model is then used to make a prediction on a new sample that was not available during training. The process of unsupervised learning includes training a model on unlabeled data, in which the correct output is not supplied throughout the training process. It is necessary for the model to uncover the underlying structure of the data by using methods such as clustering. Finally, the fundamental principle of reinforcement learning is that an agent may learn from its interactions with its surroundings by receiving rewards and penalties as feedback.

In this Thesis, a supervised approach is taken for pianist identification. Alongside the distribution-based similarity estimation for pianist identification, two machine learning models, K-Nearest Neighbour (KNN) and Support Vector Machine have been used as a baseline model in Chapter 3. Hence, the definition and the calculation method for these two models are first described in this Section. We next introduce two well-known deep learning models, the convolutional neural network and the recurrent neural network, with the CNN and its many variants being used in Chapter 4 and the variant of RNN network being used significantly in Chapter 6.

2.6.1 Supervised Learning Models

2.6.1.1 K-nearest neighbors

K-nearest neighbors (KNN) is a supervised non-parametric machine learning algorithm that was initially developed by Fix and Hodges [138] and then further extended by Cover and Hart [139], for the purposes of both classification and regression. However, since pianist identification can be considered a multiclass classification problem, only the theory behind classification using KNN is discussed here. The basic principle behind KNN is to first determine the K number of training instances that are closest (in terms of some distance measure) to a new data point, and then to utilise the labels of those training examples to generate a prediction about the new data point.

The mathematical formulation of KNN is fairly straightforward. Given a set of training examples, $X = \{x_1, x_2, ..., x_k\}$, where each x_i is a d-dimensional feature vector and a corresponding label y_i, the goal is to predict the label of a new data point x using the KNN algorithm. To do this, we first need to define a distance metric. One common distance metric used with KNN is Euclidean distance [140], which is defined as:

$$\text{Euclidean Distance}(x, x_i) = \sqrt{\sum_{j=1}^{d}(x_j - x_{ij})^2} \tag{2.8}$$

where x_j and x_{ij} are the j-th features of x and x_i respectively.

Once we have defined a distance metric, we can find the K nearest neighbors of x by sorting the training examples by their distance to x and selecting the K examples with the smallest distance. The prediction for x is then made by majority vote: if the majority of the K nearest neighbors belong to a specific class, we predict that x belongs to the same class.

2.6.1.2 Support Vector Machine

Originally developed by Vapnik et al. and colleagues [141, 142, 143] at AT&T Bell Laboratories, Support Vector Machines (SVMs) are a supervised learning technique that may be used for both classification and regression problems. The primary objective of a SVM is to find a hyperplane in a high-dimensional space that most effectively divides the data into different classes.

When the SVM is provided with a collection of training data points belonging to two classes, it searches for the hyperplane that has the largest minimum distance between those points. This distance is often referred to as the margin. Out of a set of possible hyperplanes that the SVM algorithm constructs, the aim is to choose a hyperplane that has the maximum margin between it and the nearest points of either class in the training set, increasing the likelihood that unlabelled data will be correctly classified. Therefore, another term for the SVM is the Maximum Margin Classifier and the points closest to the hyperplane are known as support vectors. Using a linear kernel function, we fit a linear hyperplane across classes if the SVM data is linearly separable. It's called hard-margin [143]. Here is the mathematical formulation for a linear SVM:

Given a training set $\{(x_1, y_1), (x_2, y_2), ..., (x_k, y_k)\}$, where $x_i \in R^d$ and

$y_i \in \{-1, 1\}$, the SVM algorithm aims to find the hyperplane $w \in R^d$ and $b \in R$ that maximizes the margin between the positive and negative samples. This can be formulated as the following optimization problem:

$$\begin{aligned} &\min_{w,b} \frac{1}{2} \quad |w|^2 \\ &\text{subject to} \quad y_i(w \cdot x_i + b) \geq 1, \quad i = 1, 2, ..., k \end{aligned} \tag{2.9}$$

Here, $||w||$ is the Euclidean norm of w and x, and b is the bias term. The decision boundary of the SVM is given by the equation:

$$w \cdot x + b = 0 \tag{2.10}$$

Points on one side of the decision boundary are classified as one class, and points on the other side are classified as the other class. In the event that the data is not linearly separable, the soft margin approach [143] and the kernel trick method [142] are presented as potential solutions for fitting a non-linear boundary between classes; however, the specific mathematical formulation of these methods is beyond the scope of this thesis but can be checked in [144]. SVMs have been used extensively in various domains, including music information retrieval. In MIR, SVMs have been used for tasks like instrument classification [145], music genre classification [146], emotion recognition [147] and artist identification[19]. Thus, the SVM is utilised as a baseline to assess the effectiveness of our suggested strategy, which will be discussed in detail in Chapter 3.

2.6.2 Deep learning models

The ability of deep learning to discover complicated patterns and trends in the data has made it more relevant in music analysis. Music often exhibits complex variations and structures that can be challenging to identify. Furthermore, music is profoundly influenced by its performance context, and these contextual variations can profoundly impact the inherent patterns and trends. Deep learning algorithms show impressive generalisation abilities, meaning they can successfully apply their knowledge to new data that may be somewhat different from the data they were trained on. Hence, they can be particularly useful for MIR tasks like pianist identification, where the input data may vary significantly from one performance to another.

Deep neural networks (DNN) have been increasingly popular in MIR research, including music classification [148], transcription [149], and generation [150] as a means of automatically discovering the necessary representations to describe and identify musical expression. In light of this, we made an effort to leverage some of the deep learning architectures, such as convolutional neural networks and Recurrent Neural networks, both of which will be covered in this section.

2.6.2.1 Convolutional Neural Network

In the realm of deep neural networks, Convolutional Neural Networks (CNN), also known as convolutional networks, are a type of artificial neural network that was first proposed by LeCun et al. [151]. It was developed specifically for processing data organised in a grid-like fashion, such as images [152]. CNNs are more complex than Feed Forward Neural networks (FFNs) because the hidden units of CNNs use convolution rather than the more traditional matrix multiplication.

A typical CNN has a multi-layered architecture, with each layer performing some operation on the input data before passing it on to the next layer. The CNN typically consists of four types of layers: an input layer, convolutional layers, pooling layers and fully connected layers. Input layer is the initial CNN layer and receives input data. Input data is usually an image or time series passed through the rest of the CNN for processing. The function of convolutional layers is to discover spatial connections in the input data. It is done by applying a set of filters, also known as convolutional kernels, to the input data in order to identify patterns or features in the data. The filters are typically small and are moved across (in a sliding window fashion) the input data in a process called "convolution," and the output of the convolution operation is a set of feature maps, which allows the CNN to learn patterns at different scales and locations in the data.

Typically, a pooling layer is used after a convolutional layer to downsample the input data and lower the dimensionality of the feature maps generated by the convolutional layer. There are several types of pooling layers, including max pooling [153] and average pooling [154], which take the maximum or average value within a window of data points as the output of the pooling operation, respectively. In order to bring non-linearity into a model, an activation function is often introduced after the convolution and pooling layers. This nonlinearity enables CNN to discover more complex data patterns. Finally, fully connected layers are used to make predictions based on the information learned by the convolutional layers. This is done by taking the dot product between the outputs of the convolution and pooling layers with a set of weights that determine how much emphasis should be placed on each feature in the final prediction.

Although CNNs were designed for learning internal representations from two-dimensional data, the same technique can be leveraged using a one-dimensional CNN for one-dimensional sequences of data, such as in the instance of piano performance data for pianist identification. In recent years, CNN has become the predominant paradigm for music classification and music tagging tasks [155]. It is used in Chapter 4 to train a multiclass classifier that can identify pianists based on their performances.

2.6.2.2 Recurrent Neural Network

A recurrent neural network, often known as an RNN [156], is a subcategory of artificial neural networks developed specifically for processing input in the form of sequences or time series. In contrast to feed forward neural networks where data points are independent of each other, RNNs models the temporal dependencies of a time series or sequential data where each data point depends upon the previous data point. The idea of memory in RNNs enables them to store the states or information from previous inputs and use it to generate the next output in a series.

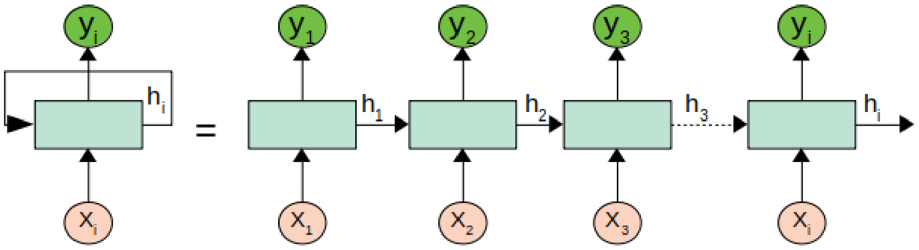


Figure 2.4: Example of an unrolled recurrent neural network.

As shown in 2.4, the internal structure of an RNN consists of a loop, which is used to propagate information about the previous time step to the next one. At each time step, i, the RNN takes in an input, $x_i \in \mathbb{R}$, and the previous hidden state, $h_{i-1} \in \mathbb{R}$, and produces an output, $y_i \in \mathbb{R}$, and a new hidden state, $h_i \in \mathbb{R}$. The hidden state is a record of all the inputs and outputs that have come before it, and it records the relationships and dependencies that exist between the various parts of the sequence. Mathematically, an RNN can be represented as follows:

$$\begin{aligned} h_i &= f(W_{xh} \cdot x_i + W_{hh} \cdot h_{i-1} + b) \\ y_i &= g(W_{hy} \cdot h_i + b') \end{aligned} \tag{2.11}$$

Here, x_i is the input at time step i, h_i is the hidden state at time step i, y_t is the output at time step i, W_{xh} and W_{hh} are the weight matrices for the input and hidden state, respectively, and W_{hy} is the weight matrix for the output. b and b' are the bias terms for the hidden state and output, respectively, and f and g are the activation functions for the hidden state and output, respectively. Sigmoid [157], Relu [158], and Tanh [159] are the most frequently used activation functions in RNNs. The symbol $*$ represents matrix multiplication.

An RNN is trained by randomly setting its initial weights and biases, forward passing the input through the network to produce an output, calculating the error between the predicted and target outputs with an objective function, backpropagating the error through the network to determine its gradient with respect to the weights and biases, and finally, updating the weights and biases with the gradient and a learning rate. The same procedure is repeated for each input-output pair in the training set, and as the training progresses, the RNN should learn to accurately predict the output given the input. However, one of the primary disadvantages of RNNs is that they may suffer from a vanishing gradient problem [160], in which the gradients used to calculate the weight update may approach zero and prevent the network from learning new weights. To address this limitation, Hochreiter and Schmidhuber [161] proposed a variant of RNN which is known as Long short-term memory (LSTM) network which we will briefly cover next.

2.6.2.3 Long short-term memory Network

Long short-term memory, also known as LSTM, is a variant of RNN that was developed to handle the vanishing gradient problem. It is capable of learning long term dependencies by remembering information for long period of time and particularly useful for tasks that require understanding of context over a long period of time. An LSTM functions similarly to an RNN cell. It consists of three different parts that interact with one another and are known as gates: an input gate, an output gate, and a forget gate. They are responsible for controlling the flow of information into and out of the cell. These gates are implemented using sigmoid activation functions, σ, which produce a number between 0 and 1 that represents the likelihood of an event occurring.

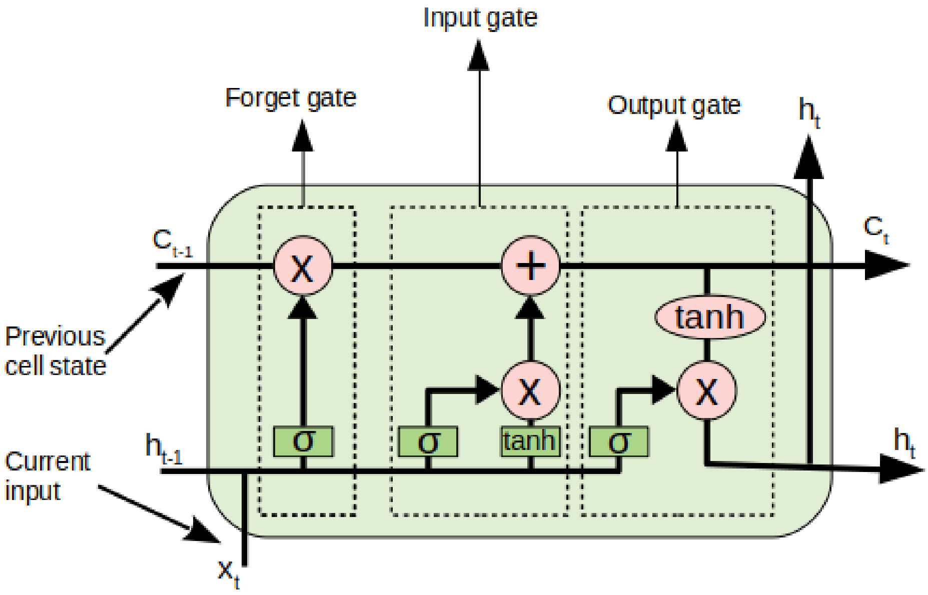


Figure 2.5: Internal architecture of a Long short-term memory network.

As depicted in Figure 2.5, an LSTM, like a regular RNN, has a hidden state, denoted by h_{t-1} for the hidden state at time $t-1$ and h_t for the hidden state at time t. Additionally, LSTMs contain a cell state denoted by C_{t-1} and C_t, where $t-1$ is the prior time stamp and t is the current time stamp. The input gate determines which values from the input sequence should be passed on to the cell state. The forget gate determines which values from the previous cell state should be discarded. And the output gate determines which values from the cell state should be passed on to the output. Mathematically, the LSTM cell for time step t can be represented as follows:

$$i = \sigma(W_{ix} \cdot x_t + W_{ih} \cdot h_{t-1} + b_i) \tag{2.12}$$

$$f = \sigma(W_{fx} \cdot x_t + W_{fh} \cdot h_{t-1} + b_f) \tag{2.13}$$

$$o = \sigma(W_{ox} \cdot x_t + W_{oh} \cdot h_{t-1} + b_o) \tag{2.14}$$

$$c_t = f \cdot c_{t-1} + i \cdot \tanh(W_{cx} \cdot x_t + W_{ch} \cdot h_{t-1} + b_c) \tag{2.15}$$

$$h_t = o \cdot \tanh(c_t) \tag{2.16}$$

Here, x_t is the input at time step t, h_t is the output at time step t, c_t is the cell state at time step t, and i, f, and o are the input, forget, and output gate activations, respectively. W_{ix}, W_{fx}, W_{ox}, W_{cx} are the weight matrices for the input gate, forget gate, output gate, and cell state, respectively. W_{ih}, W_{fh}, W_{oh}, W_{ch} are the weight matrices for the hidden state. b_i, b_f, b_o, and b_c are the bias terms for the input gate, forget gate, output gate, and cell state, respectively.

The training process of LSTM is same as RNN which is discussed in Section 2.6.2.2. Due to its capability of modelling long-term dependencies in data, LSTMs have been widely used in the field of music information retrieval. As a temporal series of events, music is well suited for LSTMs, which are capable of capturing patterns and relationships through time. This makes LSTMs useful for tasks that need a knowledge of the musical context. For example, it has been widely used for tasks like music generation [162], emotion recognition [163], beat tracking and tempo estimation [164], genre classification [165] and composer classification [166]. In Chapter 6, it is used to construct a hierarchical encoder model that is trained on a large-scale dataset of piano performances and then used as a pianist identification model.

2.7 Evaluation Methods

To quantitatively assess the performance of our pianist identification methods, we use classification evaluation metrics (F-score and Confusion Matrix) which have been used universally to evaluate multi-class classification algorithms. They provide for a standard by which model performance can be compared and the reliability of individual models can be evaluated. In addition, we use cross-validation in this thesis in order to mitigate the effects of bias brought on by the straightforward partitioning of the dataset. The concept of cross-validation and the evaluation metrics that have been stated are defined in this section.

2.7.1 F-score

The F-score or F-measure is an objective evaluation metric of a model's performance that was first established to evaluate binary classification models where the model can only categorise "positive" or "negative" instances, such as whether or not an email is spam. For both the ground-truth classes, y_i and the predicted classes, $\hat{y_i}$, positive instances are represented as "1" and the negative instances are represented as "0". Hence, the result of the model is evaluated only for the positive label by default, using two separate metrics, precision, P_1 and recall, R_1 and the overall model performance is determined by the harmonic mean of precision and recall, which is the F-measure. They can be defined as:

$$P_1 = \frac{N_{TP}}{N_{TP} + N_{FP}} \tag{2.17}$$

$$R_1 = \frac{N_{TP}}{N_{TP} + N_{FN}} \tag{2.18}$$

$$F_1 = 2 \times \frac{P_1 \times R_1}{P_1 + R_1} \tag{2.19}$$

where precision, P_1 is the proportion of true positives, N_{TP} (where, $y_i = \hat{y_i} = 1$) among all positive predictions made by the model. It is calculated as the number of true positives, N_{TP} divided by the sum of the true positives, N_{TP} and false positives, N_{FP} (where $y_i = 0$ but $\hat{y_i} = 1$). R_1, also known as recall or sensitivity, is the proportion of true positive, N_{TP} predictions among all actual positive cases. It is calculated as the number of true positives, N_{TP} divided by the sum of the true positives, N_{TP} and false negatives, N_{FN} (where $y_i = 1$ but $\hat{y} = 0$). F_1 is the harmonic mean of P_1 and R_1, and is calculated as 2 times the product of P_1 and R_1 divided by the sum of P_1 and R_1. It measures a model's accuracy that balances precision and recall, giving equal weight to both.

However, the standard F1 score is unsuitable for multi-class classification since it only takes into account the classifier's overall performance, rather than its performance for each individual class. This may be problematic since it might result in a classifier with a high overall F1 score but poor performance for certain classes. To address this issue, there are several variations of the F1 score that can be used for multi-class classification [167]. One of these variations is called the macro-averaged F1 score, and it is derived by first calculating the F1 score for each class and then averaging the results. This can be defined as:

$$P_{macro} = \frac{1}{N} \sum_{n=1}^{N} P_n \tag{2.20}$$

$$R_{macro} = \frac{1}{N} \sum_{n=1}^{N} R_n \tag{2.21}$$

$$F_{macro} = 2 * \frac{P_{macro} * R_{macro}}{(P_{macro} + R_{macro})} \tag{2.22}$$

where, P_n and R_n are the precision and recall for class n, respectively and N is the number of classes. Another variation is the micro-averaged F1 score, which calculates the overall precision and recall for the classifier across all classes and then calculates the F1 score using those values. This can be defined as:

$$P_{micro} = \frac{\sum_{n=1}^{N} TP_n}{\sum_{n=1}^{N} (TP_n + FP_n)} \tag{2.23}$$

$$R_{micro} = \frac{\sum_{n=1}^{N} TP_n}{\sum_{n=1}^{N} (TP_n + FN_n)} \tag{2.24}$$

$$F_{micro} = 2 * P_{micro} * R_{micro} / (P_{micro} + R_{micro}) \tag{2.25}$$

Where TP_n is the number of true positive predictions for class n, FP_n is the number of false positive predictions for class n, FN_n is the number of false negative predictions for class n and N is the number of classes.

The macro-average method is more suitable when the goal is to evaluate the performance of the classifier across all classes, and the data for each class is of equal size. The micro-average method is more suitable when the data for each class is not of equal size, or when the goal is to evaluate the overall performance of the classifier across all data. In the context of this thesis, the goal is to evaluate the performance of the classifier across all classes, and the data for each class is of equal size, so the macro-average F1 method is preferred over micro-average F1.

2.7.2 Confusion Matrix

While the macro-averaged F1 score may be used to assess the overall performance of a multi-class classifier, it does not provide detailed information about the performance of the classifier for each individual class. In cases when the performance of the classifier for each class is of particular interest, such as pianist classification, this might be problematic since it is uncertain whether or not all classifications are being predicted equally well. The confusion matrix may be used to solve this issue since it provides a comprehensive evaluation of the classifier's accuracy for each class. This has been widely used in many MIR tasks [168, 25].

It is a matrix where each row represents a predicted class and each column represents the actual class, and each entry represents the number of samples that were predicted to belong to that class. For example, if we have N classes in our dataset, the confusion matrix would be a $N*N$-dimensional matrix. Elements on the diagonal show the percentage of samples for which the predicted labels exactly match the actual labels. The samples that the classifier incorrectly labelled make up the non-diagonal elements. A heat map may also be used to display the confusion matrix, which can give further insights into how well the classifier is performing.

In this thesis, the goal is to evaluate the performance of a classifier for recognizing each piano performer in a dataset, and the confusion matrix is being used to provide a detailed breakdown of the classifier's performance for identifying each performer.

2.7.3 Cross Validation

When evaluating a machine learning model's performance, it is standard practice to divide the dataset into a training set and a test set. After the model has been trained using the training set, its effectiveness is measured using the test set. However, a straightforward split can introduce bias or exhibit selection bias problems [169], causing the model to overfit, as it becomes highly sensitive to the training set's characteristics.

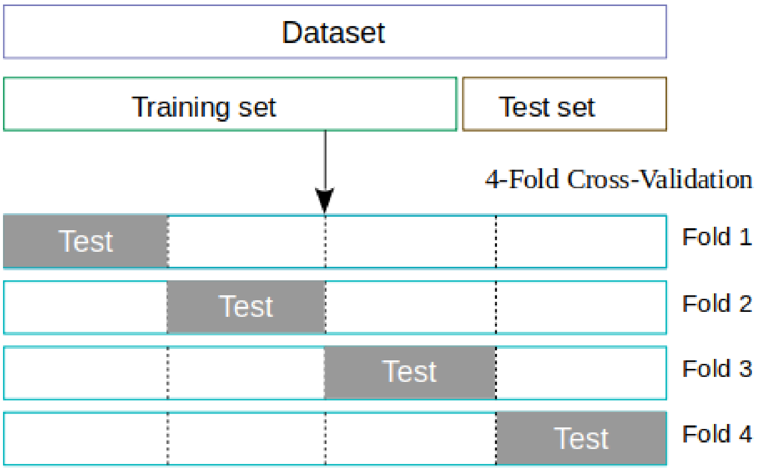


Figure 2.6: Graphical representation of the distinctions between the straightforward train/test split and the four-fold cross-validation technique

To mitigate this issue, cross-validation can be used to evaluate the model's generalization ability. It includes splitting the dataset into K folds of equal size, training the model on K-1 folds, and then assessing it on the final remaining fold. This procedure is carried out K times, with a new fold serving as the evaluation set each time. The model's peformance is measured by averaging its results over all K folds. Figure 2.6, shows the difference between a straightforward train/test split and a four-fold cross validation technique. The procedure for performing a 4-fold cross-validation can be described as below:

- Split the dataset into 4 folds of approximately equal size.
- For each unique fold, consider it as the test set and the remaining 3-folds as the training set.
- Train the model on the training set and evaluate it on the test set using a set of evaluation metrics.
- Repeat this process four times for each fold and average the performance across all folds to obtain the final performance measure.

To eliminate the possibility of bias in the folds, it is crucial that the data be shuffled before being partitioned. It is common practise to set the value of K between 5 and 10, however this is not a rule of thumb.

Estimating the model's performance using a higher K value takes more time, but yields more reliable results. Many variations of cross-validation exist, such as the Leave one/P out cross-validation and the Leave one/P group(s) out cross-validation [170]. In this thesis, the performance of the pianist identification system has been validated using Leave one group out cross-validation (LOGOCV).

2.8 Summary

This chapter outlined the technical background of the thesis, starting with a review of musical expression and the factors that affect it. The chapter then delves into the various techniques and algorithms that have been developed to analyze and replicate the expressive elements of a musical performance using computational modeling approaches. The use of expressive features to quantitatively measure a performer's playing style is also explored which will be used extensively in Chapter 3, 4, 6 and 7. The chapter includes an overview of statistical models and music similarity measurement algorithms, which are used in Chapter 3 to analyze musical expression and identify patterns and characteristics in a performer's playing style. The definitions of machine learning and deep learning algorithms are also covered, as well as their potential use in music information retrieval. Finally, the chapter discusses various evaluation metrics and techniques that have been developed to assess the reliability and performance of computational models.

Chapter 3

Pianist Identification via Probabilistic Density Estimation

3.1 Introduction

The artistry with which a pianist interprets and performs the notes of a piece of music is often a reflection of their own personal style. It is crucial for a pianist to play the notes and rhythms as they are written in the score, but it is equally important for them to infuse each note with their own unique and expressive style. Given that individual notes form the basic foundation of all Western classical musical compositions [171] and pianists have the ability to shape those notes to convey their emotions, this chapter will focus on analyzing and understanding the styles of various pianists in terms of individual notes. Therefore, features at the note level will be utilized to model expressive performances.

While it is not common to identify performers based solely on their interpretation of individual notes in real life, this approach can be useful as a starting point for research. For example, it may be easier for researchers to study the nuances of a pianist's style by focusing on individual notes rather than an entire piece of music. Additionally, it is also worth noting that the variations in melody and rhythm that result from a succession of notes may affect the overall musical style; but, it is not yet clear if individual notes are capable of expressing style on their own. Further research is needed to investigate this question and determine the extent to which note-level features contribute to a performer's style. Moreover, understanding the relationships between note-level expressive features and performers' styles is essential for researchers in the field of music information retrieval (MIR), as it would provide a solid foundation for developing hierarchical analyses of musical styles. By comprehending these relationships, MIR researchers can also develop methods for quantifying and modifying musical styles as well as reproducing them. These issues will be addressed in this chapter by proposing a technique for identifying pianists using hand-crafted expressive features extracted from isolated notes and modeling them using various statistical models.

Among the various factors that shape the emotional impact and character of a piece of music, expressive tempo and timing are among the most significant. They allow performers to create a sense of narrative within the music and express a wide range of emotions. As previously discussed in section 2.4.2, the tempo of a piece of music refers to the speed at which it is played and can be indicated in beats per minute (bpm) or a specific note value. Global tempo refers to the overall speed of a piece and is usually indicated in the music notation with a metronome marking. Local tempo, on the other hand, refers to the speed at which a specific section or passage is played and may be considered as deviations from the global tempo. Timing, on the other hand, refers to the precise placement of musical events such as notes and rhythms within a measure, while expressive timing refers to intentional deviations from the local tempo in order to shape the music and evoke a desired emotion in the listener. These variations can be measured at the note level by comparing the actual performance with a mechanically precise version of the piece. As previously reviewed in section 2.4.2, numerous studies have also attempted to identify pianists based on timing and tempo features. Therefore, we have selected these features as distinguishing factors for identifying famous pianists, and section 3.3.2 proposes the technique for feature extraction.

In addition to time and tempo considerations, the pianist's expressive dynamics and articulation also play a major role in shaping their unique style. Expressive dynamics refer to the variation in volume and intensity of the music, while articulation refers to the manner in which the notes are played, such as the attack and decay of the sound, the use of legato or staccato, and the use of accents. These elements contribute to the overall character and emotion of the music and can be controlled by the pianist to add expressivity to their performance. In section 3.3.2, we propose a technique for extracting expressive dynamics and articulation features at the note level in order to better understand the styles of different pianists.

In conclusion, the interpretation and performance of individual notes by a pianist plays a significant role in shaping their distinctive style. By studying note-level features such astiming, dynamics, and articulation, researchers and individuals can better understand and analyze the styles of various pianists. This understanding can also have practical applications, such as helping piano students to imitate the styles of virtuoso pianists or developing methods for quantifying and modifying musical styles. In this chapter, we propose techniques for extracting these expressive features at the note level in order to better understand the relationships between them and pianists' styles.

The organization of the remainder of the chapter is as follows: In Section 3.2, a new dataset comprising 9 performers is introduced. The feature extraction, feature distribution estimation, and pianist identification methods are described in Section 3.3. The experiments and results of pianist identification are presented in Section 3.4, followed by a discussion and analysis of the results in Section 3.4.3. Finally, the chapter is summarized in Section 3.5.

3.2 Dataset

The data used in this study is obtained from the International Piano-e-competition [172] and consists of performances played and recorded on a Yamaha CFX concert Grand Piano. As shown in Table 3.1, the dataset includes performances by 9 virtuoso pianists playing the same four movements of Sonata in B-flat Major, D960 by Franz Schubert. These movements are Molto moderato, Andante sostenuto, Scherzo: Allegro vivace con delicatezza – Trio, and Allegro ma non troppo – Presto. The number of notes in each movement is described in Table 1. All of these performances have been recorded in both raw audio and MIDI format using state-of-the-art Disklavier Pro recording technology. According to a study by Goebl and Bresin [173], the recording and reproducing capabilities of the Yamaha CFX

Composer	Piece	Movement	No.of Notes
F.Schubert	Sonata in B-Flat Major,D960	I	7582
		II	2005
		III	2717
		IV	4676

Table 3.1: Dataset details with the number of notes for each composition.

were tested and found to have slightly more precise onset capturing (+/- 10 ms) than its reproduction (-20 to +30 ms), and a systematic (linear) error in recording over time. In terms of dynamics, the Yamaha CFX performed well only in a wide middle range.

For the purposes of this thesis, we are primarily interested in working with the symbolic representation of piano music, particularly MIDI data. The advantage of MIDI over raw audio is that it explicitly describes each note event with four parameters: onset (the start time of a note), offset (the end time of a note), pitch (a numerical value for each note ranging from 0 to 128), and velocity (loudness of the notes). This eliminates the need for manual annotation of the data, which can be time-consuming and require expert annotators. However, using symbolic data like MIDI may not capture the nuances and richness of the raw audio data, although the audio and MIDI files are aligned with an accuracy of approximately 3 milliseconds.

3.3 Methodology

As shown in Figure 3.1, the proposed method for pianist identification is divided into four main stages. The first stage involves calculating the norm performance by aligning multiple renditions of the same piece using a symbolic music alignment method. This norm performance and the digital score, both rendered in MIDI format, are then used to align with the corresponding performances in the dataset, followed by the extraction of norm and score deviation features. Next, we estimate the distribution of each feature such that the distributions themselves may serve as a compact representation of the performers' individual artistic style. Three statistical models are thoroughly analyzed and presented for their suitability in modelling these distributions in Section 3.3.3. Finally, we compute a similarity estimation of the feature distribution for each individual feature as well as the fused features to identify pianists accurately, which is illustrated in Section 3.3.4.

3.3.1 Symbolic Music Alignment

Symbolic music alignment is a process that aligns a symbolic music representation, such as a MIDI file, with a reference score or performance MIDI. It is used to match each note in a music performance with the corresponding note in a score or reference performance, allowing for the analysis and comparison of the performance with the notated score. This is useful for a variety of purposes, such as analyzing the expressive performance of a musician, comparing different performances of the same piece, or generating a new symbolic representation of a performance from an audio recording.

For situations where a score MIDI is not available, symbolic music alignment can be used to compute the performance norm, which is the average performance of a piece calculated using a different group of performers. When comparing multiple performances, there may be timing and pitch differences that make it difficult to accurately compare and determine the average performance without an alignment algorithm. By using a symbolic music alignment tool to align the notes of the same piece of music performed by a group of different performers, these discrepancies can be minimized, enabling an accurate calculation of the average performance. The average performance can then be used as a reference signal to compare with individual performances and identify expressive variations. There are several algorithms available for symbolic music alignment, including Hidden Markov Model (HMM) based algorithms and state-of-the-art methods such as those proposed by Gingras and McAdams [174], Chen et al. [175], Nakamura et al. [176]. In this thesis, the HMM-based algorithm proposed by Nakamura et al. [177] was used, which was found to have the highest accuracy for all datasets and superior computational efficiency compared to other algorithms.

The HMM-based algorithm aligns the reference and performance signals using temporal HMMs, detects performance errors, and uses a merged-output HMM [178] to automatically correct these errors in a post-processing (realignment) step. The preliminary alignment and the realignment process is illustrated in Figure 3.2 using brief excerpts from the 2nd movement of Schubert Sonata in B-Flat Major, D960, obtained from the Schubert dataset. In this visual representation, the reference signal represents the score file, while the aligned signal corresponds to the performance. After

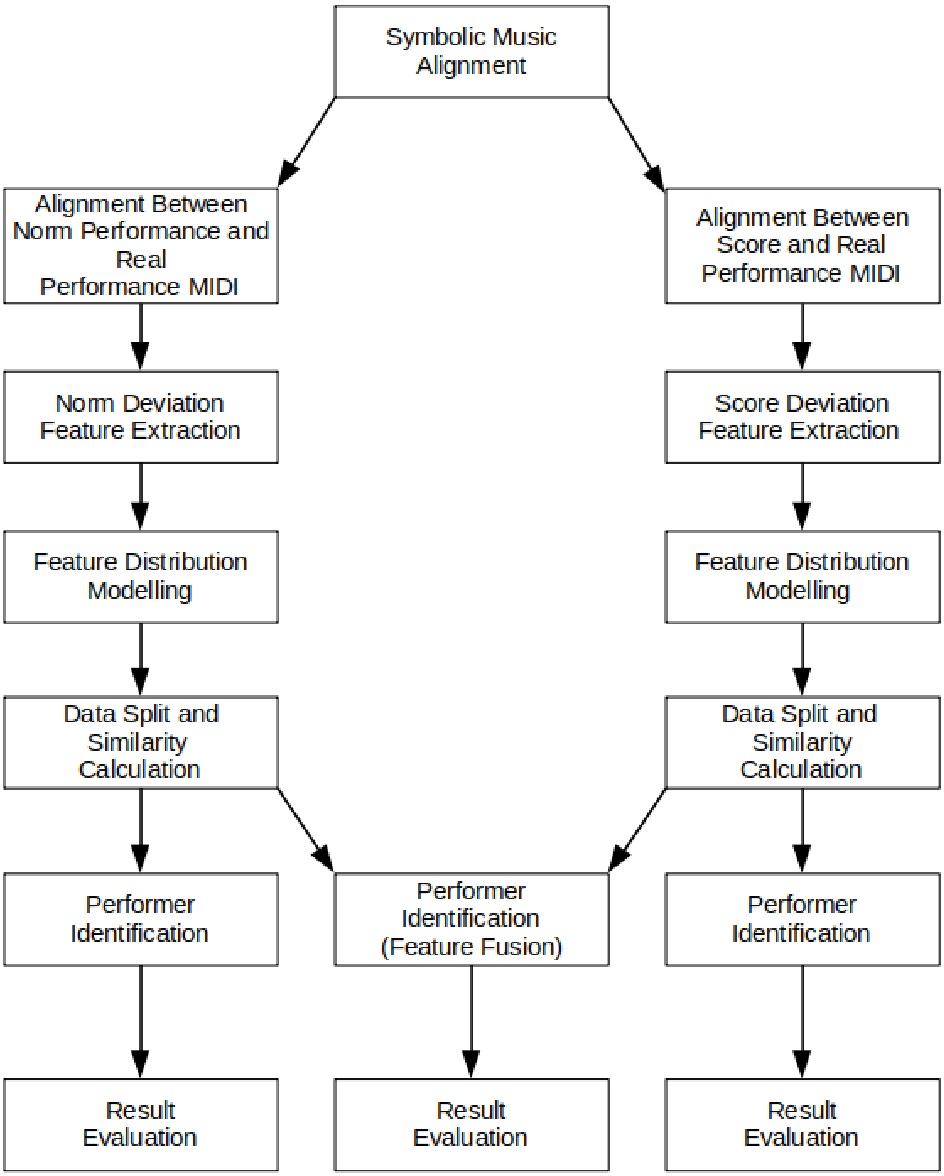

Symbolic Music Alignment
Alignment Between Norm Performance and Real Performance MIDI
Alignment Between Score and Real Performance MIDI
Norm Deviation Feature Extraction
Score Deviation Feature Extraction
Feature Distribution Modelling
Feature Distribution Modelling
Data Split and Similarity Calculation
Data Split and Similarity Calculation
Performer Identification
Performer Identification (Feature Fusion)
Performer Identification
Result Evaluation
Result Evaluation
Result Evaluation

Figure 3.1: Schematic diagram of the proposed method for pianist identification

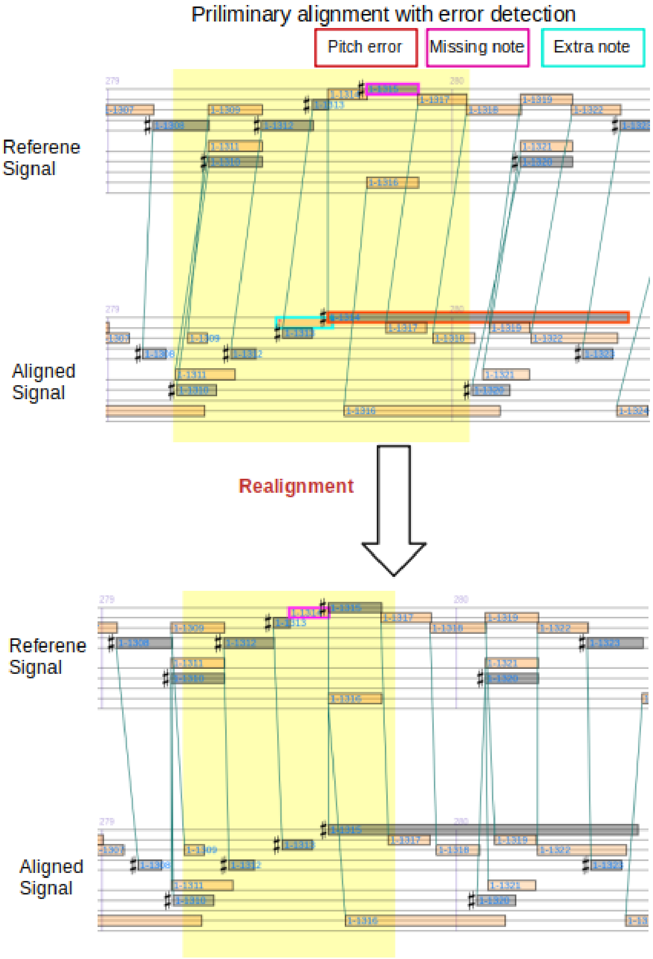


Figure 3.2: Graphical illustration of the alignment and realignment process.

the preliminary alignment, an error detection algorithm is used to identify pitch inaccuracies (highlighted in red), missing notes (in pink), and extra notes (enclosed within cyan rectangles) within the given alignment result. Segments referred to as error regions are then defined, encompassing both the aligned and reference signals. The extent of alignment errors within these regions is examined using segments of varying sizes. An automatic post-processing realignment method is then used to rectify the error regions, which combines the method using merged-output HMMs with a voice(hand) separation method [179]. Importantly, this realignment method does not require prior knowledge of voice information, ensuring more accurate analysis even when voice information is missing or unclear. Finally, In order to achieve accurate alignment with the score, every final alignment was manually inspected, during which extra notes were removed, pitch errors were fixed, and missing notes were interpolated.

There are two modes of the alignment process: the first one is Score-to-MIDI alignment where the score file is a MusicXML file with no performance parameters, and the second one is the MIDI-to-MIDI alignment. The MIDI-to-MIDI alignment algorithm uses any two midi files to find the corresponding notes between them. One of them can be used as a reference signal and the other can be used as the performance signal. The reference signal is first converted into a score file and the score-to-MIDI alignment algorithm is then used for the converted score and the performance MIDI file.

Symbolic music alignment is not only important for aligning score for creating an average performance from several performances of the same composition because it allows for the precise alignment of the different performances, enabling an accurate calculation of the average performance. Without symbolic music alignment, it would be difficult to accurately compare the different performances and calculate the average performance, as there may be timing and pitch discrepancies between the performances and the reference score. By aligning the symbolic representations with the reference score, these discrepancies can be minimized, allowing for a more accurate calculation of the average performance.

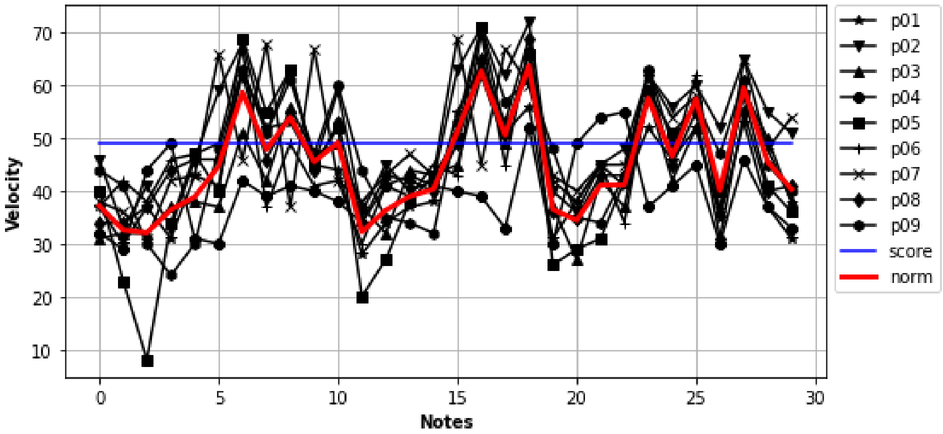


Figure 3.3: MIDI velocities for the first 30 notes of the Sonata in B flat major,D960, Mvmnt. II, as performed by pianists p01-p09

3.3.2 Feature Extraction

3.3.2.1 Score Deviation Feature Extraction

Expressive performance can be defined as the intended deviation from the score which is a purely mechanical rendition of the musical piece in terms of tempo, dynamics and articulation with no expressive variations. These deviations can be used to characterize the performer's individual expressive performance characteristics and to identify common expressive performance principles shared by many performers, as well as the distinctions among them.

In Figure 3.3, the performances of the first 30 notes of the Sonata in B flat major,D960, Mvmnt. II by pianists p01-p09 from our dataset are depicted in terms of midi velocity. To establish a relative standard of reference point for analysis, we used MuseScore software to transcribe the musicXML scores into score MIDIs, with their pre-specified tempos as indicated in their musicXML files. Each note in the score MIDI includes absolute note-on, note-off, and default velocity values. The default midi velocities, which represent a non-expressive and purely mechanical interpretation of the score, is indicated by a straight blue line. The figure demonstrates that the performers tend to deviate from these default velocity values, with some performers exhibiting similar deviations (similar peaks and dips) in specific notes or passages. As demonstrated by Stamatatos and Widmer [1] these similarities may become more pronounced when we take a more global look at each performance by smoothing over longer melodic segments as also illustrated using our data in Figure 3.5a. This phenomenon could be due, to a certain extent, to the correlation between expressive variations and structural elements in the music, such as phrase centers and phrase boundaries, as discussed in prior studies by Clarke [180] and Palmer [181]. However, if a more fine-grained analysis is performed by examining individual notes, dissimilarities among performances can be observed. With the utilization of MIDI data, which provides accurate note information, these expressive variations can be quantified for each note, potentially contributing to the characterization of individual performer's styles.

To represent the expressive deviations, various expressive features can be defined as shown in Figure 3.4. For example, onset time (OT) refers to the starting time of a note, inter-onset interval (IOI) refers to the time between

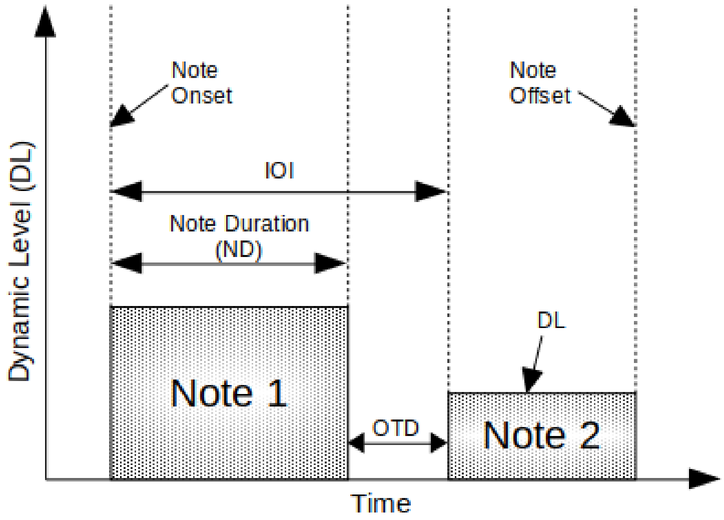


Figure 3.4: Graphical illustration of the parameters that are considered to characterize note-level performance details adapted from Stamatatos and Widmer [1].

the starting times of two consecutive notes, off-time duration (OTD) refers to the time between the ending time of one note and the starting time of the next note, velocity (VL) refers to the loudness or intensity of a note, and note duration (ND) refers to the length of a note.

By comparing the values of these features as they are indicated in the score to the values of the same features as they are played in a performance, it is possible to quantify the deviations of the performance from the score. For example, the deviation of onset time from the score could be calculated as the difference between the onset times as indicated in the score (OT_s) and the onset times as played in the performance (OT_p). Similarly, the deviation of inter-onset interval from the score could be calculated as the difference between the inter-onset intervals as indicated in the score (IOI_s) and the inter-onset intervals as played in the performance (IOI_p). The deviation of additional features, such as the off time duration, note duration and the velocity, can also be calculated in a similar manner.

Table 3.2, shows the proposed score deviation features where, D(x,y) is a feature vector containing the deviations of each aligned note in reference signal x and performance signal y. The features that are calculated by comparing the performances to the score are represented as D(OT$_s$,OT$_p$), D(IOI$_s$,IOI$_p$), D(OTD$_s$,OTD$_p$), D(VL$_s$,VL$_p$) and D(ND$_s$,ND$_p$). Initially, we extract note level features from each aligned note in both the score and the performance midi, which include onset time (note starting time), inter-onset interval, off-time duration (the time between the offset time of one note and the onset time of the next note), velocity, and note duration. For instance, OT$_s$ represents a vector containing the onset times of all notes in the score MIDI, while OT$_p$ represents a vector containing the corresponding onset times in the performance MIDI. We then calculate the deviation of each aligned note of the score and performance midi which is represented by D(OT$_s$,OT$_p$).

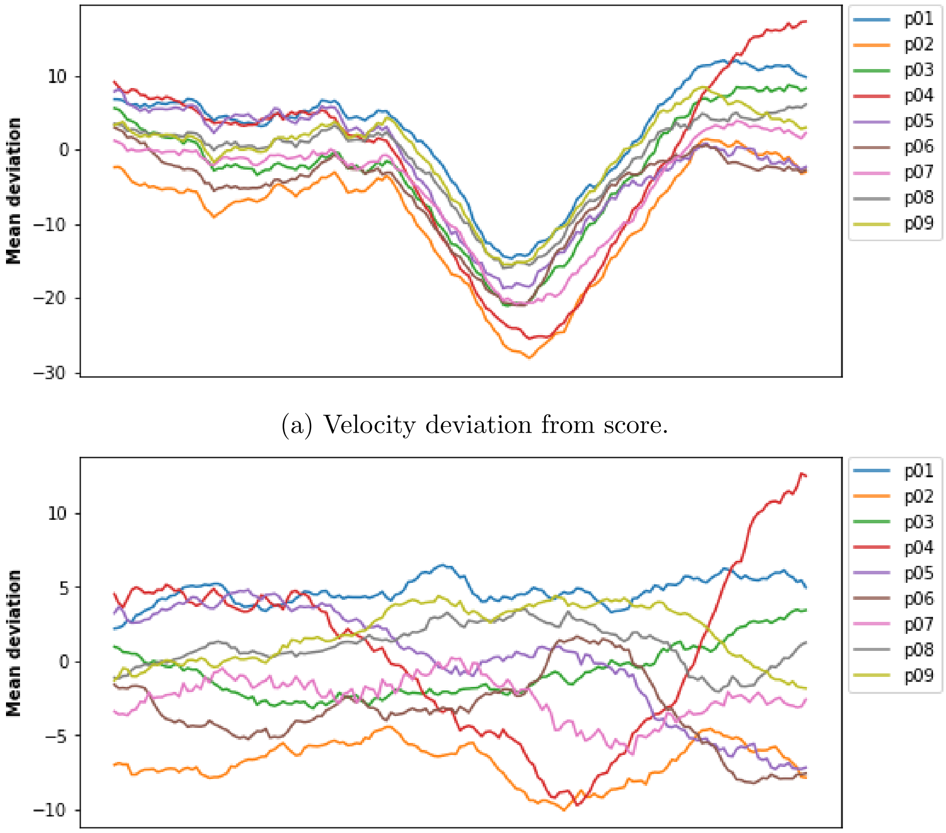
Original Feature Name	Shortened Name	Score Deviation Features	Norm Deviation Features
Onset Time	OT	D(OT$_s$,OT$_p$)	D(OT$_n$,OT$_p$)
Velocity	VL	D(VL$_s$,VL$_p$)	D(VL$_n$,VL$_p$)
Note Duration	ND	D(ND$_s$,ND$_p$)	D(ND$_n$,ND$_p$)
Inter Onset Interval	IOI	D(IOI$_s$,IOI$_p$)	D(IOI$_n$,IOI$_p$)
Off Time Duration	OTD	D(OTD$_s$,OTD$_p$)	D(OTD$_n$,OTD$_p$)

Table 3.2: Summary of the proposed features.

3.3.2.2 Norm Deviation Feature Extraction

One potential problem with the feature extraction by comparing the performances with their respective scores is the difficulty in obtaining digital scores, particularly for older or lesser-known musical pieces. This is due to several factors including: copyright restrictions limiting the accessibility of digital scores without proper authorization from the copyright holder, lack of digitization or online availability, and the possibility of inaccuracies in transcriptions of the original score leading to inaccurate performance feature extraction. This issue can potentially be addressed by considering performance norm, defined as the average performance of a specific piece based on a group of performers, as a reference point for extracting features to discriminate between individual performers.

The idea of performance norm can be better understood from the Figure 3.3, where the bold red line denotes the average performance calculated from pianists p01-p09 playing the same piece in terms of midi note velocities. It is evident that the performance norm closely follows the fundamental shape of the individual performances. In comparison to the score deviation velocity



(a) Velocity deviation from score.

(b) Velocity deviation from norm.

Figure 3.5: The smoothed deviation of velocity from the non-expressive score interpretation (above) and from the norm (below) for pianists p01-p09.

feature analyzed in Section 3.3.2.1, the deviations from the performance norm display less similarity, as evidenced by differences in peaks and dips, when considering longer melodic segments through smoothing, as depicted in Figure 3.5b. This suggests that the structural properties of the piece have limited impact on the divergence of a particular performance from the norm, making the norm an ideal candidate for characterizing individual performer styles, as the features extracted from a norm-performance would exhibit greater distinctiveness towards individual performers.

An empirical study by Stamatatos [73] found that deviation from the average performance is more powerful in representing performer's individuality than deviation from the printed score [182]. In addition, their result also shows that the norm based features are proved to be very accurate for intrapiece tests (training and test set taken from the same piece) and inter-piece tests (training and test set taken from different pieces). To represent these deviations, the same expressive features as in the score feature extraction method can be used. For example, the deviation of onset time from the performance norm could be calculated as the difference between the onset times as indicated in the performance norm (OT_n) and the onset times as played in the performance (OT_p). Similarly, the deviation of inter-onset interval from the performance norm could be calculated as the difference between the inter-onset intervals as indicated in the performance norm (IOI_n) and the inter-onset intervals as played in the performance (IOI_p). The deviation of additional features, such as the off time duration, note duration and the velocity, can also be calculated in a similar manner.

Table 3.2, shows the proposed norm deviation features where, D(x,y) represents the deviation of a vector of numerical values y from a reference vector x. The norm based deviation features are represented as D(OT$_n$,OT$_p$), D(IOI$_n$,IOI$_p$), D(OTD$_n$,OTD$_p$), D(VL$_n$,VL$_p$) and D(ND$_n$,ND$_p$) where OT$_n$, IOI$_n$, OTD$_n$, VL$_n$ and ND$_n$ are onset time, inter-onset interval, off time duration, velocity and note duration respectively, as calculated from the performance norm. OT$_p$, IOI$_p$, OTD$_p$, VL$_p$ and ND$_p$ represents the onset time, inter-onset interval, off time duration, velocity and note duration respectively calculated from the real performance.

3.3.2.3 Feature Standardisation

Feature standardization is a common technique used in machine learning to scale the features of a dataset so that they have zero mean and unit variance. This is important because features with different scales may have a disproportionate influence on the model, which can lead to poor performance. One method of feature standardization is z-score normalization, where the features are standardized by subtracting the mean of the feature values and dividing by the standard deviation. This results in a new set of feature values with a mean of 0 and a standard deviation of 1. Mathematically, this can be represented as:

$$z = \frac{(x - \mu)}{\sigma} \tag{3.1}$$

where x is the original feature value, μ is the mean of the feature values, and σ is the standard deviation of the feature values.

We use z-score feature standardization to standardize both the score and norm deviation features. This allows the features to be compared on the same scale, which is important for accurately measuring the differences between performances. Additionally, z-score normalization can help to mitigate the effects of outliersand standardise the features which can improve the performance of some machine learning models.

3.3.3 Feature Distribution Estimation

Different performers playing the same music piece will inevitably express the piece in their own unique style, as depicted in Figure 3.3 where the valocity variation for nine performers shows that each performer has their own way of expressing the piece to the audience. The individual peaks and dips show how much each performer deviates from both the score and the average performance. These deviations characterize each performer individually. To model the distinct characteristics of performers, we use the distribution of different types of features extracted from the performance, including score deviation features and norm deviation features. We model these deviation distributions using Histograms, Kernel Density Estimations (KDEs), and Gaussian Mixture Models (GMMs) to create compact representations of the performers' idiosyncratic style, which can later be used for pianist identification.

3.3.3.1 Score Deviation Feature Distribution

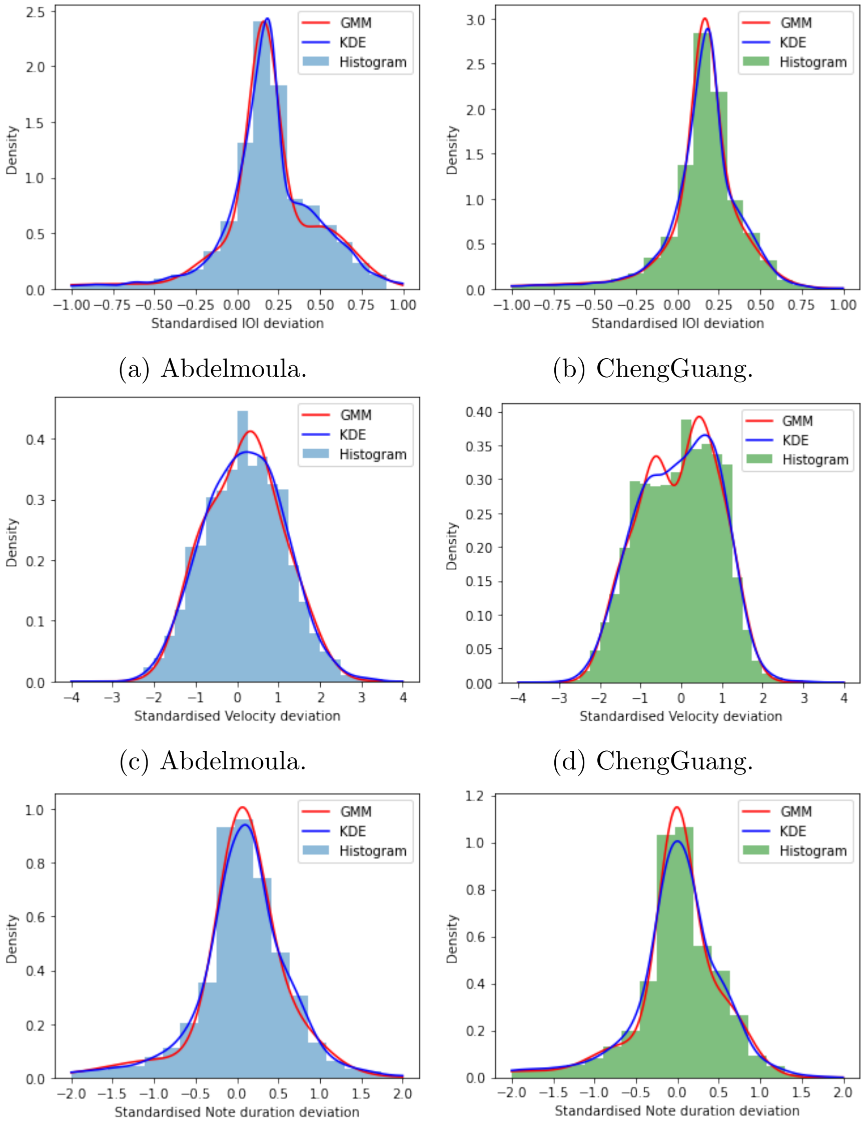
Figure 3.6 shows a comparison of the distributions of three score deviation features for two performers, Abdelmoula and ChangGuang, from the "Schubert 4x9" dataset. The blue line in Figure 3.6 represents the estimated distribution resulting from the use of a Gaussian kernel to estimate the kernel density of score deviation features data from Abdelmoula and ChangGuang. This figure also shows that, the PDF curves of both performers show similar properties to the histogram. The red line represents the Gaussian Mixture Models. We trained the GMMs with 2,3,5 and 7 components. Empirical results show that GMMs do not require more than 3 components to model the distributions. The red pdf curve of the GMMs further demonstrates that it accurately depicts the Histograms and KDEs. We can see the continuous distributions of features for each performer based on these curves, and their differences should represent their individual characteristics.

For example, the global distribution of the Inter onset interval feature is shown in Figure 3.6a and 3.6b and it is apparent that the two distributions have slightly different shapes, where the distribution of ChengGuang looks more symmetrical and has a more gentler slope in the right side than Abdelmoula. Similarly, Figure 3.6c and 3.6d show the global distributions of velocity deviation features for Abdelmoula and ChengGuang, respectively; it is clear that the shape of the two distributions is different, with Abdelmoula's distribution being more symmetrical and the peak appearing to be very close to the point of origin. The distribution of ChengGuang, on the other hand, does not follow a symmetrical pattern. Instead, it has two peaks, one on the right side of the origin and the other on the left side of the origin. The note duration deviation feature distributions for the two performers are shown in the bottom two figures. Dissimilarities between the two distributions may be seen in terms of slope, width, etc., but they are not as readily apparent as in the first two sets of plots.

Based on observations similar to those for the other features, onset time deviation, and off-time length deviation, we assume that the global feature distributions reflect the performer's playing style and the variations in the distributions may be utilised to uniquely identify performers.

3.3.3.2 Norm Deviation Feature Distribution

Similar to the score deviation features, we model the norm deviation feature distributions using Histograms, Kernel Density Estimations, and Gaussian Mixture Models to create compact representations of the performers' idiosyncratic style, which can later be used for pianist identification. Figure 3.7 compares the distributions of three norm deviation features for two performers, Abdelmoula and ChangGuang, from the "Schubert 4x9" dataset. The blue line in the figure represents the estimated distribution using a Gaussian kernel to estimate the kernel density of the norm deviation features data from the two performers. The figure also shows that the probability density function (PDF) curves of both performers are similar to the histogram. The red line represents the Gaussian Mixture Models, which were trained with 2, 3, 5, and 7 components. Empirical results demonstrate that the GMMs do not require more than 3 components to effectively model



(a) Abdelmoula. (b) ChengGuang.

(c) Abdelmoula. (d) ChengGuang.

(e) Abdelmoula. (f) ChengGuang.

Figure 3.6: Distribution of two performers score-deviation features.

the distributions. The red PDF curve of the GMMs further confirms that it accurately represents the histograms and kernel density estimations. By examining these curves, we can see the continuous distributions of norm deviation features for each performer, and the differences between the two distributions should reflect their individual characteristics.

For instance, Figure 3.7a and 3.7b depict the global distributions of the inter-onset interval norm deviation feature for Abdelmoula and ChengGuang, respectively. It is apparent that the two distributions have a positive skew with Abdelmoula's distribution appearing more wider as compared to ChengGuang's distribution. Similarly, Figure 3.7c and 3.7d show the global distributions of velocity norm deviation features for Abdelmoula and ChengGuang, respectively. The shape of the two distributions is somewhat different, with ChengGuang's distribution being more symmetrical and the peak appearing to be on the point of origin, while Abdelmoula's distribution is slightly positive skewed and the peak is away to the right of the point of the origin. The note duration deviation feature distributions for the two performers are shown in the bottom two figures and both of the distributions seem to be positively skewed. In addition. Abdelmoula's distribution seems to peak somewhat to the right of the origin, whereas ChangGuang's appears to peak at the origin itself.

It is presumed that the global feature distributions indicate the performer's playing style, and differences among the distributions may be utilised to identify performers, similar to what we discussed in Section 3.3.3.1.

3.3.4 Pianist Identification using Feature Distribution

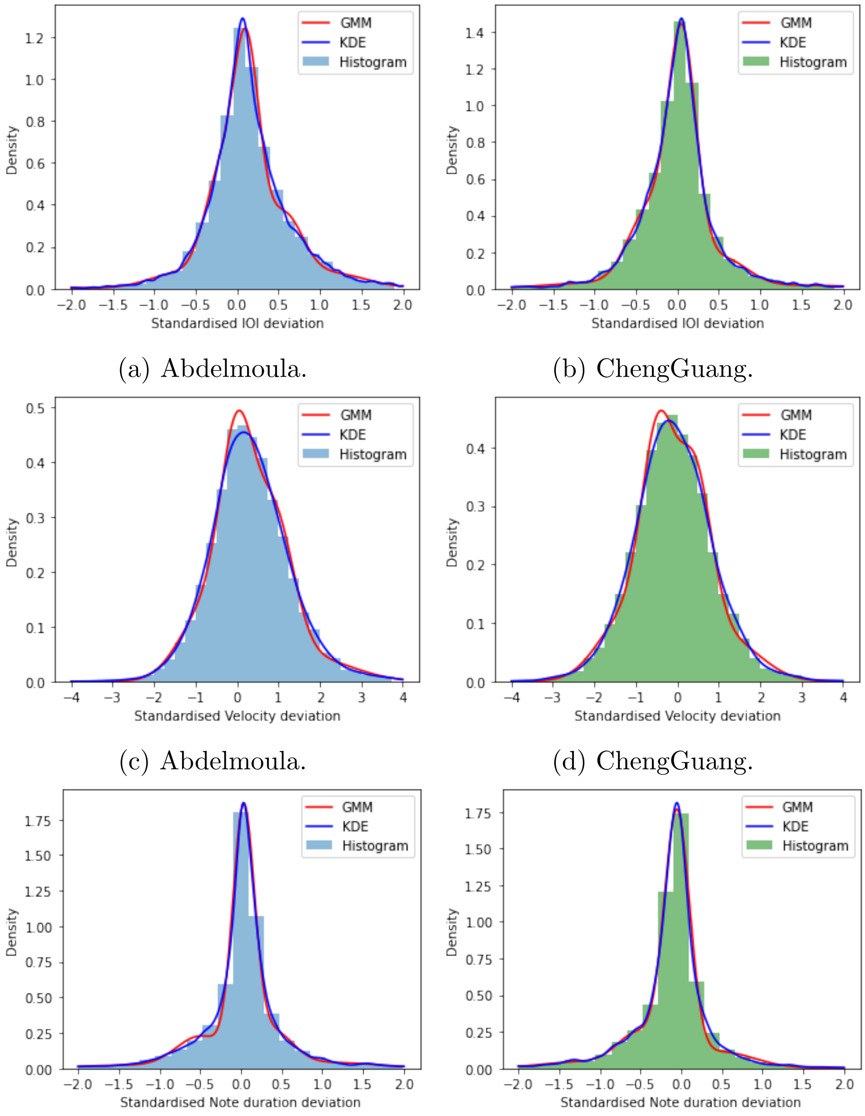
In this section, we describe the pianist identification that includes the feature distribution estimation that we discussed in section 3.3.3 and a similarity calculation. We first discuss the identification methods in terms of individual features and then, we discuss a feature fusion technique.

3.3.4.1 Pianist Identification Using Individual Features

To quantify the differences between performers from the distributions, we take a similarity measurement step of each feature distribution for every performer in our dataset using Kullback-Leibler (KL) divergence [183]. The mathematical formulation of KL-divergence is given below:

$$D_{KL}(P \| Q) = \sum_{x \in \chi} P(x) log(\frac{P(x)}{Q(x)}) \tag{3.2}$$

The Kullback-Leibler (KL) divergence, also referred to as relative entropy, is a statistical measure used to quantify the dissimilarity between two probabil-



(a) Abdelmoula.

(b) ChengGuang.

(c) Abdelmoula.

(d) ChengGuang.

(e) Abdelmoula.

(f) ChengGuang.

Figure 3.7: Distribution of two performers norm-deviation features.

ity distributions. As represented by equation 3.2, it computes the likelihood ratio between two distributions and tells how probability distribution Q diverges from the probability distribution P by computing the cross-entropy minus the entropy.

We measure the KL-divergence for Kernel densities using the approach introduced in [184]. We also calculate the KL-divergence of two GMMs, but since the KL-divergence for two GMMs has no closed form expression, it is not analytically tractable. Hence, we use a variational Bayes approximation method [185] to circumvent this issue. We use KL-divergence to calculate the divergence between an unknown performer's feature distribution and every known performer's feature distribution in the dataset in order to classify the unknown performer. Finding the minimum divergence between the unknown and the known performer's distribution identifies the unknown performer.

3.3.4.2 Pianist Identification Using Feature Fusion Method

Besides classification using individual features, we use feature fusion techniques that are able to combine multiple features. In this study, we combined estimates of similarity across distributions of different features using linear combination with equal weights. We use Leave One Group Out Cross Validation (LOGOCV) technique introduced by Pedregosa et al. [186] which is a variant of k-fold cross validation. LOGOCV splits the data into groups or clusters and systematically excluding one group during each validation cycle, thus training the model on the remaining groups. Unlike conventional cross-validation that randomly selects individual data points for testing, LOGOCV involves leaving entire groups or clusters of data out during each iteration.

First, We concatenate the same feature vectors extracted from all four movements of D960 for each performer. Subsequently, we divided this concatenated feature vector into 8 folds, assigning a unique identification number to each data point within each fold. We then implemented LOGOCV using a loop. In each iteration of the loop (P iterations in total, where P is the number of performers), one performer's data is held out as the test set, while the data from all other performers are used for training. This simulates leaving one performer out for testing. Within each iteration, we modeled the training and test data using distribution models (Histogram, KDE, or GMMs). We calculated the Kullback-Leibler (KL) divergence between the distributions of training and test data. This quantified the dissimilarity of feature distributions for the held-out performer compared to the others. The resulting KL divergence values for each iteration determined which performer was most similar to the held-out performer. Finally, We followed the same steps of similarity estimation for for different features.

This is formulated in equation 3.3:

$$KL_{total} = \sum_{i=1}^{|N|} w_i * KL_{N_i} \tag{3.3}$$

where, N = {N1, N2, N3, N4, N5} denotes the set of statistical models corresponds to OT deviation, IOI deviation, OTD deviation, VL deviation and ND deviation features respectively (see section 3.3.2) which are computed separately. w_i denotes the corresponding feature weight which is set to one in our experiment. However, the feature fusion technique used in this study is not unique. We can combine any 2, 3, 4 or 5 features together to calculate the overall KL-divergence. In the next section, we discuss several experiments using single and fused features to validate pianist identification methods and assess how accurate the methods are for the task.

3.4 Experiments and Results

In this section, we assess the proposed pianist identification method using the proposed "Schubert 4x9" dataset(see Section 3.2). We verify the models effectiveness using Leave One Group Out Cross Validation (LOGOCV) technique and compute the classification results in terms of F-measure which is a way to combine both precision and recall into a single measure that captures both the properties. In addition, normalized confusion matrices are also used to demonstrate the performance of identification for each performers in the dataset.

3.4.1 Baseline Methods

We assess the effectiveness of the proposed features and our pianist identification algorithm against two baseline methods. To begin, we choose KNN and SVM as baselines due to thier popularity in performer identification tasks [170, 19] and apply them on our proposed dataset. Both of the baseline techniques have been evaluated using the individual score and norm deviation features, as well as the fused score and norm deviation features extracted from the dataset.

In the case of performer identification using single feature, we extract the deviation features as outlined in Table 3.2 for all four movements of

D960 associated with each performer. Subsequently, we take each individual feature vector and sliced it into uniform segments, each containing 200 notes. In cases where a segment contains fewer than 200 notes, we apply zero-padding to ensure uniform segment size. Each segment is considered an independent data point that can be used either a training or test sample.

In the context of the fused feature scenario, wherein two or more features are combined, we extract deviation features, as outlined in Table 3.2, for all four movements of D960, each associated with a specific performer. Depending on the chosen combination (e.g., the combination of D(OT$_s$, OT$_p$), D(IOI$_s$, IOI$_p$), and D(VL$_s$, VL$_p$)) for our test case, we initiate the process by segmenting each feature sequence into uniform segments, each encompassing 200 notes. If a segment has less than 200 notes, it is padded with zeros. Subsequently, we concatenate segments from each feature vector that correspond to the same notes. For example, we concatenate the initial segment of the feature vector D(OT$_s$, OT$_p$), comprising 200 notes, with the corresponding initial segments of D(IOI$_s$, IOI$_p$) and D(VL$_s$, VL$_p$)). This process is illustrated in Figure 3.8 using a dummy example where A, B and C are the feature vectors.

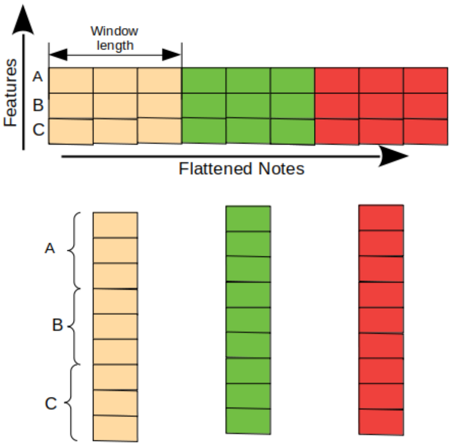


Figure 3.8: Illustration of the Slicing and Concatenation Technique for Combined Features.

In both cases, each segmented feature vector is assigned with the performer ID that was originally assigned to their corresponding piece before segmentation. The proposed dataset is then randomly divided into a training set and a test set, with the training set accounting for 80% and the test set for 20% of the data. During the training phase, feature vectors representing individual or fused features, together with the corresponding performer IDs from the training set, are used to train the baseline models. During testing, however, just the feature vectors are used, and the F-measure is used for evaluation. In subsequent sections, we shall display the outcomes of the two baseline models.

3.4.2 Pianist Identification Results

The effectiveness of the proposed features and the pianist identification technique will be assessed here using the proposed "Schubert 4x9" dataset. Following an initial evaluation and comparison of findings based on individual score and norm deviation features, we will provide results based on a fusion of these two sets of features.

3.4.2.1 Identification Results Using Individual features

In our experiment, a total of 16980 aligned notes have been extracted from each pianist's performance. Since we consider the note level local deviation features in our experiment, the same amount of deviation features is also extracted from the notes. We perform leave one group out cross-validation to separate each performer's data into 8 folds to maintain a high number of cross-validation folds as well as to ensure there are enough data in every test set.

Hence, 2122 notes are designated for each of the first 7 folds, and the last fold contains 2126 notes. We then select a random performer out of the 9 performers and designate one fold of data from that performer as test data. The rest of the folds are considered as training data. The distributions of both the test and training set are calculated using Histogram, KDE and GMMs. The Histogram bins, Kernel density bandwidth and the GMM hyper-parameters are optimised in this experiment and optimum values for each feature are kept constant.

Finally, we calculate the KL-divergence between the test distribution and the training distribution for every performer in the dataset in order to measure the similarity. The minimum KL-divergence value identifies the unknown performer. In other words, we can say that, the corresponding performer's training distribution which has the minimum distance with the

F1-score Feature Model	ΔOT	ΔIOI	ΔOTD	ΔND	ΔVL
KNN(k=5)	0.200	0.016	0.006	0.015	0.018
SVM	0.032	0.036	0.026	0.008	0.018
Histogram	0.532	**0.509**	0.275	0.356	0.502
KDE	**0.595**	0.505	**0.375**	**0.402**	0.536
GMM	0.578	0.466	0.346	0.383	**0.561**

Table 3.3: Pianist identification results based on score deviation features using different statistical models.

F1-score Feature Model	ΔOT	ΔIOI	ΔOTD	ΔND	ΔVL
KNN(k=5)	0.344	0.096	0.108	0.135	0.170
SVM	0.220	0.084	0.114	0.219	0.146
Histogram	0.613	0.451	**0.458**	0.393	0.516
KDE	**0.626**	**0.480**	0.355	**0.477**	**0.564**
GMM	0.600	0.444	0.279	0.389	0.545

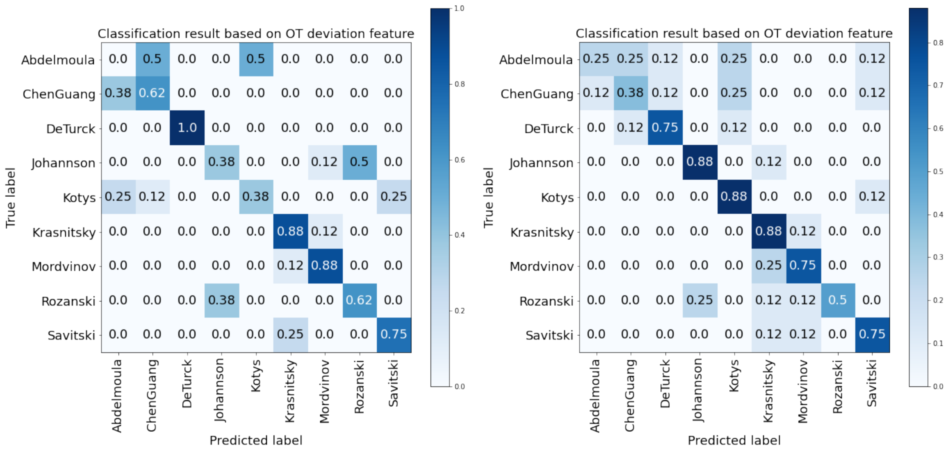
Table 3.4: Pianist identification results based on individual norm deviation features using different statistical models.

test distribution is the identified performer. Table 3.3 shows the identification result in terms of F1-score for each score deviation feature. At the same time, Table 3.4 shows the identification results for each norm deviation feature. Abbreviations used in the Table may be looked up in Table 3.2. The results in bold represent the highest F1-score obtained by any model for each feature. We observe that, of all the features, the onset time (OT) deviation feature performs the best when considered alone. Also, compared to the other models, KDE is the most successful overall. Moreover, both results indicate that our proposed techniques outperform the baseline models for both score and norm features.

The normalized confusion matrices for both the score and norm deviation onset time feature using KDE are also shown in Figure 3.9. The x-axis corresponds to the predicted performers label and the y-axis corresponds to the true performers label.

3.4.2.2 Identification Results Using Fused Features

As discussed in Section 3.3.4.2, feature fusion is a method of combining two or more features to remove redundant and irrelevant features for better



(a) Score.

(b) Norm.

Figure 3.9: Normalized confusion matrix for pianist classification using OT deviation.

Fused	Score Feature			Norm Feature		
Feature	Precision	Recall	F1-score	Precsion	Recall	F1-score
2FF	0.808	0.722	0.762	0.808	0.722	0.762
3FF	**0.829**	**0.806**	**0.817**	**0.871**	**0.819**	**0.844**
4FF	0.828	0.778	0.802	0.828	0.778	0.802
5FF	0.724	0.708	0.716	0.715	0.708	0.711

Table 3.5: Pianist identification using Histogram for different combinations of features.

classification accuracy. The combined features are tested against each statistical models for the pianist identification task. As we see from Tables 3.3 and 3.4, OT deviation and VL deviation features perform better than the other features when considered individually. Hence, it would be practical to consider a combination of features and test them against each model. We show the result obtained by each distribution model for both the score and norm deviation fused features using Equation 3.3.

The feature combination method is not unique and we combine any 2, 3, 4 or 5 features together and assess their performances for the performer identification task. We tried every possible combination and the best combinations for which the Histogram model produces the best result are shown in Table 3.5. We start by fusing the VL and IOI, which we name “2 Fused Feature (2FF)” in Table 3.5. We then combine NL, VL, and IOI, which we'll refer to as 3FF; OTD, VL, IOI, and NL will be referred to as 4FF. Finally, the OT fused with the VL, IOI, NL and OTD is denoted as 5FF. The results in bold correspond to the best performing fusion features when modelled using Histogram. We see that fusing features produces better pianist identification results than using separate features, and 3FF performs best in terms of F1-score regardless of whether we use score or norm deviation features.

Fused	Score Feature			Norm Feature		
Feature	Precision	Recall	F1-score	Precsion	Recall	F1-score
2FF	0.843	0.806	0.824	0.855	0.819	0.837
3FF	**0.907**	**0.903**	**0.905**	0.906	0.901	0.903
4FF	0.899	0.889	0.894	**0.929**	**0.917**	**0.923**
5FF	0.721	0.736	0.729	0.743	0.750	0.746

Table 3.6: Pianist identification using KDE for different combinations of features.

Since, KDE performs better than any other models for the majority of the individual features, we also use it for modelling the distribution of the fused features and evaluate the results using Equation 3.3, which accounts for both the score and the norm deviation of the fused features. To maintain the generalization of the method, we use the same combination of features as used for the Histogram. Hence, the abbreviations would correspond to the same set of features. From Table 3.6, we observe that fusing features produces better pianist identification results by KDE than using separate features. Additionally, when comparing the score deviation fused features, the 3FF performs the best and is more effective in characterising the playing styles of players, whilst the 4FF feature outperforms all the norm features.

Fused	Score Feature			Norm Feature		
Feature	Precision	Recall	F1-score	Precsion	Recall	F1-score
2FF	0.865	0.833	0.849	0.864	0.830	0.846
3FF	0.869	0.861	0.865	0.868	0.861	0.865
4FF	**0.916**	**0.903**	**0.910**	**0.916**	**0.903**	**0.909**
5FF	0.581	0.625	0.602	0.596	0.625	0.610

Table 3.7: Pianist identification using GMM for different combinations of features.

Finally, we show the pianist classification result using the Gaussian Mixture Model (GMM). Table 3.7 shows the classification result in terms of F1-score where each deviation feature in the fused feature set is modelled using a GMM and then using the Equation 3.3, the similarity estimation is

F1-score Feature Model	2FF		3FF		4FF		5FF	
	Score	Norm	Score	Norm	Score	Norm	Score	Norm
KNN(k=5)	0.137	0.116	0.163	0.119	0.159	0.101	0.114	0.187
SVM	0.213	0.193	0.220	0.217	0.312	0.275	0.220	0.359
Histogram	0.762	0.762	0.817	0.844	0.802	0.802	0.716	0.711
KDE	0.824	0.837	**0.905**	**0.905**	0.894	**0.923**	**0.729**	**0.746**
GMM	**0.849**	**0.848**	0.865	0.865	**0.910**	0.909	0.602	0.610

Table 3.8: F1-score by various classification models for different feature combination.

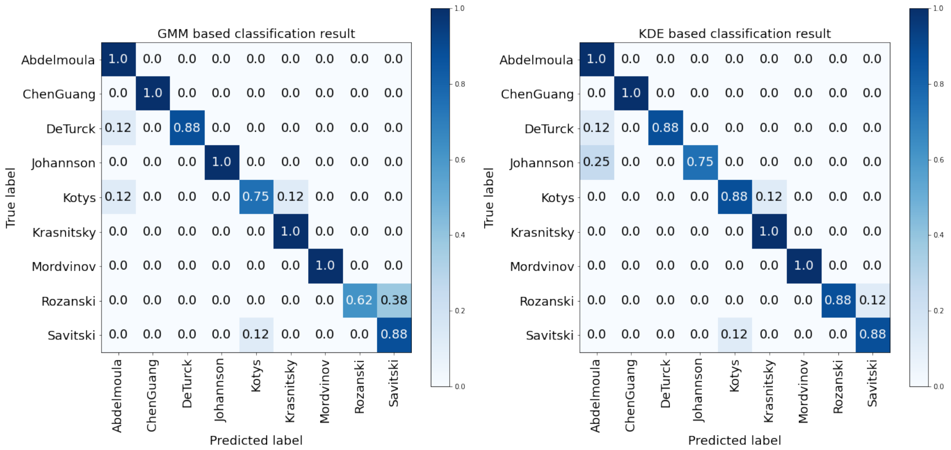
calculated for each feature in the feature set. The mean similarity of the fused features are then used to classify the most propable pianist. The result show that fusing features produces better pianist identification results by GMM than using separate features. In addition, the 4FF performs best in terms of F1-score regardless of whether we use score or norm deviation features.

We also present confusion matrices for 4FF-based pianist identification; Figure 3.10a displays the results achieved by GMM for the 4FF score deviation feature, and Figure 3.10b displays the results obtained by KDE for the 4FF norm deviation feature.

3.4.3 Analysis and Discussion

As discussed in Section 3.3.4, there are two main methods used for pianist identification. First, we classify the performers using three distribution models considering the individual features and second, we use a feature fusion method to identify each performer. Tables 3.3 and 3.4 present the F1-scores for pianist identification based on individual features. Notably, the Kernel Density Estimation (KDE) distribution consistently outperforms the baseline models for both the score and norm deviation features. The onset time deviation (OT deviation) feature, regardless of the model used, consistently demonstrates strong performance compared to other features. This suggests its potential in characterising pianist styles effectively. Norm deviation features, in general, perform well in characterising pianist styles, with the highest F1-score of 0.626 achieved using the KDE distribution.

To provide a more detailed understanding, we generated normalized confusion matrices for the OT deviation feature using KDE. These matrices (Figures 3.9a and 3.9b) showcase strong identification results for several pi-



(a) Score.

(b) Norm.

Figure 3.10: Normalized confusion matrix for pianist classification using 4FF fusion features.

anists, such as DeTurck, Krasnitski, Mordvinov, and Savitski, with F1-scores of 0.75 or higher. Overall, the OT deviation feature performs reasonable well for identifying performers as compare to other features, given that identifying performers from their playing is an exceptionally challenging task. However, certain performers can not be identified correctly since they may share the similar OT feature characteristics.

Considering this problem, we use the feature fusion method (as described in Section 3.3.4.2) for the performer identification task. The results in section 3.4.2.2 demonstrate that fused features works much better than individual features for accurately identifying performers. Table 3.8 compares the baseline models and our suggested models in terms of F1-score for a number of fused features. The performance of our proposed models outperforms the baseline models and KDE performs better overall. The highest F1-score acheived by KDE is 0.923 when we use 4FF norm deviation feature. This indicates that characterising the pianists' distinct styles is improved by combining IOI, OTD, VL, and ND features. In addition, we find that combining OT deviation with other features does not significantly increase performance, despite the feature's strong performance when evaluated alone.

Based on these results, we see that Single features are less effective for pianist identification, while fusion features perform better. This suggests that capturing a pianist's unique style is complex, and a single characteristic may not suffice. The normalized confusion matrices in Figure 3.10 demonstrate the fusion features' effectiveness. Both score and norm deviation fusion features consistently and accurately identify Abdelmoula, Cheng-Guang, Krasnitski, and Mordvinov. Additionally, Deturck, Johannson, and Savitski can be identified with F1-scores of 0.75 or higher using both score and norm features. However, the norm feature is more reliable in identifying Rozanski. In conclusion, our findings illustrate the effectiveness of both the fused features and the models for the identification of pianists, a job that is extremely difficult and often needs expert musicians.

3.5 Summary

In this chapter, we presented an identification method for the recognition of virtuoso pianists using our proposed dataset. To do this, we first designed and extracted five expressive performance features—OT, IOI, OTD, VL, and ND—to describe the playing style of a performer, and then we use three statistical models to determine the distribution of each feature, thereby providing a compact representation of the performer's playing style. Next, we used the LOGOCV technique to split our dataset into training and test sets, and the similarity between the training and test data distributions was computed to identify the pianists. Finally a feature fusion techniqe has been proposed using these five expressive paramters.

Our result shows that the proposed fusion features are robust and accurate to discriminate each performer in the dataset. The classification accuracy results (92% F1-score) achieved by our model demonstrate promising performance, indicating potential improvements over prior research [1]. It is worth noting that 92% is a high success rate in a nine-class classification task with evenly distributed classes, which is a task that is typically seen hard for a human in a setting similar to ours. The comparison between the score based features and the norm based features demonstrated that both types of features works very well in capturing individual performer's styles. This eliminates the need to seek for digital score music, which may be a time-consuming procedure and is sometimes rather difficult to track down. Therefore, when there are several performances of the same work, it's sensible to compare individual performance to an accepted quasi interpretation than comparing to a mechanical performance derived from the score. Our results also demonstrate that features related to expressive timing and loudness are the most informative when fused together followed by the aspect of note duration.

Chapter 4

Parametric Learning for Pianist Identification

4.1 Introduction

Western classical music is a highly structured form of art, with a hierarchical organization of elements. In order to accurately model a piano performance, it is crucial to extract information from each level of this hierarchy. Performers frequently engage with this structural information, for instance, they must be aware of the transitions between movements or sections and convey those transitions to the audience through their playing, while also adding their personal artistic interpretation. Conventional statistical modelling approaches, as we have seen in Chapter 3, may not be able to effectively capture the complex relationship between the structure of the music and the performer's interpretation. This is due to their inability to incorporate the context-dependent characteristics of musical performance, where the performer's interpretation can vary based on the specific piece of music and performance scenario. Conversely, Convolutional Neural Networks (CNNs) are well-suited for learning hierarchical representations of data. However, it's worth noting that even Convolutional Neural Networks (CNNs), may not fully encompass the broader "performance scenario." This aspect may require additional considerations beyond the capabilities of CNNs. Nevertheless, CNNs remain a suitable choice for modeling piano performances, particularly due to their proficiency in capturing local temporal dependencies within the music, which aids in preserving the subtleties of a performance.

In this chapter, we propose a pianist identification model (*convnet-beat*) that utilizes a multichannel 1D CNN, which is a type of neural network that is particularly well-suited for processing data hierarchically. The proposed model is designed specifically to exploit the beat level structure of Western classical music by incorporating a beat-specific kernel in the first layer of the CNN which is discussed broadly in Section 4.2.2. The filters in the first layer are designed with the objective of efficiently learning musically relevant features at the beat level,a crucial element of musical hierarchy. This approach allows for a more powerful abstraction of the performance by capturing the subtle variations in performances at the beat level.

The organization of this chapter is as follows: In Section 4.2, we first present the techniques employed for data preprocessing and input-output representation. We then proceed to describe the architecture of the proposed multichannel 1D CNN in detail. In Section 4.3, we establish the experimental setup and optimize the hyperparameters to identify the optimal model. The results of these experiments, including those for various hyperparameter settings, are presented in this section. We also conduct a comparison of the optimal model with several state-of-the-art multidimensional time series classification models. In Section 4.4, we analyze and discuss the results obtained. Finally, in Section 4.5, we summarize the key findings of this chapter.

4.2 Methodology

In this section, we first discuss the data and the preprocessing and representation methods employed for the multi-channel one-dimensional convolutional neural network (1D-CNN) used in this study. This includes information on the input and output representations of the data used for training the network. We also describe the multi-channel 1D-CNN architecture, which has been specifically tailored to address the task of automatic performer identification. We provide a detailed breakdown of the network's components, including the convolutional layers, pooling layers, and fully connected layers, as well as any other relevant architectural elements. Additionally, we discuss the various hyperparameters used in the network and their impact on the performance of the model.

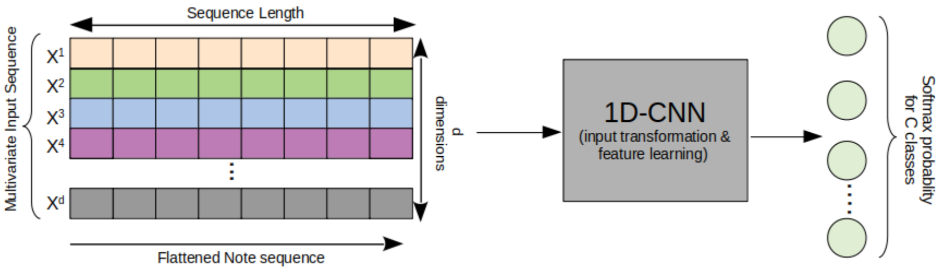

Sequence Length
Multivariate Input Sequence
X1
X2
X3
X4
Xd
d dimensions
Flattened Note sequence
1D-CNN
(input transformation & feature learning)
Softmax probability for C classes

Figure 4.1: Visualisation of classification framework using multivariate input sequence.

4.2.1 Data Pre-Processing and Representation

In this chapter, we utilize the "Schubert 4x9" dataset, which is described in Table 3.1, as the primary dataset for our experiments. The feature extraction method employed is consistent with the method outlined in Section 3.3.2. Furthermore, the set of features used in this study is also identical to those outlined in Table 3.2. In addition to the deviation features outlined previously, we also used the Onset Time (OT), Inter-Onset Interval (IOI), Off Time Duration (OTD), Velocity (VL), and Note Duration (ND) features as baseline features in this study. These features were computed independently from any reference point, such as a score or norm, in order to evaluate their effectiveness as a measure of performer style. To illustrate more, we extracted precise onset times and velocity values for each note as indicated in each performance MIDI and further calculated IOI, OTD, and ND for the same set of notes.

Training CNNs to tackle music information retrieval tasks is a computationally demanding process and needs an ideal input representation. Hence, it is necessary to select an input representation that effectively encodes symbolic music data in a form that allows for efficient processing. While spectrogram representation has been widely utilized for the representation of audio data [148, 187], piano-roll representation has been frequently employed for the representation of symbolic music in the context of training CNNs [188, 189].

Although piano-roll representation is widely used, it has its limitations, including the inability to encode note-off information, which makes it difficult to differentiate between long notes and repeated short notes. This lack of detail can hinder the accurate quantification of expressive performance in music. Therefore, we use an alternative technique to represent music data as a multidimensional input sequence, where each data point corresponds to a note and is ordered by its appearance order (onset time) and pitch in the alignment result. Each dimension in the sequence corresponds to the specific features extracted for each note. This is also visualised in Figure 4.1, where the d-dimensional multivariate input sequence $X = [X^1, X^2, X^3, X^4 \ldots X^d]$ consists of d different univariate sequences is the multidimensional input sequence with each $X^i \in \mathbb{R}^M$. Each X^i represents a separate univariate input sequence, $X^i = [x_1, x_2, x_3, \ldots, x_M]$, where the length of X is equal to the number of notes, M. Each univariate sequence represents a deviation feature calculated from each note position. Hence, when using a 1D CNN to process the data, each feature sequence can be treated as a separate channel, where a channel is a dimension of the input data that represents a different feature or aspect of the data. To train our model, we initially converted the performer name labels into one-hot vectors. These one-hot vectors are structured such that for a given performer's name, the vector contains a '1' in the position corresponding to the index of that performer, while all other positions are '0'. When we give the model some test data to classify, it provides a score for each artist showing how likely the test data belongs to each one. The artist with the highest score gets chosen as the final classification for the test data. To summarize, our dataset is represented as $D = (X_1, Y_1), (X_2, Y_2), \ldots (X_N, Y_N)$, where N is the number of datapoints. Each data pair (X_i, Y_i) consists of a multivariate input sequence X_i and its corresponding one-hot label vector Y_i. The label vector has a length of C (no. of different classes in our dataset), with each element, $k \in [1, C]$ being set to 1 if the class of X_i is k and 0 otherwise.

4.2.2 Multi-Channel 1D Convolutional Neural Network

Deep Convolutional Neural Networks were originally developed for image recognition tasks, but have seen widespread success in a variety of other domains [190]. Inspired by the impressive achievements of CNNs, researchers have begun to apply them to the realm of music analysis, particularly music classification [148]. While CNNs have been widely used in the audio domain, there has been relatively little research on the symbolic representation of music. In light of this, we propose the application of a multi-channel 1-dimensional CNNs to hand-crafted feature sequences extracted from MIDI

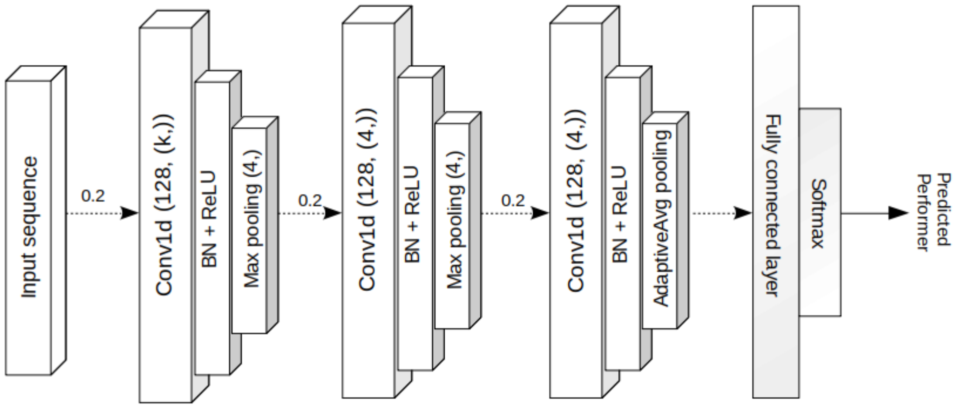


Figure 4.2: The architecture of our proposed multi channel 1D-CNN network. The dropout operation is shown by dashed line.

data for the purpose of performer classification.

Drawing inspiration from the use of CNNs in multivariate time series classification [191, 192, 193], our proposed multi-channel 1D-CNN network employs a similar architecture with modifications in various layers. As illustrated in Figure 4.2, our model is built upon a foundation of convolutional and max-pooling layers, which are followed by an Adaptive average pooling layer and a single fully-connected layer with 9 softmax outputs. k denotes the kernel size of the convolutional filter. Unlike 2D-CNN, which utilizes a two-dimensional filter (width and height) to analyze images, the convolution operation in our case involves the use of a one-dimensional filter (width) to slide over a time series. The filter can be interpreted as a means of applying a generic non-linear transformation to the time series. To illustrate this, we visualise the first convolutional layer and the sunsequent max pooling layer, along with the associated feature maps in Figure 4.3. The input time series data, which typically comprises multiple channels, is processed by a set of one-dimensional convolutional filters. The figure illustrates the process of feature map generation through the application of the convolutional filters, as well as the utilization of a max pooling layer to reduce the dimensionality of the feature maps, thus enabling more efficient processing in subsequent layers. These feature maps encapsulate a wealth of information that is crucial for the classification task.

The filters used within a convolutional neural network can be specifically tailored to incorporate domain-specific knowledge, thereby enabling the ex-

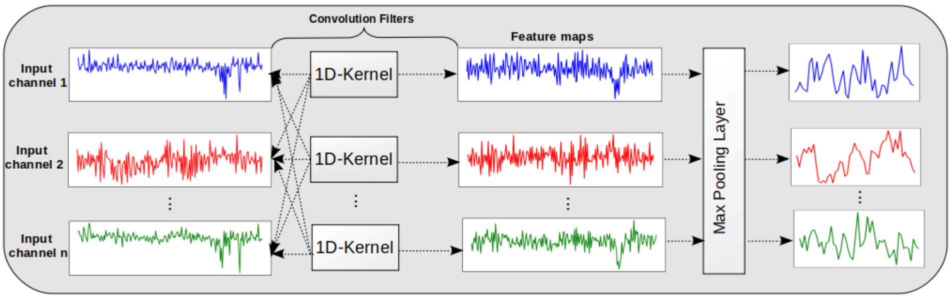


Figure 4.3: Visualisation of the first layer of the proposed multi-channel 1D-CNN. The input is a sequence with n channels. The input to the network is a sequence of data, with n channels. A set of one-dimensional convolutional filters are implemented to learn the structure of the sequence and extract underlying temporal features. Subsequently, a max pooling layer is applied to reduce the dimensionality of the resulting feature map.

traction of specific features from the data. For instance, in image processing, it is common to employ filters with shapes such as (5x5) or (12x12) [194]. Similarly, these shapes are often utilized in music information retrieval research [195, 196]. In particular, the design of filter shapes in the very first layer can be guided by domain knowledge to learn musically relevant contexts [197]. In determining the optimal value of (k,) for the first layer of our CNN, we selected its values based on considerations related to the temporal structure of western classical music, particularly in relation to beats. The choice of k is made with an emphasis on effectively capturing musical elements within a temporal context, specifically, to encompass approximately one beat.

While the time signature information is not included in MIDI files of real musical performances, we found that the corresponding scores in our dataset have time signatures such as $\frac{2}{4}$,$\frac{3}{4}$ and $\frac{4}{4}$. These time signatures indicate the number of beats per measure and the type of note that represents a beat. For example, in 4/4 time, a quarter note or two eighth notes or four sixteenth notes or eight thirty-second notes would receive one beat. In order to further analyze the dataset, we applied the beat tracking algorithm proposed by Dixon [198] to examine the number of notes appearing in each beat of each performance. The histogram in Figure 4.4 illustrates the distribution of notes in each beat. As seen in the histogram, most beats consist of 4, 5 and 6 notes. Therefore, we assessed the performance of our multi-channel 1D

CNN on the validation dataset with kernel sizes {2, 4, 6, and 8} in addition to the commonly used ones for the first convolutional layer, as described in Section 4.3. The result of applying a convolutional filter to an input sequence can be considered as a transformed version of the original sequence that has undergone a filtering process.

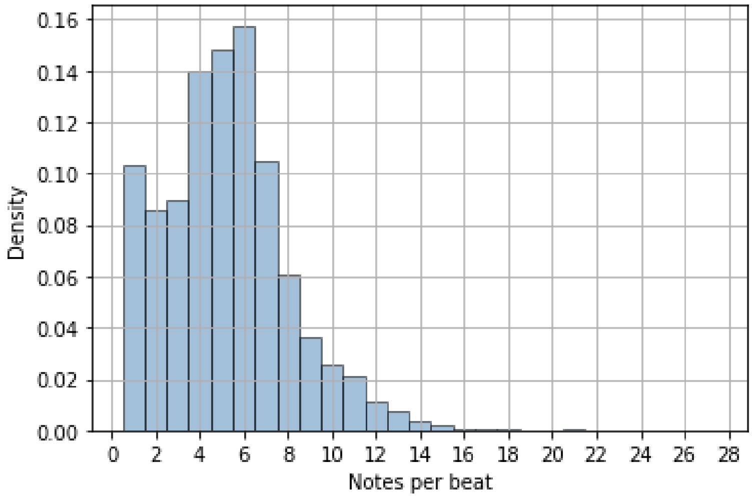


Figure 4.4: Frequency distribution of the number of notes played for each beat across all performances in the dataset.

In addition to the convolutional layer, we implement a batch normalization layer [199] to normalize the activations of the layer across a batch of input data. This serves to stabilize the distribution of activations within each batch and can improve the convergence rate of the network. Additionally, batch normalization can improve the generalization performance of the network by making it less sensitive to the scale of the input data. The transformed data is then passed through a ReLU activation function [158], followed by a max-pooling layer. To prevent overfitting, the data is then input into a dropout layer, where a randomly selected subset (20%) is set to zero. This helps to regularize the model and improve its generalization performance as demonstrated in the study by Nasrullah and Zhao [23] on artist classification. To further reduce the risk of overfitting, we use an adaptive average pooling layer in the final convolutional layer. In this type of pooling, the size of the pooling window and the stride are dynamically determined based on the desired output size, allowing the pooling operation to adapt to the characteristics of the input data. This helps to significantly reduce the dimensionality of the data. The final network consists of three convolutional blocks, each with filter sizes of {128, 128, 128}, followed by a fully connected layer. The final label is produced by applying a softmax function to the output of the fully connected layer. Our basic convolutional block can be formulated as:

$$\begin{aligned} y &= W \otimes x + b, && (4.1) \\ b &= BN(y), && (4.2) \\ r &= ReLU(b), && (4.3) \\ O &= f_{dropout}(r). && (4.4) \\ & && (4.5) \end{aligned}$$

where, $\otimes$ is the convolution operation and BN stands for batch normalisation.

4.3 Experiments

In this section, we first detail the experimental setup and proceed to conduct hyperparameter optimization for the purpose of identifying the optimal model for performer classification. We then present the results in terms of accuracy and macro F1-score. To gain insight into the model's decision-making process, we apply the Class Activation Map (CAM) method to visualize the class-specific regions within the input signal. Finally, we compare the performance of our optimal *convnet-beat* model with state-of-the-art multidimensional time series classification models.

4.3.1 Experimental Setup

When leveraging deep neural networks, specifically convolutional neural networks for sequence data modelling, the primary challenge is obtaining sufficient amounts of representative training samples. The generalization capacity of CNNs is highly dependent on the quantity and diversity of the training data. In scenarios like ours, where the available dataset for training is limited, the model's performance may be severely hindered due to overfitting and under-representation of the underlying data distributions. To mitigate the issue of limited training data, we employed data augmentation by slicing the original multi-dimensional feature sequences into smaller, non-overlapping subsets, thus increasing the size of the training dataset, drawing inspiration from prior research conducted by [1] and [19] on automatic performer identification. This enhances the number of samples available for training, thereby facilitating the ability of the CNN to learn more expressive representations of the underlying data distributions. We empirically determined an appropriate slicing window size of 200, resulting in 86 segments per class in our dataset.

The augmented dataset was divided into 80%/20% to generate the training and validation sets respectively, resulting in 619 and 155 samples. We ensured that a specific performer/piece combination could only appear in either the training or the validation and test set. To ensure that the data adheres to a standard scale, we applied z-score standardization to the dataset. The model was trained utilizing the ADAM optimizer with a learning rate of 0.001 and a batch size of 32. Training was terminated when the model's accuracy failed to improve for 10 consecutive epochs. The categorical cross-entropy loss function was used to calculate the difference between the predicted and true labels. To assess the model's discriminative power, we employed the F-score metric as well as class-wise performance evaluation through a confusion matrix.

Model	**Kernel Size** (l_1, l_2, l_3)	**Accuracy (%)**	**F1-Score**
convnet - baseline	(2,4,4)	54.22	51.89
convnet - beat	(3,4,4)	55.42	55.13
	(4,4,4)	56.62	54.99
	(5,4,4)	56.62	55.36
	(6,4,4)	**62.65**	59.36
	(7,4,4)	57.83	56.39
	(8,4,4)	49.39	45.73
Average		56.11	54.12

Table 4.1: Comparison of Pianist Classification Performance using Baseline Input features across various Models. l_i denotes the layer number.

4.3.2 Hyperparameter Optimisation and Results

In order to optimize the hyperparameters of our proposed multi-channel 1D-CNN model for the classification of performers, we trained and evaluated the

Model	**Kernel Size** (l_1, l_2, l_3)	**Accuracy (%)**	**F1-Score**
convnet - baseline	(2,4,4)	69.88	68.26
convnet - beat	(3,4,4)	71.08	69.87
	(4,4,4)	75.90	74.37
	(5,4,4)	**78.31**	76.95
	(6,4,4)	**78.31**	**77.29**
	(7,4,4)	75.90	74.69
	(8,4,4)	73.49	71.08
Average		74.68	73.21

Table 4.2: Comparison of Pianist Classification Performance using Score-Deviation Input Features across various Models.

Model	**Kernel Size** (l_1, l_2, l_3)	**Accuracy (%)**	**F1-Score**
convnet - baseline	(2,4,4)	85.93	84.44
convnet - beat	(3,4,4)	85.51	84.24
	(4,4,4)	86.71	84.72
	(5,4,4)	88.28	85.90
	(6,4,4)	**90.42**	**90.14**
	(7,4,4)	85.55	84.24
	(8,4,4)	86.51	85.23
Average		**86.98**	**85.55**

Table 4.3: Comparison of Pianist Classification Performance using Norm-Deviation Input Features across various Models

model with different configurations. We evaluated the model's performance utilizing three distinct feature sets, including baseline features (calculated independently from any reference point), score deviation features, and norm deviation features. The results of the model's performance, represented in terms of accuracy and macro F1-score, are presented in Tables 4.1, 4.2 and 4.3. As previously discussed in Section 4.2.2, we employed a beat-specific kernel optimization strategy, wherein the assumption is that each kernel is specifically attuned to accommodate the notes that correspond to a single beat, to achieve optimal performance.

Based on the average accuracy and F1-score metrics, it is demonstrated that the utilization of norm deviation input features results in a higher performance. This suggests that norm deviation features are an appropriate representation for the input to the model. The highest F1-score of 90.14, as recorded in Table 4.3, was also attained with the use of norm deviation features as input in the *convnet-beat* model, with a kernel size of 6 in the first convolutional layer. This serves as evidence that the beat-specific filters employed in the model are capable of learning relevant temporal dependencies that are discriminative towards pianists.

convnet-beat	**Channel**	**Layers**	**Accuracy (%)**	**F1-Score**
Models with Reduced Channel	16	3	79.51	77.23
	32	3	86.51	85.79
	64	3	86.74	84.33
Models with Reduced Layer	16	2	60.24	53.89
	32	2	74.69	73.60
	64	2	79.52	77.30
	128	2	84.33	82.38
Default Model	**128**	**3**	**90.4**	**90.1**
Models with More Channel	256	3	85.54	83.43
	512	3	83.13	80.81
Models with More Layers	128	4	89.15	88.44
	128	5	87.95	87.15

Table 4.4: Comparison of Pianist Classification Performance using Norm-Deviation Input Features across variants of *convnet-beat* model.

In order to assess the relationship between model performance and the number of channels and layers, we conducted a series of experiments on the *convnet-beat* model, using a fixed kernel size of 6 and norm deviation input features. We selected this particular model for experimentation due to its superior performance, as evidenced by the highest accuracy and F1-score values among all models tested. The results of this experimentation are presented in Table 4.4, which shows the model performance for both reduced and increased number of parameters, specifically channels and layers. It can be observed that as the number of layers is decreased, there is a significant reduction in performance, and although increasing the number of channels does not result in an improvement in performance, increasing the number of layers results in consistent performance. However, it should be noted that an increase in the number of layers may result in overfitting, particularly when working with a small dataset like the one in this case and this could result in decreased accuracy on the validation set.

The *convnet-beat* model, configured with a kernel size of 6, channel count of 128, and 3 layers, exhibited the highest performance among all models evaluated. Utilizing norm-deviation features as input data resulted in improved performance, as indicated by the highest accuracy and F1-score values in bold, shown in Table 4.4. This model was considered as the optimal configuration and was selected for comparison against state-of-the-art multidimensional time series classification models in Section 4.3.4.

4.3.3 Identifying Contributing Regions for Performer Prediction using Class Activation Maps

In order to gain a deeper understanding of the underlying mechanisms of the *convnet-beat* in the context of performer identification, we employ the Class Activation Map (CAM) method. CAM, first introduced by Zhou et al. [200], is a visualization technique, commonly used for image-based tasks, allows for the identification of discriminative regions within an images. A one-dimensional CAM, which was previously presented by Wang et al. [193] for time series classification, is employed in order to uncover the regions of the input sequence that exert the most significant influence on the output at each layer of our neural network, as depicted in Figure 4.5.

The incorporation of the Adaptive Average Pooling (AAP) layer, which operates similar to a Global Pooling Layer (GAP), facilitates the computation of the CAM. For example, to compute the CAM for the final convolutional layer, we first extract the output of that layer, denoted as H(t), a multivariate sequence with d variables. For each variable $i \in [1, d]$, we can represent the univariate sequence as $H_i(t)$, which corresponds to the activation of the i^{th} filter in the final convolutional layer. The output of the AAP for filter n can be formulated as:

$$F_i = \sum_t H_i(t) \tag{4.6}$$

Moreover, we can denote the softmax weights between the i^{th} filter output and the class output neuron as w_i^c. Thus, the input to the final softmax function can be represented mathematically as:

$$\begin{aligned} h_c &= \sum_i w_i^c \sum_t H_i(t) \\ &= \sum_i \sum_t w_i^c H_i(t) \end{aligned} \tag{4.7}$$

We finally calculate the Class Activation Map (CAM), which is a univariate sequence. Each element at step t in the sequence is a weighted sum of the M datapoints. The weights are learned by the convolutional layer through optimization during the training phase. The Class Activation Map for class c can be mathematically defined as:

$$CM_c(t) = \sum_i w_i^c H_i(t) \tag{4.8}$$

Where $CM_c(t)$ denotes the class activation map, a visual representation of the regions of the input sequence that are most critical to the classification process. It quantifies the importance of the activation at temporal location t in relation to the classification of the input sequence as class c. To ensure the output length of the last convolutional layer is the same as the input length, we have upsampled the output with linear interpolation.

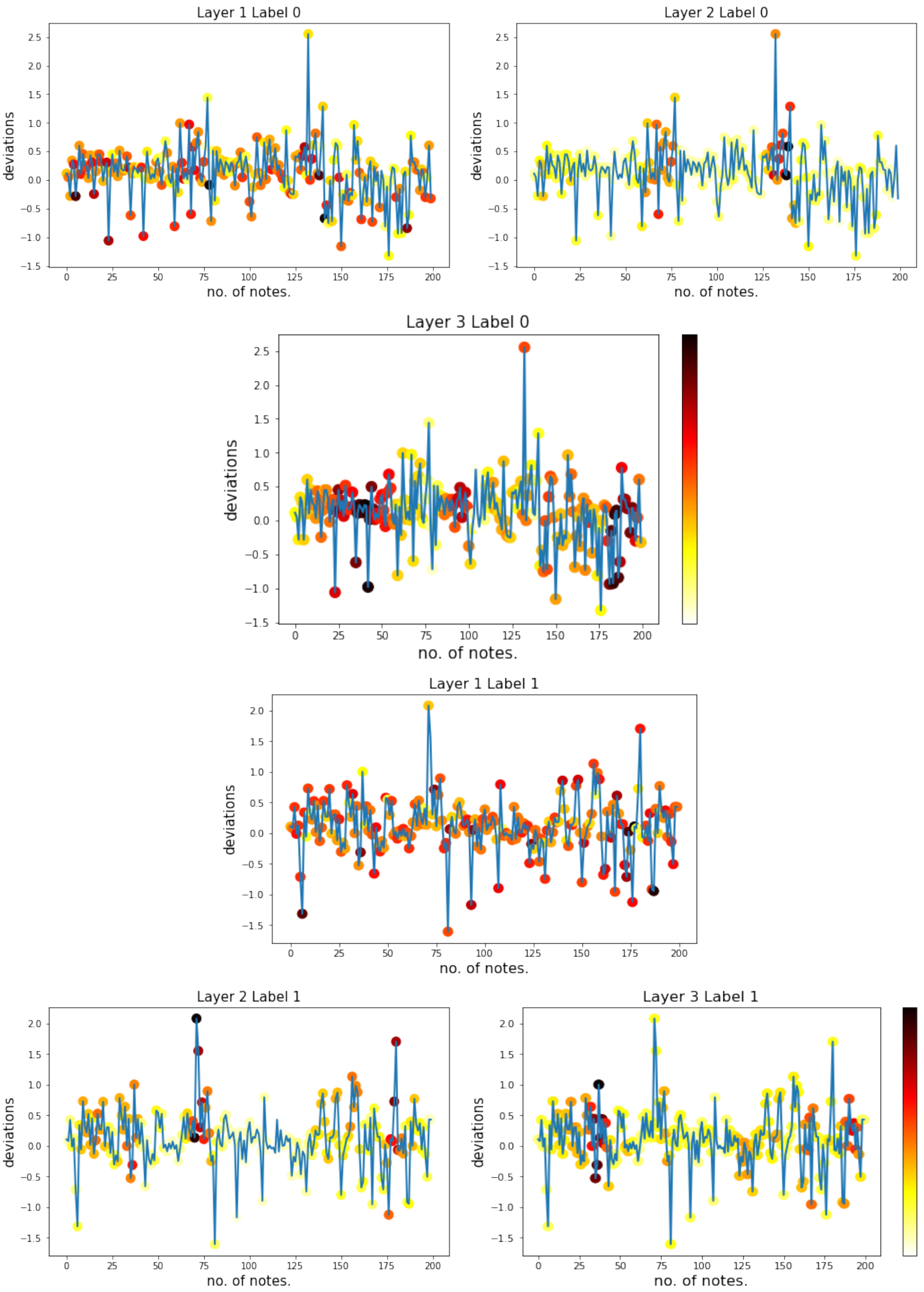
4.3.4 Comparison of *convnet-beat* with State-of-the-Art time series classifcation Models

In this section, we present a comparison of the proposed *convnet-beat* model with several state-of-the-art multidimensional time-series classification models using our "Schubert 4x9" dataset for pianist classification. All of these models were trained for 100 epochs using the Adam optimizer, with categorical cross-entropy as the loss function. Through experiments with various epoch settings, we identified that 100 epochs represented the optimal point where each model reached a performance plateau. The comparison of the models is presented in Table 4.5, which displays the number of trainable parameters for each model and the corresponding classification accuracy. The results indicate that the *convnet-beat* model outperforms the other models, with a classification accuracy of 90.42%.

Model	**Total params**	**Accuracy (%)**
XceptionTime [201]	400770	84.33
ResCNN [192]	259210	83.13
InceptionTime [201]	456777	83.13
ResNet [193]	481417	81.92
FCN [193]	268809	80.72
LSTM {'n_layers': 3, 'bidirectional': True}	570609	62.65
convnet-beat	202377	**90.42**

Table 4.5: Comparison of the Proposed *convnet-beat* Model with Other State-of-the-Art Models on our dataset in terms of Classification Accuracy

Notably, the *convnet-beat* model has the lowest number of parameters



Layer 1 Label 0

Layer 2 Label 0

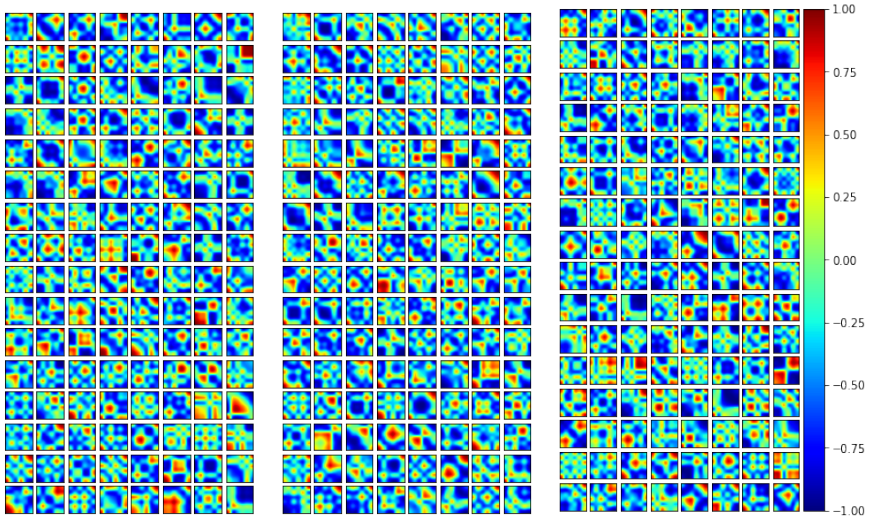
Layer 3 Label 0

Layer 1 Label 1

Layer 2 Label 1

Layer 3 Label 1

Figure 4.5: Class Activation Map (CAM) illustrates the contribution of various sequence regions in each layer to correct class identification for two classes in our dataset using the conv-beat model. Dark Red regions indicate high contribution, while yellow regions indicate minimal contribution.



(a) Layer 1.

(b) Layer 2.

(c) Layer 3.

Figure 4.6: Visualization of learned filters from the *convnet-beat* model. The filters shown represent a sample of filters from different channels, c_i and were learned during the training process of the first, second and third convolutional layer.

among all the models, with 202377, while still achieving the highest classification accuracy. On the other hand, the LSTM model, which is a commonly used deep learning architecture for sequence data, has the largest number of parameters (570609) yet it achieves the lowest classification accuracy of 62.65%. These results demonstrate the effectiveness of the *convnet-beat* model for the task of multi-class pianist classification.

4.3.5 Comparison between convnet-beat and statistical models

In this section, we delve into a comparative analysis between the convnet-beat model and the similarity-based identification models proposed in Chapter 3. This comparison is essential to understand the strengths and limitations of different approaches when it comes to pianist identification. Table 1 presents the performance of these models in terms of both the score and norm deviation characteristics. Histogram provides a good balance between precision and recall for both Score and Norm features, with Norm feature slightly outperforming Score feature in terms of F1-Score. However, KDE

Model	Score Feature			Norm Feature		
	Precision	Recall	F1-score	Precision	Recall	F1-Score
Histogram	0.829	0.806	0.817	0.871	0.819	0.844
KDE	0.907	0.903	0.905	0.929	0.917	0.923
GMM	0.916	0.903	0.910	0.916	0.903	0.909
convnet-beat	0.783	0.766	0.773	0.913	0.892	0.901

Table 4.6: Comparison of the proposed *convnet-beat* model with the similarity based identification models proposed in Chapter 3.

shows excellent results, especially for the Norm feature, where it achieves a precision of 0.929, a recall of 0.917, and an F1-score of 0.923. This indicates that the KDE has been effective in capturing the nuanced aspects of Norm performance. The Gaussian Mixture Model (GMM) exhibits similar performance to that of KDE. It yields nearly identical results for both the Score and Norm features, suggesting consistent performance irrespective of the feature type. On the other hand, convnet-beat model achieves its best performance using the Norm feature, with an F1-Score of 0.901. While this is slightly lower than the KDE and GMM models for the Norm feature, it's noteworthy considering that neural network models often require substantial data and fine-tuning.

In conclusion, Statistical models (KDE and GMM) seem to perform better when modelling the pianists' individual artistic style, especially with the Norm feature. This could be due to their capability to capture the distribution of features effectively. The convnet-beat, despite being a neural model, shows competitive performance. This indicates the power of neural networks in capturing subtle variations, especially when specialized structures like beat-specific kernels are employed. This model might offer more nuanced insights into the data or even outperform the statistical models with more data or further tuning. The differentiation in performance between Score and Norm features across models indicates the importance of feature selection in pianist identification. The Norm feature generally provides better or comparable results, suggesting that the features extracted from norm performance is a vital aspect of capturing the unique styles of pianists.

4.4 Result Analysis and Discussion

The *convnet-beat* model's predictions can be further understood by visualizing the learned filters from each layer, as depicted in Figure 4.6. This allows for an interpretation of the information the model is learning from each convolutional layer. The Gramian Angular Summation Field (GASF) technique is utilized to transform the temporal filters in each layer into a 2D image representation, as outlined in [202]. GASF is a method for intuitively mapping multi-scale correlation in 1D space by representing each value in a sequence as a pixel in an image. To acheive this, the input sequence is first normalized to ensure that all values fall within the range of [0, 1]. Subsequently, the sequence is mapped into polar coordinates by encoding the values as the angular cosine and the time steps as the radius. For a given sequence $S = \{s_1, s_2, \ldots, s_n\}$, the process of transforming the temporal filters in each layer of the *convnet-beat* model into a 2D image representation is mathematically defined as follows:

$$\tilde{s}_0^i = \frac{s_i - min(S)}{max(S) - min(S)} \tag{4.9}$$

$$\phi = \arccos(\tilde{s_i}), 0 \leq \tilde{s_i} \leq 1,\ \tilde{s_i} \in \tilde{S} \tag{4.10}$$

$$r = \frac{t_i}{K},\ t_i \in \mathbb{N} \tag{4.11}$$

Where K is a constant factor that regularizes the span of the polar coordinate system and t_i represents the time step. The temporal correlation within different time intervals can then be identified by considering the trigonometric sum between each point, defined as:

$$\begin{aligned} GASF &= \cos(\phi_i + \phi_j) \\ &= \tilde{S}' \cdot \tilde{S} - \sqrt{I - \tilde{S}^2}' \cdot \sqrt{I - \tilde{S}^2} \end{aligned} \tag{4.12}$$

Where $I = [1, 1, \ldots, 1]$ is the unit row vector. The inner product can be defined as $< x, y >= x \cdot y - \sqrt{1 - x^2} \cdot \sqrt{1 - y^2}$ and $< x, y >= \sqrt{1 - x^2} \cdot y - x \cdot \sqrt{(1 - y^2)}$ and Gramian angular summation field can be represented as quasi-Gramian matrices $[< \tilde{s}_1, \tilde{s}_1 >]$ [202].

In Figure 4.6, we visualize the filters of the first channel for all three layers in the *convnet-beat* model. The convolution operation is performed by sliding the filters, which act as a weighted moving average, over the input data to extract local temporal dependent features among different time intervals. In time series data, these features may be the local maxima (peaks) and minima (dips). As we move deeper in the layers, we can observe that the filters have learnt more complex patterns, probably multiple peaks and dips. Although there is similarity in the patterns observed across layers, there are also unique patterns, indicating the heterogeneity of the local patterns learned by the filters across multiple layers. Additionally, it is observed that the weights in the first layer tend to have higher magnitudes compared to those in subsequent layers. This could be a result of the filters in the first layer capturing more information in the input data, which supports the use of beat-specific kernels in the first layer to capture beat-level micro information.

In order to gain a deeper understanding of the input signal's contribution to the decision made by the *convnet-beat*, we have employed the use of class activation maps (CAMs) to identify the specific regions within the input signal that had the greatest impact on the network's output. This is demonstrated in Figure 4.5 using segments of Schubert's Sonata in B-Flat Major, D960, movement IV, performed by two different pianists, as part of the test set. It is evident that for both of the performers, many discriminative regions have been identified in the first layers, whereas there are not many discriminative regions identified from the second and third layers. This suggests that using a beat-specific filter in the first layer can learn the micro-variations injected by performers to each note, such as variations in timing, velocity and articulation. Additionally, this creates the avenue for experimenting with a measure-specific filter that can capture the variations within each measure, such as variations in dynamics, harmonic structure, and melody, and eventually help to capture the global temporal relationship of the performance. This can lead to a better understanding of how the neural network is able to discern subtle variations in performances and aid in the design of more sophisticated music-specific models.

Neural networks, particularly convolutional neural networks, are highly prone to overfitting due to their inherent complexity and the potential for training with limited data. In our case, the convnet model was expected to exhibit overfitting given the small size of the dataset. However, we were able to mitigate this issue by implementing several techniques to improve the model's generalization capabilities. Specifically, we employed batch normalization in each layer, which is known to improve training speed and generalization. Furthermore, the use of an Adaptive Average Pooling layer before the fully-connected layer greatly reduced the number of parameters, thus reducing the risk of overfitting. Additionally, the incorporation of dropout in each layer further helped to alleviate overfitting. By incorporating these technical tricks into the network architecture, we were able to achieve better accuracy on our small dataset that consists of recorded MIDI files from computer-controlled pianos, which are often limited in size and variety.

To sum up, our proposed *convnet-beat* model demonstrates the ability to accurately identify performers from unseen musical excerpts. Despite being trained on a relatively small dataset, the model exhibits minimal overfitting. The utilization of a beat-specific filter in the first convolutional layer allows for effective learning of micro-variations introduced by performers. Additionally, the model achieves superior F1-score performance when compared to state-of-the-art multivariate time series classification models, while using fewer parameters.

4.5 Summary

In this chapter, a deep learning technique, specifically convolutional neural networks, was applied to the task of pianist identification using handcrafted features extracted from the symbolic representation of music. The proposed approach leveraged the ability of CNNs to learn a hierarchical representation of the data, thus enabling the capture of unique performer styles. The results of the study demonstrated the effectiveness of the proposed *convnet-beat* model in identifying performers from unseen musical excerpts. Visualization of the filters of the first channel for all three layers in the model revealed that the filters had learnt more complex patterns as they moved deeper into the layers. Furthermore, the use of class activation maps revealed that many discriminative regions had been identified in the first layer, supporting the use of beat-specific filters in the first layer to capture beat-level micro information. To address the issue of overfitting, commonly encountered in neural networks, particularly CNNs, several techniques such as batch normalization, Adaptive Average Pooling and dropout were employed. The model also exhibited superior performance when compared to state-of-the-art methods. These findings demonstrate the potential of the *convnet-beat* model for pianist identification and provide a foundation for the development of more advanced music-specific hierarchical model as presented in Chapter 6.

Chapter 5

Large Scale Dataset Construction

5.1 Introduction

In the field of expressive piano performance, data-driven approaches have been used to analyze and generate realistic and expressive renditions of music played on the piano. These approaches have gained popularity in recent years, with an increasing focus on data-hungry, deep learning techniques [48, 46]. However, in order to build comprehensive models and compare different approaches, large-scale datasets of expressive piano performances are required, and these datasets must adhere to certain standards. These datasets should include multiple performances of the same piece of music by multiple performers in order to capture expressive details and common performance idioms, and should allow for the study of expressiveness and styles across different performers. In the past, datasets with very limited numbers of pieces were recorded and organized by researchers [203, 204, 205, 70], but the non-trivial task of Composition Entity Resolution (CER), which involves aligning the complex naming schemes of classical music, has made it difficult to obtain multiple performances of the same music at a larger scale.

The representation of the data is also important. For instance, audio recordings provide the most accurate representation of a performance, but it is very challenging and labor-intensive to extract expressive features from the waveform [206], and often requires complex processing. MIDI files, on the other hand, can serve as a mid-level, piano-roll like representation of piano performances, but they may not be able to preserve as many fine-grained details as audio files. The repertoire and diversity of the dataset are also important considerations, as different datasets cover different time periods and styles of music. A symbolic score, such as a MusicXML file, is typically needed for tasks such as performance rendering [101], as it allows for the observation of expressive deviations by comparing with a quantized, dead-pan score. Furthermore, the size of the dataset is important, as large datasets are essential for training deep neural networks.

To address the need for a large-scale dataset of expressive piano performances, we propose a new performer-oriented dataset[1] called Automatically Transcribed Expressive Piano Performance (ATEPP) [2], which consists of 11742 virtuoso recordings with 1007 hours of music. This dataset is intended for studying expressiveness and styles across different performers in Western classical piano music. Unlike previous datasets, which were created by recording MIDI files from computer-controlled pianos, our dataset was created by applying state-of-the-art piano transcription models [207, 208] to audio recordings of performances. This allows for the inclusion of a wider range of performances and the exploration of performer-specific expressiveness and different schools of playing. In order to create our dataset, we first performed an error analysis of existing piano performance transcription models to verify the reliability of the transcribed performances in Section 5.2, followed by a listening test to evaluate the quality of the transcriptions in Section 5.2.4. Finally, we construct the dataset by overcoming the challenge of Composition Entity Resolution (CER) which is discussed in Section 5.3.1, followed by an audio matching and solo filtering pipeline.

Overall, the release of our dataset represents a significant step forward in the field of expressive piano performance, as it provides a dataset with sufficient richness and variety for studying expressiveness and styles across different performers. The inclusion of audio recordings, rather than MIDI files, allows for the exploration of performer-specific expressiveness and different schools of playing, and the dataset can be used for a range of tasks beyond just performance analysis and generation, such as performance attribute

[1] Released dataset and supplementary material (Appendix): `https://github.com/BetsyTang/ATEPP`. The dataset is made available under Creative Commons Attribution 4.0 International Public License (CC BY 4.0).

analysis [209, 210, 211], comparison of performances and styles [212], and performance visualization [213]. The ATEPP dataset is a valuable resource for researchers working in this field and has the potential to significantly advance the state-of-the-art in expressive piano performance research.

The compilation of the dataset was a joint endeavor in which I collaborated with two of my colleagues from the Center for Digital Music (C4DM), Huan Zhang and Jingjing Tang. Each party contributed equally to the construction of the dataset. My specific responsibilities included obtaining performance recordings from the internet, aligning the performances with the scores, calculating errors, and utilizing the transcribed data to train a pre-existing performance rendering model, as well as analyzing the rendered expressive performances to validate our transcribed data for musical expression analysis. The materials presented in this chapter have been utilized with the express consent of my aforementioned co-authors and this work has been presented in the *The 23rd International Society for Music Information Retrieval Conference (ISMIR 2022).*

The remaining portion of this chapter is organized as follows: We begin by thoroughly examining the reliability of the state-of-the-art transcription models through objective analysis and human listening tests in Section 5.2. Next, we delve into the techniques used to gather and refine our data, including audio matching and noise filtering, in Section 5.3. Finally, we present an overview of the key statistics of our dataset in the same section. The chapter concludes with a summary in Section 5.4.

5.2 Transcription Evaluation and Post-Processing

5.2.1 Automatic Piano Transcription

Automatic Music Transcription (AMT) is defined as the design of computational algorithms to convert acoustic music signals into some form of human-readable music notation. However, it is regarded as exceptionally challenging due to the fact that it necessitates the estimation of multiple subtasks, including (multi-)pitch estimation, onset and offset detection, instrument recognition, beat and rhythm tracking, interpretation of expressive timing and dynamics, and score typesetting [149]. Recent state-of-the-art AMT systems have achieved great precision with relatively low error rates

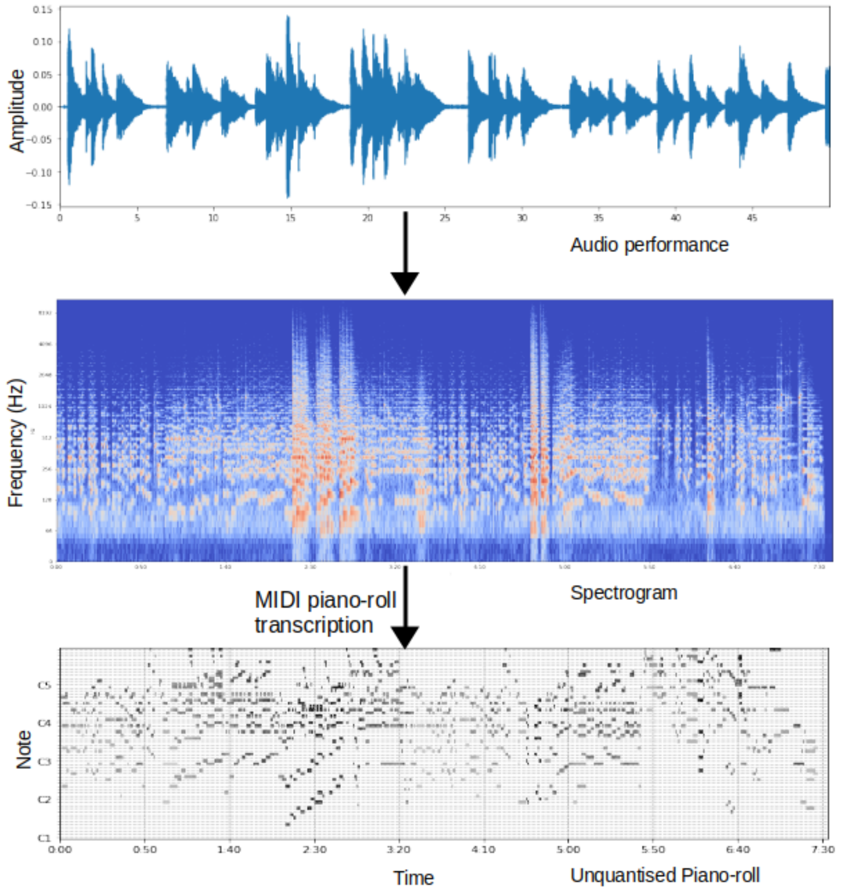


Figure 5.1: Data representation on an AMT system, with the input waveform on top, the internal time-frequency representation in the middle, and the unquantized piano-roll representation of the output at the bottom. The example corresponds to L.V. Beethoven's Piano Sonata No. 3. 2nd movement (taken from our dataset).

thanks to deep learning models. For example, the Onsets and Frames transcription model [214] that conditioned framewise note detection task on top of determined piano onsets, a full piano roll with velocity was transcribed. The High-resolution model developed by [207] improved the precision by regressing the exact timestamp of each note. Recently, generic encoder-decoder architecture [215] exploits language-like modelling that achieved model simplicity while retaining performance. As demonstrated in Figure 5.1, a typical AMT system uses the raw audio waveform, which is shown at the top of the figure, as its input. After that, it does an internal calculation

Model	HE	SN	MS	Other
High-Resolution [207]	3.2%	1.5%	1.2%	5.6%
OnsetsFrames [216]	4.2%	2.4%	0.1%	6.7%
Seq2Seq [215]	8.1%	2.9%	0.3%	7.9%

Table 5.1: Error produced by the three transcription models on the Mazurka dataset [2].

to determine the time-frequency representation, which is seen in the centre, and then it generates an unquantized representation of the pitches over time (also called a piano-roll representation, as shown at the bottom).

5.2.2 Common Transcription Errors

Despite having high accuracy by DNNs, a small amount of transcription errors would negatively affect the expressive performance style modelling. Consequently, we use three state-of-the-art transcription techniques [207, 216, 215] and test them on the MazurkaBL [217] dataset to ensure the reliability of the transcribed performances. To do this, the performed Mazurkas in the dataset are first transcribed using all three transcription models. Given that the MazurkaBL dataset has no ground truth data, we use their MusicXML score files and a symbolic music alignment algorithm (discussed in Section 3.3.1) proposed by [177] to align the performances with their corresponding score files, assuming the performances match the same score version. Finally, we compare the alignment result for each performance with their corresponding score and observe three prevalent types transcription errors:

- **Harmonic Errors** (HE): Incorrectly detecting notes that are harmonically related to other played notes. This occurs when the harmonic series of two notes overlap, leading to a missing or extra octave or fifth.
- **Segmented Notes** (SN): Notes that splits into two with a brief gap (<10ms) between offset and onset and might be caused by amplitude modulation [218].
- **Mis-touched Short notes** (MS): Random spurious short notes (<16ms) appear in the transcription.

In addition, we evaluate the presence of these errors in the Mazurka performances transcribed by each of the three transcription models [207, 216, 215], and compute the error rate for each error category. As shown in Table 5.1, the High-resolution model [207] creates the fewest overall errors but produces more short notes than the other transcription models.

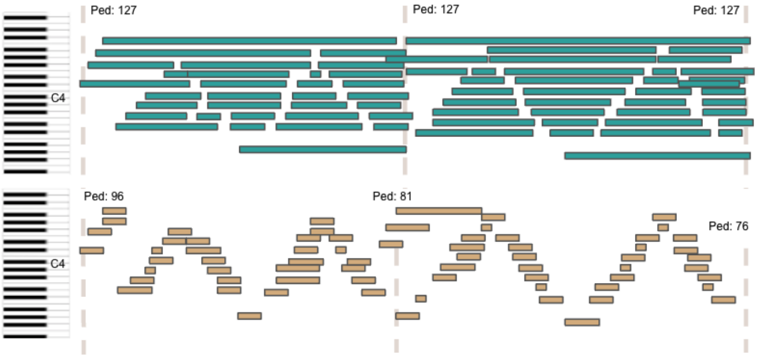


Figure 5.2: A comparison of the output pianorolls of the original High-Resolution model (top) and the joint note-pedal model (bottom). Dashed lines indicate pedal-on message with velocity. [2]

We further notice that the High-resolution [207] model stretches the duration of notes in order to simulate the sustain effect, which is normally achieved by the performer pressing a sustain pedal on a piano. However, in MIDI rendering software, such stretching does not result in a perceptible difference; nonetheless, correct end locations of each note are essential for piano performance analysis. To address this problem, we make some adjustments to the original High-Resolution model [207] by including a joint note-pedal training technique, which combines the 88 keys and 3 pedal channels of a piano into 91 prediction classes with velocity for each channel, and removes the extension of note offsets. This is dissimilar to their original training, where they trained separate networks for key activity and sustain pedal with binary velocity. By conditioning the sustain effect on both the key-down time and the pedal controls, the joint-note pedal training aims to provide more precise note offset.

The transcription comparison in Figure 5.2 between the original High-Resolution model and the modified joint-note pedal version demonstrates that the note-pedal transcription produces more realistic note offsets that are not prolonged with pedal, in addition to the velocity of sustain pedals.

It also illustrates the concept of modelling the pianist's key action rather than the note's damping time, which might be caused by the sustain pedal or key stroke. Evaluation results showed that the onset F1 score (tol = 50 ms) after 300k iterations was 92.1%, and onsets and offsets evaluation achieved 68.2%.

5.2.3 Score-to-Performance Alignment & Error Correction

Symbolic music alignment is a process of automatically matching a note in a music performance with a corresponding note in a score or a reference performance. We use a Hidden Markov Model (HMM) based symbolic music alignment algorithm proposed by Nakamura et al. [177]. Although, the algorithm achieved high accuracy, it could not outperform the current state-of-the-art methods available in the literature [174, 175, 176]. These models, on the other hand could only achieve great accuracy when applied to their own data. The HMM-based algorithm was applied to those data as well as their own data to evaluate its accuracy and computational efficiency. The HMM-based algorithm obtained the highest accuracy accross all datasets and the computational efficiency surpasses all the other algorithms.

To identify the errors with reference to score, we aligned the transcribed recordings with their corresponding score using the alignment algorithm. We corrected the transcription errors resulted from the alignment using simple rules such as, **1)** extra notes in the transcription but not in the score are deleted, **2)** mismatched pitches are corrected to the pitch written in the score, and **3)** missing notes are interpolated and written back to MIDI using the following rule:

$$N(t) = N(t - \Delta_p) + (N(t + \Delta_n) - N(t - \Delta_p))\frac{\Delta_p}{\Delta_p + \Delta_n} \tag{5.1}$$

where $N(t)$ denotes the onset or offset timestamp of the missing note at beat t, and Δ_p, Δ_n represent the beat distances between the missing note and the previous or next existing notes, respectively.

5.2.4 Listening Evaluation

To assess the perceptual quality of the transcribed piano performances, we conducted a subjective listening test in which participants were asked to rate the similarity of transcribed MIDI files to a reference recording. We compared the ground truth recording with four different transcribed MIDI renderings produced by four different transcription systems: the original High-Resolution system [207] (**C1**), a joint note-pedal model (as described in Section 5.2.2) (**C2**), a score-corrected version (**C3**) of the joint note-pedal model, and a language-model transcription system [215] (**C4**). Each MIDI performance is rendered on a KAWAI CA49 electric piano and recorded with a Zoom H4n Pro Recorder followed by a basic noise reduction using Audacity.

The test was carried out using the MUSHRA protocol [219], and it consisted of five 20-second classical piano excerpts with varying styles (Q1-Liszt, Q2-Debussy, Q3-Bach, Q4-Rachmaninov, Q5-Mozart). Each style had five recordings, including one reference and four transcribed stimuli. Participants were then asked to compare the transcribed MIDI renderings with the reference recording and rate their similarity on a 100-point scale. They were also asked to ignore the timbral or acoustic differences, but to base their judgements on the expressive differences between the stimuli, such as dynamics and timing. We received a total of 1075 ratings from 43 participants, with half of them having more than five years of piano playing experience.

	Q1	Q2	Q3	Q4	Q5	Overall
Reference	4.42±0.24	4.17±0.29	4.24±0.31	4.28±0.31	4.46±0.27	4.30±0.12
C1	**4.12±0.38**	3.52±0.37	3.88±0.32	3.60±0.41	3.88±0.4	3.81±0.16
C2	3.83±0.42	**3.86±0.39**	**4.28±0.31**	**3.97±0.42**	**4.06±0.45**	**4.01±0.17**
C3	3.44±0.37	2.96±0.36	3.44±0.35	3.32 ±0.42	3.76±0.41	3.38±0.17
C4	3.88±0.34	2.32±0.47	3.60±0.29	3.84±0.33	3.68±0.37	3.46±0.18

Table 5.2: Listening test results interms of mean opinion scores (MOS) [2].

Table 5.2 presents the mean opinion scores converted from the original ratings to a 5-point scale. Evidently, all stimulus groups varied considerably from the reference, and there was a clear preference for the joint note-pedal model (C2), whereas the score-corrected transcription (C3) and language-model (C4) transcriptions were rated much lower. Based on these results, the joint note-pedal model (C2) was used for transcribing our dataset. The results also revealed a perceptual difference in the transcription quality across music styles, indicating a bias in transcription models towards transcribing fast, arpeggio-heavy passages being rated as lower perceptual quality. On the other hand, good transcriptions sound considerably more like the reference when the texture is slow and sparse.

5.3 Dataset Overview

5.3.1 Data Acquisition & Refinement

Our data collection pipeline, illustrated in Figure 5.3, initiated with 49 famous pianists. We used the Spotify API[2] to obtain metadata from their discographies, encompassing details such as composer, performer, album, title, and track duration. We then construct a composition-movement hierarchy by removing non-solo keywords like concerto or trio. Secondly, given the heterogeneity of classical music naming conventions, the next challenge was Composition Entity Resolution (CER) [220], defined as identifying tracks that correspond to the same piece of music.

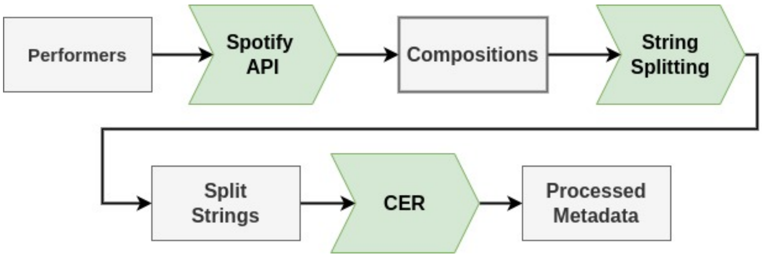


Figure 5.3: Data acquisition pipeline.

We take several steps to deal with CER which includes: **1**) compiling a lexicon of words for interchangeable phrases like *Prelude* ↔ *Praeludium*, **2)** extracting unique information like key and catalogue number (Opus. BWV, K. D., etc.) from the title string, **3)** matching composition title and movement title by computing the similarity score using normalized Levenshtein distance [221]. However, it is also important to remember that this approach of string matching is not always reliable since many songs in our discography have generic names like “Piano Sonata”. Thus, Algorithm 1 describes our three-step approach to composition entity resolution, where inputs are composition title C, movement title M, and duration D. Finally, we download each track from YouTube music based on the refined metadata and validate using the same CER algorithm.

[2] `https://developer.spotify.com/documentation/web-api/`

Algorithm 1 Composition Entity Resolution

```
# UniqueInfo extracts canonical key and composer-specific catalogue number.
for $k_1, k_2$ in $\mathit{UniqueInfo}(C_1, C_2)$ do
  if $k_1 \neq k_2$ then
    return False
  end if
end for
$S_c \leftarrow 1 - (\mathit{Levenshtein}(C_1, C_2)) / \max(|C_1|, |C_2|)$
$S_m \leftarrow 1 - (\mathit{Levenshtein}(M_1, M_2)) / \max(|M_1|, |M_2|)$
$S_d \leftarrow \frac{\mathrm{abs}(D_1 - D_2)}{\max(D_1, D_2)}$
$S \leftarrow \frac{S_c + S_m}{2} - S_d$
return $S \geq 0.6$
```

5.3.2 Audio Verification

The CER algorithm described above provided us with grouped performance audios by different compositions (or movements). Despite it's high accuracy of gathering performances interpreting the same pieces, a small portion of performances were erroneously identified and assigned to a non-matched movement. Therefore, to address this problem, we apply Chen et al. [222]'s cover song detection algorithm within each group of performance to compare each performance to a reference one. To determine the degree of similarity between the reference and the performance, we initially extract the Harmonic Pitch Class Profile (HPCP) [223] from both the audio signals. We then calculate the similarity between the tracks using the Q_{max} measure [224], which takes into account the maximum value of a cumulative matrix derived from the HPCP descriptors. If the similarity is greater than 0.9, as defined by Eq. 5.2, we confidently include the track in our dataset.

$$Sim = 1 - Q_{max}, \quad 0 < Q_{max} < 1 \tag{5.2}$$

5.3.3 Noise Filtering

In order to create a dataset of high-quality solo piano recordings, we needed to filter out any sounds that might not be part of a solo piano performance. This includes extraneous sounds such as applause or speech from live recordings, which would be transcribed as random pitch in the MIDI file. To accomplish this, we trained a deep learning model based on the Musicnn architecture [225] to identify and remove non-solo-piano segments from the audio. The model was trained using a binary classification approach, with a subset of AudioSet [226] containing various environmental sounds as negative examples and solo piano recordings as positive examples. Once the model had been trained, it was able to predict the probability that a 1-second segment of audio contained non-piano sounds. Using this information, we were able to apply a post-processing step to the audio tracks, which searched for and removed the longest continuous stretches of non-solo audio from the beginning and end of each track. Out of the 11742 tracks in our dataset, 567 of them were found to contain such segments and were successfully cleaned. Finally, we conducted a manual verification process to ensure the accuracy of the audio splicing.

5.3.4 MusicSML Score Collection

In addition to collecting audio recordings of musical performances, we also gather the corresponding musical scores in MusicXML that corresponds to our performance data. These scores were obtained from two different sources: 228 files were collected from the ASAP dataset [227], and an additional 90 files were retrieved from the MuseScore[3] online library. The MuseScore online library is a collection of scores that have been created and shared by users of the MuseScore software. In total, we collected 319 movements, which correspond to 5124 tracks in our dataset (43% of all tracks). To ensure that the scores and performances match up correctly, we used an automated process to determine the score-performance correspondence based on name matching, followed by a manual correction step to address any discrepancies.

5.3.5 Dataset Statistics and Content

The dataset represents a comprehensive and diverse collection of solo piano music, with 11742 tracks comprising 1580 movements in total. These tracks exhibit a low degree (0.2%) of overlap with the GiantMidiPiano dataset [228], making our dataset a valuable and unique resource for music research.

[3] `https://musescore.com/sheetmusic`

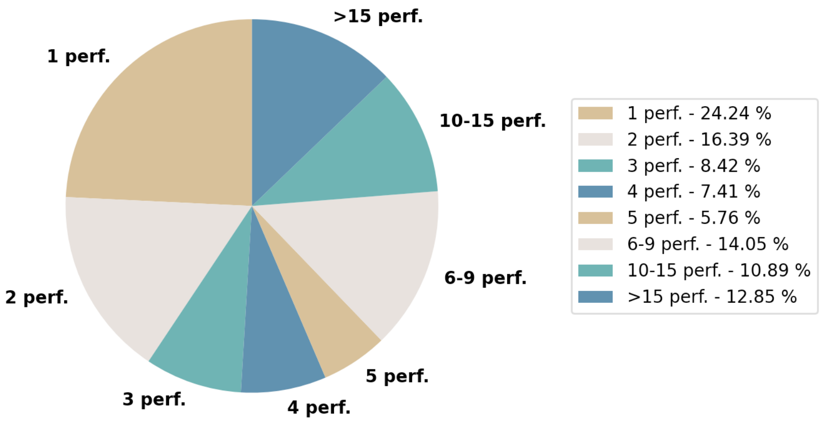


Figure 5.4: Distribution of movements by number of performances. E.g. 12% of our data have more than 15 performances [2].

The distribution of movement-performance occurrences within the dataset is depicted in Figure 5.4, with 44% of the 1580 movements having more than 5 recorded performances. This allows us to investigate a diverse range of interpretations for the same piece of music.

In addition, we show the distribution of the top 25 pianists' performances in our dataset, as shown in Figure 5.5, with Sviatoslav Richter with the most contributions to the dataset. In terms of composers, the our dataset includes solo piano works from 25 Western classical composers, covering a wide range of time periods from the Baroque to the Modern era.

To gain insights into the expressive nature of the performances in our dataset, we made a rough estimate of the deviations present by calculating the standard deviation of note velocities (σ_{vel}) and the standard deviation of inter-onset-intervals (σ_{ioi}) for each piece. These deviations are depicted in a box plot and compared with those of other datasets containing piano MIDI data [229, 230, 231, 228] in Figure 5.6. For the purposes of this analysis, the velocity values were normalized from the range [0, 127] to [0, 1], and the inter-onset-intervals were expressed in seconds. This analysis provides a useful overview of the expressive characteristics of the performances in our dataset.

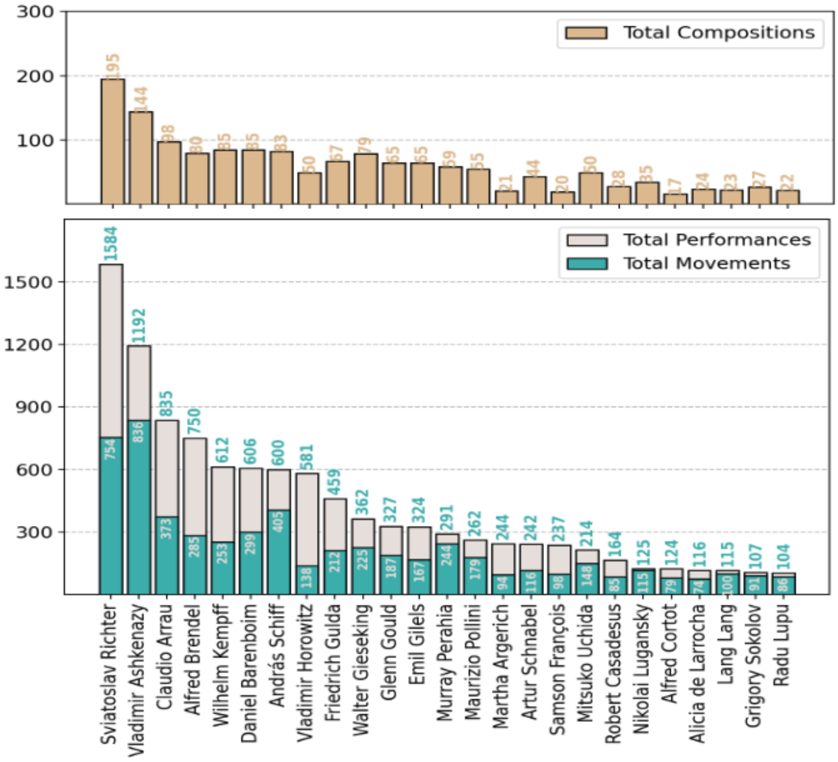


Figure 5.5: Statistics of the top 25 pianists in-terms of their performances in our dataset [2].

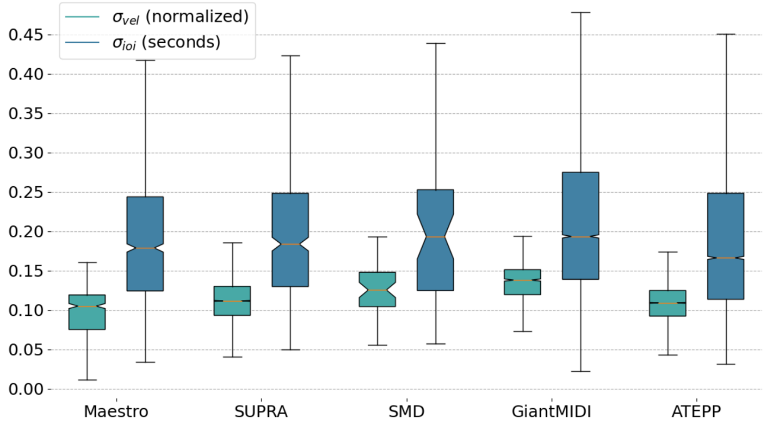


Figure 5.6: Note velocity deviation and IOI deviation for 5 datasets [2].

5.4 Summary

In this Chapter, we introduced an Automatically Transcribed Expressive Piano Performance (ATEPP) dataset, a comprehensive collection of 11742 virtuoso piano recordings totaling 1007 hours of music. The dataset was created using state-of-the-art piano transcription models applied to audio recordings of performances, rather than MIDI files recorded from computer-controlled pianos as in previous datasets. This allows for the inclusion of a wider range of performances and the exploration of performer-specific expressiveness and different schools of playing. To ensure the reliability of the transcribed performances, we conducted an error analysis and listening test of existing transcription models in Section 5.2. We also addressed the challenge of Composition Entity Resolution (CER) in constructing the dataset and implemented an audio matching and solo filtering pipeline in Section 5.3. The dataset is a valuable resource for researchers studying expressiveness and styles in Western classical piano music, and can be used for a variety of tasks including performance feature analysis, comparison of performances and styles, stylistic performance generation and performance visualization.

Chapter 6

Hierarchical Performance Modelling for Pianist Identification

6.1 Introduction

Previous studies in the literature have established a relationship between the structure of a piece of music and its performance characteristics, with certain structural elements influencing expression and emotion in the performance [180, 181]. For example, decrease in tempo and dynamics are commonly used to mark the boundaries of phrases [232], and the degree of slowing at these boundaries can indicate the importance of the phrase within the overall structure of the music [64, 65]. Furthermore, it has been observed that the most pronounced expressive differences between notated and performed interpretations tend to occur at lower levels of phrase structure [181, 13], while expressive timing and loudness exhibit a strong relationship at intermediate phrase levels [39, 10]. Additionally, metrical structure plays a significant role in shaping the expressiveness of a performance, particularly through the manipulation of accentuation through variations in duration and timing [232].

Computational models allow us to investigate these relationships between musical structures, such as phrase structure, and performance elements like timing and dynamics in human performance. However, there has been limited research on how to capture these relationships with mathematical descriptors, which could be used to identify the unique style of individual performers and build automatic performer identification systems. This is primarily because (a) all studies are limited to small dataset containing a small selection of pieces or performers, (b) characterising expressive performance requires a sophisticated feature extraction method, and (c) the information provided by the expressive performance model is not yet fully understood for the task of performance based performer identification. In addition, majority of the previous studies make use of features that only captures the tiny local context of the music, resulting in models that are unable to account for the long temporal relationship, which is essential for understanding the global aspects of performance expression.

To address these limitations, we (i) construct a substantial subset of the ATEPP dataset (see Chapter 6) with multiple performances of the same piece by different performers (ii) investigate the individual performance style by analysing different expressive performance features that characterizes a performer, and (iii) model the expressive features using a Recurrent Neural Network (RNN) based hierarchical approach for the task of automatic performer identification. To the best of our knowledge, the majority of previous works attempted performer identification using traditional statistical models or machine learning algorithms. However, deep learning models, in particular RNNs, have never been used to distinguish pianists, despite their demonstrated ability in modelling sequence data representing features or characteristics of musical expression for various MIR applications [101, 53, 233].

In order to evaluate the hypothesis that the hierarchical representation of Western classical music can improve representation accuracy, we propose the Hierarchical Performer Identifier (*HIPI*) model. This model utilizes a beat and measure level hierarchical structure, inspired by the work of Yang et al. [234] in language modeling. The *HIPI* model consists of three encoders; a note encoder, a beat encoder, and a measure encoder, all based on Bi-LSTM architecture and incorporating beat and measure level attention mechanism [235, 236]. The idea is to construct a beat level representation by summarising note representations using a beat level attention mechanism and then create a measure level representation by summarising beat-level representations using a measure level attention mechanism. This approach enables the model to learn information from the very granular to a higher-level perspective of musical performances, resulting in a comprehensive representation of the performer's playing style. The input to the model is comprised of note-level features calculated from the deviations of each performance from a reference score, and the output is the prediction of the most probable pianist.

The remainder of the chapter is organized as follows: Section 6.2 presents a performer-oriented dataset, derived from the ATEPP dataset [2], containing 6 performers playing the same compositions. In Section 6.3, the data preprocessing techniques are described, including the alignment of the score and actual performance, the extraction of score deviation features (as listed in Table 3.2), the hierarchical position encoding of each note level feature, and the formation of the input representation. The architecture of the proposed hierarchical model and the pianist classification process are also detailed in the same section. Section 6.4 outlines the baseline models, the experimental setup, and the case studies for segment-wise and piece-wise pianist classification. The results are analyzed and discussed comprehensively in Section 6.5. Finally, in Section 6.6, the chapter is summarized, highlighting the contributions and offering potential avenues for future research.

6.2 Dataset

Data-driven approaches for performers' style analysis need large corpora of music performances to derive expressive performance parameters. One possible reason for Deep Neural Networks (DNNs) not being used for performer identification is the lack of large-scale datasets with overlapping performances by different performers. In addition, constructing such a large-scale dataset is not a trivial job since the dataset must comprise both the recorded performances (as audio or MIDI files) and their corresponding scores. Besides, the scores and performances need to be aligned in such a way that we can obtain a mapping between elements in the performance (temporal position for audio recordings or pitch, onset and offset times for MIDI) and the elements in the corresponding score (a mapping between a performed MIDI note and a note in the score). These details are essential for calculating expressive features, for example, expressive timing, which may be conceptualised as the amount by which onset and offsets deviate from the times shown in the score.

In this section, a subset of the ATEPP dataset (detailed in Chapter 5)

Dataset	Compositions	Movements	Performances	Composers	Performers
Vienna 4x22 [[237]]	4	4	88	4	22
Schubert 4x9 [[238]]	1	4	36	1	9
Repp [[88]]	4	4	120	4	10
Proposed Dataset	19	35	474	2	6

Table 6.1: An overview of major symbolic piano datasets with multiple performances of the same piece by different performers, compared to our proposed dataset.

Composer	Composition	Performers
Beethoven	Piano Sonata No. 3 in C Major, Op. 2 No. 3	Alfred Brendel, Claudio Arrau, Daniel Barenboim, Friedrich Gulda, Sviatoslav Richter, Wilhelm Kempff.
	Piano Sonata No. 7 in D Major, Op. 10 No. 3	
	Piano Sonata No. 8 in C Minor, Op. 13 Pathetique	
	Piano Sonata No. 9 in E Major, Op. 14, No. 1	
	Piano Sonata No. 10 in G Major, Op. 14 No. 2	
	Piano Sonata No. 17 in D Minor, Op. 31 No. 2 Tempest	
	Piano Sonata No. 18 in E-Flat Major, Op. 31 No. 3 The Hunt	
	Piano Sonata No. 19 in G Minor, Op. 49, No. 1	
	Piano Sonata No. 20 in G Major, Op. 49 No. 2	
	Piano Sonata No. 22 in F Major, Op. 54	
	Piano Sonata No. 23 in F Minor, Op. 57 Appassionata	
	Piano Sonata No. 27 in E Minor, Op. 90	
	Piano Sonata No. 28 in A Major, Op. 101	
	Piano Sonata No. 30 in E Major, Op. 109	
	Piano Sonata No. 31 in A-Flat Major, Op. 110	
	Piano Sonata No. 32 in C Minor, Op. 111	
	Sonata No. 12 In A Flat Op. 26	
Mozart	Fantasia in C Minor, K. 475	
	Piano Sonata No.8 in A minor, K.310	

Table 6.2: An overview of the composers, compositions, and performers in our proposed dataset.

is introduced, consisting of multiple performances of a single composition by different performers. This subset is based on the premise that datasets containing multiple performances of a single piece by various performers can be utilized to model the similarities and differences among performers. Our dataset[1] comprises of 6 virtuoso pianists, each with 79 performances, for a total of 474 Western classical piano recordings in MIDI format as presented in Table 6.1. Notably, our dataset contains multiple performances of the same piece by each performer. The dataset was obtained by filtering out compositions with at least one distinct performance by each performer. This resulted in a corpus of 19 compositions and 35 movements from composers Ludwig van Beethoven and Wolfgang Amadeus Mozart as shown in Table 6.2.

To maintain a balanced dataset, the minimum number of performances among all the performers per composition was calculated and any excess

[1] Released dataset: `https://doi.org/10.5281/zenodo.7222768`. The dataset is made available under Creative Commons Attribution 4.0 International Public License (CC BY 4.0).

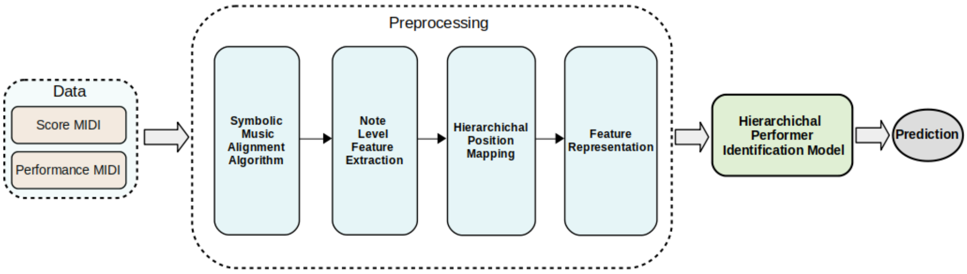

Data
Score MIDI
Performance MIDI
Preprocessing
Symbolic Music Alignment Algorithm
Note Level Feature Extraction
Hierarchichal Position Mapping
Feature Representation
Hierarchichal Performer Identification Model
Prediction

Figure 6.1: Schematic overview of the hierarchical performer identification model with all the main steps of its workflow.

performances were disregarded (using the alignment results, errors such as extra notes, mismatched pitch and missing notes were calculated and performances with higher error rate were disregarded). This approach also eliminates any potential performer-composer correlation bias, as all performers performed the same compositions by both composers. However, the limitation lies in its focus on only two composers, restricting the diversity of musical styles in the data and potentially not fully representing variation in pianist styles. We use MIDI since extracting expressive features from raw audio requires complex processing, whereas MIDI can be seen as a mid-level, piano-roll-like representation.

6.3 Methodology

As outlined in Figure 6.1, we design this section with respect to the following steps: a note-level alignment of score and the real performances, extraction of meaningful expressive performance parameters, a hierarchical position mapping where we find the beat and measure position of each note from the musicXML score file and create a mapping with the corresponding performed note, creation of a meaningful input representation for our hierarchical model and finally a comprehensive description of our hierarchical performer identification model.

6.3.1 Data Pre-processing

6.3.1.1 Alignment & Feature Extraction

The concept of expressive performance can be understood as the deviation from the notated score, which is a mechanical representation of a piece of music in terms of tempo, dynamics, and articulation. Performers deviate from the score for two main reasons: it can be challenging to perform the music exactly as written, and these deviations can enhance the dramatic, affective, and artistic qualities of the performance, which can emotionally connect with listeners. Thus, quantifying these deviations through expressive parameters can characterize the unique style of each performer. Previous research, such as the study by Stamatatos [182], has demonstrated the use of score deviation features to successfully discriminate between performances by skilled pianists playing the same piece.

Since the most important expressive parameters available to a performer are tempo, dynamics and articulation and the micro variations happen while playing each note, it is essential to perform a score to performance alignment to have a note level mapping. Consequently, we use the algorithm for symbolic music alignment proposed by [177] to align the notes in the score MIDI and the notes in the corresponding performance MIDI. This enables us to compute note level deviations and quantify them using a similar method presented in Section 3.3.2 that encompasses a set of note level performance parameters. The proposed note-level performance features (See Table 3.2) include onset time (exact time when a note is performed) deviation, Inter Onset Interval (time interval between the onsets of two notes of the same voice) deviation, offset to onset time interval(time interval between the offset of a note and the onset of the next note of the same voice) deviation, dynamic level (the loudness of a note) deviation and note duration deviation. For the rest of the article, we'll abbreviate as follows: OT = Onset Time; IOI = Inter Onset Interval; OTD = Off Time Duration; DL = Dynamic Level; ND = Note Duration.

6.3.1.2 Hierarchical Position Mapping

Western classical music has a hierarchical nature in its structure and modelling real performances using hierarchical approach may lead to a better representation of performers' individual style. However, real recorded performances in MIDI format do not always contain the hierarchical information encoded in them. These details can only be found in the actual composition, which is written out as either sheet music or transcribed as musicXML format. To overcome this limitation, we first make a matching between the notes in the musicXML file and their corresponding notes in the synthesized score MIDI which provides a mapping of the beat and measure positions for each matched pairs. Finally, we create a mapping of these matched pairs with the aligned pairs in the alignment result file generated from the MIDI-to-MIDI alignment (as discussed earlier in this section) that eventually gives us the mappings of beat and measure positions of the real performance.

6.3.1.3 Feature Representation

One of the most challenging aspects of using neural networks on music data is determining how to represent the musical content as input. A widely used input format is a piano-roll representation, which represents music as a time-pitch 2D matrix with the columns representing the time steps and the rows representing the pitches [239, 240]. Despite its popularity, piano-roll representation has its drawbacks; for instance, it is difficult to tell the difference between a long note and a repeated short note since no note-off information is available. Since these details are important for quantifying expressive performance, we follow the representation technique presented in [241, 242, 243], where music data is modelled as a 1D input sequence by ordering note events with its time position and pitch. In addition, we make a small modification by stacking each handcrafted note level deviation features together and transforming them into a multidimensional input sequence as depicted in Figure 4.1.

6.3.2 Hierarchical Performer Identifier (HIPI)

Our proposed model is based on Hierarchical attention network [234], a type of stacked RNN model designed with the goal of modeling sequential data (texts, music, video streams etc.) that has some sort of hierarchical structures. Recent studies demonstrated that better performance can be achieved using hierarchical approach in RNN models [234, 244]. Given that Western classical music has a hierarchical nature in its structure (note, beat, measure, phrase etc.), we model the expressive performance using an LSTM-based

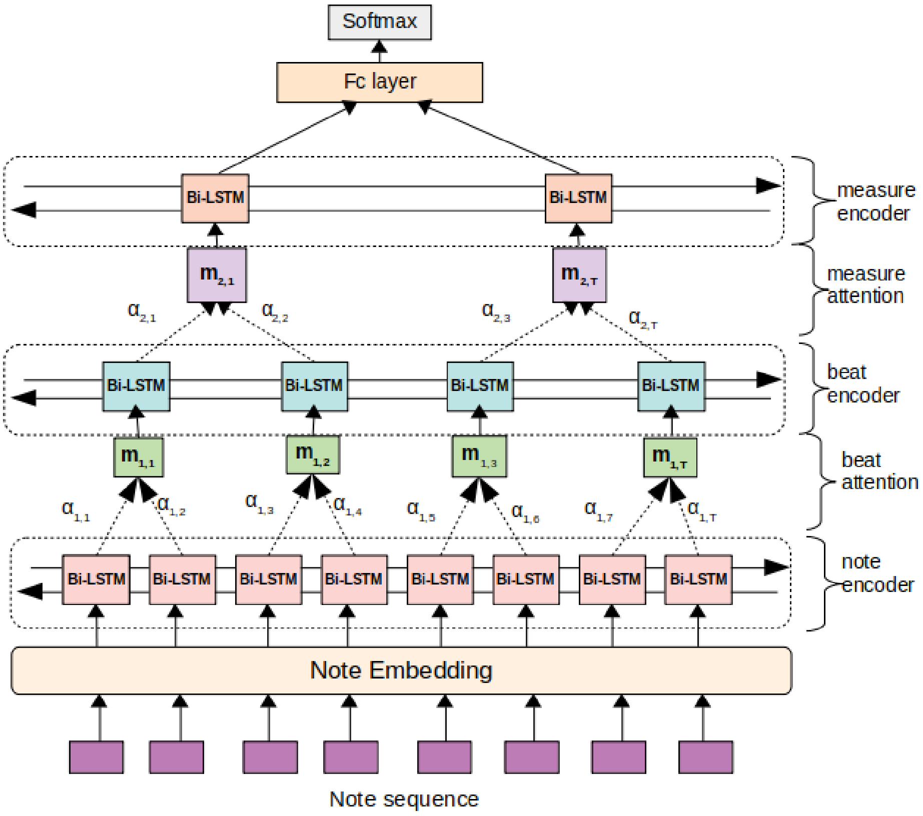


Figure 6.2: An overview of the Hierarchical Performer Identifier architecture for pianist classification.

hierarchical attention network and try to predict the most likely performer from a sequence of performance parameters.

The proposed system, depicted in Figure 6.2, incorporates a hierarchical structure composed of three levels: note, beat, and measure. Each level is encoded by a Bi-directional Long Short-Term Memory (Bi-LSTM) network with varying hidden nodes and layers. The system also includes beat and measure level attention mechanisms that summarize lower-level representations at each hierarchical boundary by computing a weighted sum, as described by Yang et al. [234]. We modify the context attention mechanism proposed by Yang et al. [234] to incorporate multi-head attention, as proposed by Vaswani et al. [245]. This involves splitting the input dimension into several heads, each with its own set of weights, allowing each attention head to focus on different types of notes. The Bi-LSTM architecture was chosen due to the context-dependency inherent in music and the need for sequential information in both forward and backward directions. For the purposes of this study, we assume that a piece consists of L measures, M_i and each measure contains B_i beats and each beat contains N notes. Thus, n_{it} with $t \in [1, N]$ represents the notes in the i^{th} beat and b_{it} with $t \in [1, B]$ represents the beats in the i^{th} measure.

6.3.2.1 Note Encoder

Initially, the note input sequence n_{it}, t ∈ [1, N] is fed into a dense layer that serves as an embedding layer where the input sequence gets multiplied by an embedding matrix, N_e generating an embedding vector, $x_{it} = N_e\ n_{it}$. The Bi-LSTM receives the embedded vector as input and generates the hidden states by summarising information from both directions for notes, thereby including contextual information into the hidden states. The Bi-LSTM is comprised of two LSTMs: a forward LSTM $\overrightarrow{f}$ that reads the notes from n_{i1} to n_{iN} and a backward LSTM $\overleftarrow{b}$ that reads the notes from n_{iT} to n_{i1}. This can be formulated as below:

$$x_{it} = N_e n_{it}, t \in [1, N], \tag{6.1}$$

$$\overrightarrow{h_{it}} = \overrightarrow{LSTM}(x_{it}), t \in [1, N], \tag{6.2}$$

$$\overleftarrow{h_{it}} = \overleftarrow{LSTM}(x_{it}), t \in [N, 1]. \tag{6.3}$$

The forward hidden state, $\overrightarrow{h_{it}}$ and the backward hidden state, $\overleftarrow{h_{it}}$ are concatenated to provide a single hidden state, $h_{it} = [\overrightarrow{h_{it}}, \overleftarrow{h_{it}}]$, of a given note n_{it}, which summarises the whole note sequence centred around note n_{it}.

6.3.2.2 Beat Attention

We introduce a beat level attention mechanism with the notion that not all note level features are equally important for the representation of performer's style. The attention mechanism pays special attention to the important local aspects of the performances. To compose a beat level node, we use the beat position for each note that we obtain by parsing the musicXML score file. Notes that belong to the same beat are indexed with t ∈ $[n_f, n_l]$, where n_f and n_l denote the index of the first and last notes in the selected beat boundary. The lower level hidden states, h_{it}, produced by the note encoder for sequence t in each beat boundary are summarized by the context attention to compose a beat-level node $\mathbf{m}_i$.

The attention mechanism is a single feed-forward neural network that takes the hidden states, h_{it}, as input and applies a softmax function to get the attention weights, α_{it}, and produce a context vector as a weighted sum of the hidden states based on the weights. The attention mechanism can be formulated as below:

$$\begin{aligned} \mathbf{u}_{it} &= tanh(W_n h_{it} + b_n), \\ \boldsymbol{\alpha}_{it} &= \frac{\exp(u_{it}^\top u_n)}{\sum_t \exp(u_{it}^\top u_n)}, \\ \mathbf{m}_i &= \sum_t \alpha_{it} h_{it}, \end{aligned} \tag{6.4}$$

Where $\mathbf{m}_i$ is the beat level node summarized through the attention and W_n, b_n and u_n are the parameters learned by the attention model after random initialisation.

6.3.2.3 Beat Encoder

Given the beat level nodes, b_i, that summarises the notes to a beat level representation, the beat encoder takes them as input and outputs the hidden states, h_{it}. Similar to the note encoder, the Bi-LSTM encodes the beats by using the forward and the backward LSTMs. This is formulated as below:

$$\overrightarrow{h_{it}} = \overrightarrow{LSTM}(b_{it}), t \in [1, B], \tag{6.5}$$

$$\overleftarrow{h_{it}} = \overleftarrow{LSTM}(b_{it}), t \in [B, 1]. \tag{6.6}$$

The hidden state of a given beat, b_{it} is obtained by concatenating the forward hidden state, $\overrightarrow{h_{it}}$, with the backward hidden state, $\overleftarrow{h_{it}}$, yielding the single hidden state, $h_{it} = [\overrightarrow{h_{it}}, \overleftarrow{h_{it}}]$, which summarises the whole beat sequence centred around beat b_{it}.

6.3.2.4 Measure Attention

The purpose of adding measure level attention is to account for long temporal dependencies, which are crucial to understanding global aspects of performance expression. This is done by summarising the beat level information into a measure level node using a measure level attention mechanism. To create a measure level node, we use the measure position obtained by parsing the musicXML score file for each beat. Beats belonging to the same measure are indexed with t $\in [b_f, b_l]$, where b_f and b_l are the indexes of the first and final beats in the given measure boundary. The lower level hidden states, h_{it}, produced by the beat encoder for sequence t in each measure boundary are summarized by the context attention to compose a measure-level node m_i.

$$\begin{aligned}
\mathbf{u}_{it} &= tanh(W_b h_{it} + b_b), \\
\boldsymbol{\alpha}_{it} &= \frac{\exp(u_{it}^{\top} u_b)}{\sum_t \exp(u_{it}^{\top} u_b)}, \\
\mathbf{m}_i &= \sum_t \alpha_{it} h_{it},
\end{aligned} \tag{6.7}$$

Where m_i is the measure level node summarized through the attention and W_b, b_b and u_b are the parameters learned by the attention model after random initialisation. This is similar to beat attention as illustrated in Equation 6.4.

6.3.2.5 Measure Encoder

Given the measure level nodes m_i, the measure encoder takes them as input to the Bi-LSTM network and generates the piece level representation. The forward and backward LSTM encodes the measure level information and produce the hidden states in both direction as follows:

$$\overrightarrow{h_{it}} = \overrightarrow{LSTM}(m_{it}), t \in [1, M], \tag{6.8}$$

$$\overleftarrow{h_{it}} = \overleftarrow{LSTM}(m_{it}), t \in [M, 1]. \tag{6.9}$$

Concatenating the forward hidden state, $\overrightarrow{h_{it}}$, with the backward hidden state, $\overleftarrow{h_{it}}$, yields the single hidden state, $h_{it} = [\overrightarrow{h_{it}}, \overleftarrow{h_{it}}]$, which summarises the whole measure sequence centred around the measure m_{it}.

6.3.3 Pianist Classification

The hidden states, denoted as h_{it}, produced by the measure encoder are input into a fully-connected layer to obtain the piece-level representation vector, denoted as $\mathbf{q}$, which encapsulates a comprehensive high-level view of the performance and may be utilized as a criterion for the performer classification task. The outputs of the fully-connected layer are then processed through a softmax activation function, which calculates the posterior probabilities over all possible performer classes. The performer classification can be expressed mathematically as follows:

$$\mathbf{P} = softmax(W_c\mathbf{q} + b_c). \tag{6.10}$$

Since, this is a multi-class classification problem, where the goal is to train our algorithm to predict one of several possible outcomes, we use categorical cross-entropy to calculate the loss between the true label and the predicted label. The loss function is formulated as follows:

$$L_{CE} = -\sum_{i=1}^{C} y_i log(\hat{y_i}), \tag{6.11}$$

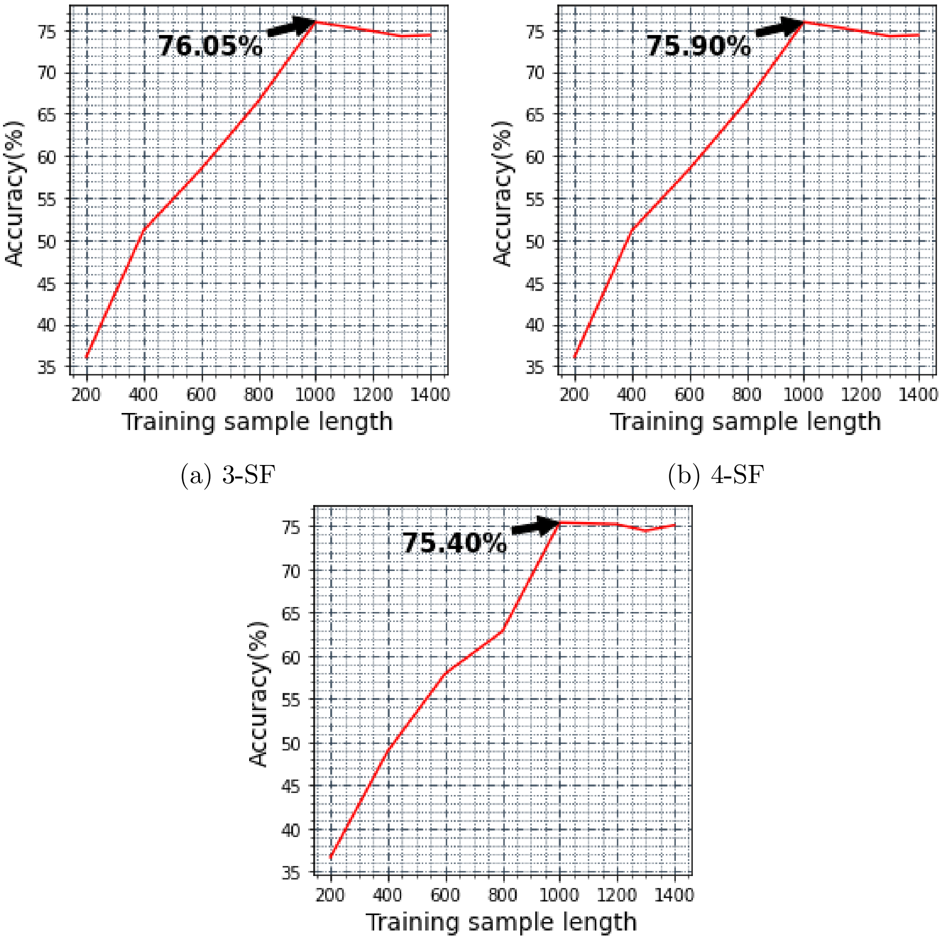
where y_i is the true class label and $\hat{y_i}$ is the predicted class label.

6.4 Experiments

In this section, we first discuss several baseline models that are used to evaluate the performance of our proposed model. Secondly, we describe the model configuration and the training method of our proposed model. Finally, we provide two experiments on identifying virtuoso pianists based on their playing style, one in which we train our model using segmented piece and attempt to classify performer based on each segment, and another in which we train and classify each performer considering the full piece of music.

6.4.1 Baseline Models

We compare our Hierarchical Performer Identifier (*HIPI*) model with the following baseline models:



(a) 3-SF

(b) 4-SF

(c) 5-SF

Figure 6.3: Training example length in terms of number of notes vs. classification accuracy for different feature combination.

1. **Bi-LSTM**: This model is a baseline single layer bidirectional LSTM [246] model with a hidden size of 64 and dropout 0.5. The output from the Bi-LSTM is fed into a fully connected layer and a softmax function is applied for the classification. The model has 167814 trainable parameters.

2. **Transformer**: With state-of-the-art performance shown by transformer models across a range of time series forecasting problems [247, 248, 249], we use the transformer encoder model introduced by [245] as a performer classification model. This model consists of

two encoder layer made up of 2048-node feed-forward neural network with an 8-head attention mechanism and dropout 0.2. The model has 562950 trainable parameters.

3. **Hierarchical Network (HN)**: The purpose of this model is to do an ablation study, which aims to investigate the benefits of the attention mechanism. This is very similar to *HIPI* but consists of 3 stacked Bi-LSTM layers without any attention mechanism. This model has 498182 trainable parameters.

4. **HIPI-B**: This is a variant of our proposed model with only beat level hierarchy. This model also serves as an ablation study to observe the importance of modelling performances in different hierarchical level for the task of performer identification. The model comprises a note encoder, a note level attention, a beat encoder, and a beat level attention, with each encoder including two layers of 64-node Bi-LSTM. This model contains 183558 trainable parameters.

5. **HIPI-VAE**: Although a typical Variational Auto-Encoder consists of an encoder, q$_\theta(z|x)$, a decoder p$_\phi(x|z)$ and a latent variable z, only the encoder part is stacked on top of the *HIPI* model. The goal of the VAE is to find a distribution q$_\theta(z|x)$ of some latent variables which can be sampled from a known prior distribution p(z). It is also known as the reparameterisation trick where the VAE imposes a prior distribution p(z) on the latent variable z where z $= \mu + \epsilon\sigma$. Here, μ and σ are the mean and standard deviation of the latent distribution. Both μ and σ are modelled by the encoder. Latent variable, z can be considered as a style vector. This model has 530950 trainable parameters.

For the sake of a fair comparison, we maintain the same training hyper-parameter settings for each of these models like our proposed system. Before being fed to each model, the note level inputs pass through a dense layer that acts as an embedding layer and generates an embedding vector that serves as an input to these models, as proposed by our model.

6.4.2 Model Configuration & Training

In our proposed system, the note-level performance encoder is comprised of a dense network with a hidden size of 64 and a Tanh activation function that

	3-SF			4-SF			5-SF		
Model	precision	recall	F1-score	precision	recall	F1-score	precision	recall	F1-score
Bi-LSTM	0.414	0.401	0.406	0.455	0.451	0.450	0.466	0.461	0.462
HN	0.476	0.459	0.457	0.599	0.560	0.542	0.540	0.542	0.520
Transformer	0.416	0.406	0.411	0.455	0.435	0.445	0.527	0.513	0.519
HIPI-B	0.703	0.672	0.671	0.655	0.628	0.632	0.674	0.652	0.653
HIPI-VAE	0.668	0.664	0.660	0.681	0.680	0.689	0.646	0.634	0.631
HIPI	**0.762**	**0.759**	**0.756**	**0.759**	**0.752**	**0.750**	**0.754**	**0.737**	**0.737**

Table 6.3: Segment-wise precision, recall and F1-score by various model architectures for different feature combination as compared to our proposed model.

acts as an embedding layer, as well as a layer of Bi-LSTM with a hidden size of 64. The beat level encoder and the measure level encoder both consist of two layers of Bi-LSTMs with a hidden size of 64. For all three encoder layers, we set the dropout to 0.5. The dataset was divided into train-validation-test split with a ratio of 8:1:1. We have 378 training samples, with 366 from Beethoven and 12 from Mozart, along with 48 validation samples and 48 test samples. Our primary criterion for constructing the validation and test sets was to select compositions that has at least three distinct performances by each performer. This means that for a given composition, each performer would have multiple interpretations or performances of it. Thus, in the train, validation and test sets, every performer has distinct performances of the same composition. Essentially, while the piece remains consistent, the interpretation or style of performance by each pianist varies, ensuring a rich variety for evaluation. Each input sequence was sliced into equal size of 1000 notes in the measure level and if any piece contains less than 1000 notes we padded the sequence with 0. Each multidimensional input sequence is normalised to have zero mean and unit standard deviation. The model is trained using ADAM optimiser with a leaning-rate of 3e-3 and a weight decay of 1e-5. We also clipped the gradients with the maximum norm of 5 in the back-propagation. The model has 398854 trainable parameters.

6.4.3 Experiment 1: Segment-wise Performer Identification

Although RNNs, in particular LSTMs [161], have been used extensively for music modelling, they struggle to learn long term dependencies due to the vanishing gradient problem as the sequence length gets longer [250]. The inability of LSTMs to deal with music structure at different temporal resolutions hinders the modelling of expressive performance. Another major obstacle to training data-hungry deep learning models is the lack of large-scale performer datasets. Regardless of the fact that our dataset is one of the largest performer dataset in terms of individual performances, a complex model like ours would greatly benefit from being trained on more data. One common approach to circumvent this issue is to split each training sequence into a number of shorter sub-sequences. Therefore, each performance in the dataset is segmented into a least size of 1000-note slices, and every slice containing less than 1000 notes is padded with zero. The slicing happens in a measure level. If the end note of the slicing window comes in the first half of a measure boundary, we take the first note in the measure boundary as the end note, and if it appears in the latter half, we use the last note in the measure boundary as the end note of the slicing window.

Results:

The correlation between the length of the training examples and the classification accuracy achieved by our proposed model is illustrated in Figure 6.3 for various feature combinations. In accordance with our previous study [238], we selected the optimal combination of features, and combined at least three features to create a new feature for the classification task. Figures 6.3a, 6.3b, and 6.3c subsequently display the accuracies for a combination of three features (IOI, DL, ND), four features (IOI, DL, ND, OTD), and five features (IOI, DL, ND, OTD, OT) stacked together to create a single multidimensional feature sequence, represented by 3-SF (three stacked features), 4-SF(four stacked features), and 5-SF(five stacked features) respectively. These figures demonstrate that the model's accuracy improves with an increasing training sample size, and that the model performs best when the sequence length is 1000 for all feature settings.

To evaluate the performance of our proposed model and the effectiveness of the combined features for performer classification, we compared our model with various baseline models as described in Section 6.4.1. Table 6.3 illustrates the results of the model evaluations for different feature combinations in terms of precision, recall, and F1-score. The results indicate that our proposed model surpasses all other models in terms of performance for all three feature settings, with the highest precision (0.762) achieved by our model when using the 3-SF feature. The table reveals that the Bi-LSTM

	3-SF			4-SF			5-SF		
Model	precision	recall	F1-score	precision	recall	F1-score	precision	recall	F1-score
Bi-LSTM	0.443	0.441	0.441	0.531	0.522	0.526	0.575	0.562	0.565
HN	0.575	0.528	0.537	0.620	0.597	0.571	0.654	0.653	0.622
Transformer	0.641	0.632	0.636	0.660	0.651	0.655	0.673	0.666	0.669
HIPI-B	0.799	0.764	0.767	0.799	0.764	0.769	0.778	0.764	0.760
HIPI-VAE	0.776	0.778	0.775	0.797	0.792	0.790	0.724	0.708	0.710
HIPI	**0.842**	**0.833**	**0.832**	**0.820**	**0.819**	**0.816**	**0.779**	**0.767**	**0.773**

Table 6.4: Piece-wise precision, recall and F1-score by various model architectures for different feature combination as compared to our proposed model.

network exhibited inferior performance in identifying performers for all three features. The results also demonstrate the importance of the attention mechanism for performer identification, as the Hierarchical Network (HN), which lacks the hierarchical attention mechanism, performed less favorably when compared to our model. These results further validate the proposed hierarchical modeling approach, as it is shown that all models (HIPI-B, HIPI-VAE, and *HIPI*) that encode the input representation hierarchically produce high precision compared to non-hierarchical models.

6.4.4 Experiment 2: Piece-wise Performer Identification

In accordance with the discussions presented in Section 6.4.3, a prevalent technique in deep learning is to segment musical pieces into smaller sequences, thereby increasing the diversity of training examples and minimizing the sequence length to mitigate the vanishing gradient problem in recurrent neural networks. However, this approach limits the ability of the models to learn across time scales larger than the predefined window size. Given the hierarchical nature of musical structure and performance, it is imperative that neural networks are able to model these longer-term relationships in order to learn the global aspects of performance expression necessary for tasks such as performer identification and expressive performance generation. Despite this need, the challenge of modeling very long temporal sequences with deep learning remains unresolved [251].

In order to address this issue, we propose a new training approach that involves dividing each multi-dimensional feature sequence, representing an individual performance within a batch of training data, into smaller contiguous segments with a size of 1000 notes each. This segmentation size was selected as the optimal value based on empirical results in segment-wise pianist classification (see Figure 6.3). Any segments with less than 1000 notes are padded with zeros. Each segment within the batch is labeled with the original performer label that was assigned to the corresponding performance. The segments of each performance within the batch are then presented to the model as a single mini-batch, with a mini-batch size equal to the number of segments per performance. The loss is computed as the sum of individual loss values across the mini-batch, resulting in a scalar value representing the loss for a single, non-segmented performance. Summing the losses over observations for each mini-batch is considered more appropriate, as it provides a direct measure of the total loss across all segments of a performance within a batch. Finally, the mean cross-entropy loss for each mini-batch within the batch is calculated, resulting in a scalar value representing the average loss over all examples in a batch. This mean loss is then back-propagated to adjust the model's weights. The loss function can be formulated as follows:

$$L_{CE_{Total}} = -\frac{1}{K}\sum_{j=1}^{K}\sum_{i=1}^{C} y_i log(\hat{y_i}), \tag{6.12}$$

where K is the total sequences in a batch and j is the sequence index.

During the inference, the same approach is employed, in which a test piece is divided into equal parts using a window size of 1000 notes. The resulting segments are then presented to the model as a single batch. The final classification is obtained by averaging the model's outputs for that specific batch and applying the log-softmax function to the averaged output.

Results:

In order to validate our proposed model and the combined features for piece level performer classification, we compare its performance to that of several baseline models (see Section 6.4.1) trained and tested on our proposed dataset. Table 6.4 shows the classification results in terms of precision, recall and F1-score for various baseline models as well as our proposed model using three different combination features. The results demonstrate that the combination of IOI, DL and ND deviation features represented as 3-SF outperforms the other two feature category and our proposed model produces the best overall outcome. We also observe that the OT (onset time) deviation feature adversely affect our hierarchical models when combined with

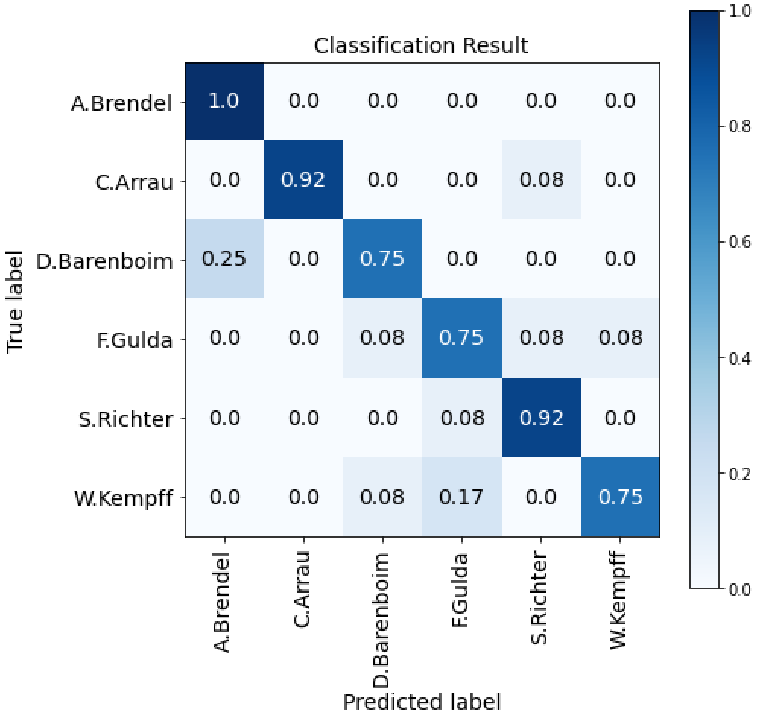


Figure 6.4: Normalised confusion matrix of 6-performer classification with *HIPI* using 3-SF feature.

the IOI, DL, and ND deviation features.

As can be seen in Table 6.4, our proposed model achieves the highest precision of 0.842 as compared to the baselines, whereas the vanilla Bi-LSTM network performs the poorest for any combination features. Moreover, given that the Hierarchical Network (HN) is identical to our model except for the lack of hierarchical attention mechanism, the results also demonstrate the significance of the attention mechanism for performer identification. We also notice that all the hierarchical models (HIPI-B, HIPI-VAE and *HIPI*) outperform the Transformer model. We assume that this is due to the fact that some musical pieces in our dataset exhibit strong local structures and transformer model equipped with multi-head attention mechanism is well known for capturing the long-term dependencies but fails to capture the local structures in sequence. On the contrary, our proposed hierarchical models capture both the local and global dependencies using different hierarchical level of attention.

	3-SF		4-SF		5-SF	
Model	seg-acc(%)	piece-acc(%)	seg-acc(%)	piece-acc(%)	seg-acc(%)	piece-acc(%)
Bi-LSTM	40.3	44.1	45.0	52.4	47.1	57.3
HN	46.3	50.0	51.0	58.3	53.0	65.2
Transformer	41.2	63.8	44.5	66.7	52.0	67.5
HIPI-B	69.4	76.4	66.3	72.2	66.5	79.2
HIPI-VAE	66.7	77.8	69.2	77.8	64.9	69.4
convnet-beat	**79.1**	81.9	**81.8**	**84.2**	**81.8**	**83.3**
HIPI	76.0	**84.7**	75.2	81.9	69.5	76.4

Table 6.5: Comparison of our proposed models with various baseline models for different feature combinations in terms of segment and piece wise accuracy(in percentage).

6.5 Analysis & Discussion

As illustrated in Section 6.4.3 and 6.4.4, there are two experiments presented on identifying virtuoso pianists based on their playing styles. The first experiment involves attempting to identify pianists based on segmented piece of music, whereas, the second one investigates the same problem while considering the full piece of music. For both of the case studies we used our proposed dataset presented in Section 6.2 and applied different combination features for performer identification. Additionally, a study contrasting the performance of several classification models in terms of precision, recall, and F1-score is provided for both the cases. Finally, comparing the two sets of results obtained from the two case studies indicates that the proposed combination features contain sufficient information to identify the performers in our dataset, and the exploration of various deep learning methods demonstrates the ability to learn performance patterns that help distinguish these performers

Table 6.5 shows the classification results of our proposed models, HIPI and convnet-beat (see Chapter 4), as well as various baseline models for different feature combinations in terms of segment and piece-wise accuracy. It shows that our proposed model, *HIPI*, achieves the highest test accuracy (**84.7%**) for piece wise classification. On the other hand, our other proposed model *convnet-beat*, achieves its highest accuracy in the 4-SF feature combination, with piece-wise accuracy reaching 84.2%, closely matching the performance of HIPI. HIPI, with its hierarchical structure of note, beat, and measure encoders, offers a more in-depth analysis of performance details, capturing nuances across multiple musical granularities. In contrast, the convnet-beat, designed for beat-level nuances, excels in rhythmic distinctions but may miss broader patterns. While both models exhibit high accuracy, the complexity and adaptability of HIPI seem to give it an advantage in the diverse challenges of pianist identification.

In addition, an accuracy of 84.7% is indeed a high success rate in a 6-way classification task while this would be an extremely challenging task even for a trained musician; imagine being required to listen to six different pianists play the same piece, and then having to identify the pianists in a recording of a different piece. It is also evident that the classification accuracy is correlated with the model complexity, where a simple model, such as Bi-LSTM, achieves the lowest accuracy and, as we go down the rows in Table 6.5, the model complexity increases so as the accuracy for both the segment and piece wise classification. This is comprehensible, since gaining insight from such lengthy musical sequences requires a complex modelling approach.

The performance of transformer network appears to be poor as compared to our proposed models for both the segment and piece wise classification cases, despite their potential for modelling and generating music with increased structural complexity [252, 253, 248]. We hypothesise that the musical pieces in our dataset have significant local structural dependencies, and that the transformer models, supported by a multi-head attention mechanism, learn the global temporal relationship but have trouble learning the local contextual dependencies. Unlike RNNs, which contain recurrence in their structure (where previous information is propagated over future time steps), the original transformer model does not, instead relying on a method called positional encoding to try to capture the temporal relationships between sequence elements. However, positional encoding hinders the model's ability to learn the precise representation of the input. The hypothesis is supported by the findings of our study, which show that the transformer model performs well for the piece level classification as opposed to the segment level classification.

A confusion matrix generated by our model for the piece-level classification using combination feature 3-SF is depicted in Figure 6.4, showing the class-wise accuracies. This demonstrates the utility of the feature for characterising individual performance style that facilitates the identification of performers. We can see that using the combination of IOI, DL and ND deviation features, our model can identify the performers most of the time. For instance, the model correctly identifies Alfred Brendel, Claudio Arrau and Sviatoslav Richter more often than it does Daniel Barenboim, Friedrich Gulda and Wilhelm Kempff. The identification of Alfred Brendel, Claudio Arrau and Sviatoslav Richter are mostly reliable with a precision of more than 90%.

6.6 Summary

In this chapter, we present a hierarchical approach for sequence modeling that addresses the challenge of automatic identification of virtuoso pianists based on their playing style. The proposed model integrates Recurrent Neural Networks (RNNs) and hierarchical multi-head attention to exploit the benefits of both recurrence and attention for music modeling. The combination of recurrence and attention is crucial for capturing both the short-term and long-term structures in pianist performances. Recurrence in the proposed model enables the learning of short-term structural dependencies in music. Meanwhile, the attention mechanism enhances the learning of long-term structural dependencies. Capturing both the short-term and long-term structures is important for pianist identification as it enables the model to take into account both the individual notes being played and the overall structure and progression of the piece.

The attention mechanism in the proposed model enables the model to focus on the most relevant parts of the input, such as highlighting the notes emphasized by a specific pianist or specific measures where the pianist applied more variations. This can be thought as an attention-based pooling approach, where lower-level information is summarized by the attention mechanism to form a higher-level representation (beats or measures), which is analogous to a pooling layer in Convolutional Neural Networks. This allows the network to focus on the most important parts of the input and create a more compact and discriminative representation while retaining information that is relevant to the task.

We also construct a dataset of six virtuoso pianists performing the same set of music pieces derived from the ATEPP dataset (as presented in Chapter 5), which allows the exploration of expressive characteristics of different performance styles. Furthermore, incorporating a context-aware multi-head attention mechanism and training the model with a combination of note-level timing, dynamics, and articulation features both result in improved performance (85%) at a piece-level performer classification. These results indicate that the hierarchical approach for music modeling is effective in the challenging task of performer identification, which typically requires trained musicians.

In conclusion, we hypothesise that the proposed RNN-based *HIPI* model is more effective for performer identification compared to the *convnet-beat* model (Chapter 4). Although the *convnet-beat* model may achieve slightly higher accuracy, it is trained on a small, less diverse dataset and only considers a single level of hierarchy in the music. In contrast, the *HIPI* model is trained on a larger and more diverse dataset with multiple levels of musical hierarchy and the ability to capture sequential dependencies, leading to its superiority over the *convnet-beat* model. A potential future direction could be training the *convnet-beat* with the same subset of the ATEPP dataset and compare their performance for the identification task. In addition, we could consider comparing individual performances to an accepted quasi-interpretation and extract performance-related features instead of comparing to a mechanical performance derived from the score.

Chapter 7

Conclusions and Further work

This Dissertation encompasses the comprehensive investigation of designing and evaluating pianist identification methodologies. The present Chapter encapsulates the seminal contributions and derives fundamental conclusions from the elaborate system design and empirical evaluations chronicled throughout the Dissertation. The potential avenues for future advancement of this research are also outlined later in this chapter.

7.1 Summary of contributions

The primary contributions of this thesis can be succinctly outlined in the following five aspects:

- Performer oriented novel datasets construction.
- Development of expressive features for describing pianists' playing style.
- Comprehensive statistical algorithms for modelling and identifying pianist's style.
- A musically motivated deep learning based approach for pianists identification from small scale dataset.
- A hierarchical approach for pianist identification.

7.1.1 Datasets Construction

The development of data-driven models for expressive piano performance requires a corpus of musical performances from which the expressive components can be derived. A significant challenge in this domain is the scarcity of suitable datasets. Unlike other areas of Artificial Intelligence where the availability of large, standard datasets has facilitated comparison and advancement, only a limited number of publicly accessible datasets exist for piano performance. This is partly due to the stringent requirements that these datasets must fulfill, including: consistent data collection with high recording quality to ensure reliable training, validation and testing; representation that effectively captures the temporal and expressive aspects of performance; coverage of diverse musical genres, styles and periods; inclusion of symbolic scores as a ground truth performance representation; multiple performances of the same piece of music by multiple performers to encompass expressive nuances and performance idioms; precise performer information to enable expressiveness and style analysis across different performers; and sufficient dataset size to allow the model to generalize to new examples.

In order to address the challenges of limited availability of suitable datasets for piano performance, we initially developed a solo-piano dataset derived from the International Piano-e-competition [172] featuring 9 virtuoso pianists. This dataset, described in detail in Chapter 3, contains performances played and recorded on a Yamaha CFX concert Grand Piano. High-quality recordings were ensured through the utilization of state-of-the-art Disklavier Pro recording technology, capturing the performances in both raw audio and MIDI formats. However, to simplify the analysis and eliminate the need for manual annotation, we opted to use the MIDI format, as it conveniently describes individual note events through onset, offset, pitch, and velocity information. Although MIDI recordings may not fully encompass the intricacies and richness of raw audio recordings, they still serve as a useful representation of the performance. In addition to the MIDI recordings, we also obtained the corresponding MusicXML score files for each performance in the dataset. This dataset was utilized to evaluate the proposed pianist identification methods in Chapters 3 and 4.

The dataset described in Chapter 3 while large in terms of number of notes, was limited in terms of composition and performance diversity. To address this constraint, a new performer-focused, large-scale dataset called Automatically Transcribed Expressive Piano Performance (ATEPP) was constructed (see Chapter 5). This dataset comprises 11742 recordings by 49 virtuoso pianists and encompasses 1007 hours of music recordings, created through the application of state-of-the-art piano transcription models to audio recordings of performances. This allows for a wider range of performances and the examination of performer-specific expressiveness and playing styles. A subset of this dataset was utilized in Chapter 6 to train a hierarchical RNN model for performer identification. The ATEPP dataset is a valuable resource for researchers working in this field, as it has the potential to significantly advance the state-of-the-art in expressive piano performance and it can be used for a range of tasks beyond just performance analysis and generation such as performance attribute analysis, comparison of performances and styles, style transfer and performance visualization.

7.1.2 Expressive Feature Development

Following the creation of datasets, a crucial aspect of our methodology involves modeling expressive performance features of pianists. These features play a significant role in accurately capturing and characterizing the unique performance characteristics and individual playing styles of each pianist. Expressive performance in music refers to the intentional deviation from the musical score, which generally indicates tempo, velocity, and articulation without expressive variations. However, it is worth noting that scores may also contain dynamic markings and other indications related to expression and playing style. The variations in the actual performance compared to the score are a result of the performer's interpretive choices and stylistic decisions. To quantitatively represent these variations, we propose to use five expressive features such as onset time, inter-onset interval, off-time duration, dynamic level, and note duration.

The comparison of the values of these musical features between a score and a performance enables the calculation of the deviation of the performance from the original musical score. This deviation captures the musical expressiveness and personal interpretation added by the performer, revealing their unique musical style and interpretation. The comparison of these values can provide a detailed understanding of how the performer deviated from the written score and what musical elements were emphasized or altered in their performance. Additionally, it allows the identification of common expressive performance principles shared by many performers, as well as the distinctions among them in note level.

However, the score-based feature extraction method is encumbered by limitations in acquiring digital scores for older or lesser-known pieces of music, primarily arising from factors such as copyright restrictions that impede access without proper authorization, limitations in digitization processes, and potential inaccuracies in transcriptions of the original scores. Furthermore, the score-based features exhibit similarities (i.e. similar peaks and valleys) among performers as a result of being influenced by the structure of the piece (see Section 3.3.2.1). A potential resolution to these issues is the utilization of performance norm (as discussed in Section 3.3.2.2) as a reference point for comparison with actual performances. Our analysis indicates that performance norms, being insensitive to the structure of the music, yield more unique features (i.e. dissimilar peaks and valleys) for characterizing pianist's style (see Figure 3.5).

In order to measure the deviations between performance norms and actual performances, we first compute the same expressive features as used in score-deviation feature extraction, including onset time, inter-onset interval, off-time duration, dynamic level, and note duration, from both the norm and the actual performance, and subsequently calculate their differences. For instance, we determine the deviation of onset time from the performance norm by computing the difference between the onset times specified in the performance norm and those executed in the performance. Similarly, we determine the deviations of the remaining expressive features, thereby providing a more comprehensive comprehension of the performer's expressive decisions and the extent to which they deviate from a standard quasi interpretation.

In the analysis presented in Chapter 3 and Chapter 4, both score deviation and norm deviation features were extensively utilized for the task of performer identification. The results demonstrated the effectiveness of these proposed features in identifying pianists from given excerpts. Additionally, in Chapter 6, the score deviation features were modeled hierarchically for pianist identification, further highlighting their utility in this context.

7.1.3 Pianist Identification Using Statistical Distribution Models

The application of modeling techniques is imperative in the identification of pianists through the utilization of expressive performance features. The expressive performance features, although essential in capturing individual playing styles and characterizing the unique performance characteristics of each pianist, are not adequate for differentiating between pianists. Thus, as an initial study, we propose a pianist identification method that utilizes similarity calculation from note-level feature distribution. We leverage the global distribution of expressive features to characterize the performer's style, and calculate similarity between feature distributions of different performers using KL-divergence. Based on this similarity, we perform identification of the performer. We evaluate three distribution models - Histogram, Kernel Density Estimation (KDE), and Gaussian Mixture Model (GMM) - to model the distribution of features in Chapter 3, and compare the identification performance of each model. Our results demonstrate that KDE yields the best performance for both score and norm deviation features. Additionally, we compare our proposed method against standard machine learning methods such as KNN and SVM, commonly used for similar Music Information Retrieval tasks, and show that our proposed method outperforms these machine learning models.

In addition, we compared the pianist identification performance based on individual score and norm deviation features. Moreover, regardless of the model used, the onset time deviation feature tends to perform better out of all other individual features. Overall, the norm deviation individual features perform best in characterizing pianist's styles, with the highest F1-score achieved using the KDE distribution (0.626). However, since identifying performers from their playing is an exceptionally challenging task, using a single feature to capture a pianist's style is not sufficient. Therefore, we proposed a feature fusion method in Chapter 3 for the performer identification task. The results show that fused features perform significantly better than individual features for accurately identifying performers. The highest F1-score achieved by KDE is 0.923 when using the 4FF norm deviation feature. This indicates that characterizing the pianist's distinct styles is improved by combining IOI, OTD, VL, and ND features. Overall, our study has demonstrated the effectiveness of the proposed fusion features and the modelling techniques in identifying virtuoso pianists from unseen data, which is a challenging task even for trained musicians. However, using these simple models to model features has a drawback in that they cannot capture the intricate interplay between the music's structure and the performer's interpretation, where the performer's interpretation is influenced by aspects of the music's structure, such as its phrase structure.

7.1.4 Musically Motivated Convolutional Neural Network for Pianist Identification

Given the limitation of the statistical modeling approach outlined in Chapter 3, we present a novel pianist identification technique utilizing a multichannel 1D Convolutional Neural Network (CNN) in Chapter 4. Our model seeks to capture the nuanced temporal aspects of piano performance by incorporating musically motivated filter shapes in the first layer of the CNN. To specifically address the hierarchical nature of Western classical music, we implement a beat-specific kernel in the first layer, experimenting with various kernel sizes aligned with the music's beats. This enables the CNN to learn the micro-variations introduced by performers within each beat, such as timing, velocity, and articulation variations. Furthermore, this approach presents the opportunity to experiment with measure-specific filters to capture variations within each measure, including dynamics, harmonic structure, and melody, thus enabling the capture of the performance's global temporal relationship.

Evaluation of the proposed *convnet-beat* model demonstrates its effectiveness in identifying performers from unseen musical excerpts. Further analysis through visualization of the filters in the first channel for all three layers in the model reveals that the filters have learnt increasingly complex patterns as they progress deeper into the layers. Utilizing class activation maps (CAMs) also highlights the identification of many discriminative regions in the first layer, supporting the effectiveness of beat-specific filters in capturing beat-level micro-information. To mitigate the issue of overfitting, commonly encountered in neural networks, particularly CNNs, we employed various techniques such as batch normalization, Adaptive Average Pooling, and dropout. Furthermore, our model exhibits superior performance when compared to state-of-the-art methods. These findings showcase the potential of the *convnet-beat* model for pianist identification and provide a foundation for the development of more advanced music-specific models in Chapter 6.

7.1.5 Hierarchical Performance Modelling

The success of utilizing musically motivated *convnet-beat* model for modeling the hierarchical unit of Western classical music has inspired us to delve deeper into the exploration of music's hierarchical structures. To this end, we propose the implementation of a recurrent neural network-based hierarchical performance encoder model. The model employs a beat-level Long Short-Term Memory (LSTM) encoder that initially encodes performance information at the beat level. The outputs are then summarized by a multi-head attention mechanism, which serves as input to the measure-level LSTM encoder. The model is trained using note-level features derived from calculating the deviations of each performance from a mechanical performance produced by a reference score, with the goal of predicting the most likely pianist. Our approach leverages the ability of LSTMs to learn short musical ideas, while utilizing a multi-head attention mechanism to address known limitations of LSTMs in learning long-term dependencies, as previously highlighted by Bengio et al. [160].

Our proposed hierarchical model has been demonstrated to exhibit superior performance in comparison to baseline models for both segment-level and piece-level classification tasks. Furthermore, the utilization of the proposed model for performer identification at the piece-level achieved an impressive 85% accuracy, outperforming the segment-level classification model which achieved 76% accuracy. This serves as compelling evidence of the dependence and benefit of the proposed features on traditional musicological definitions of musical structure. Additionally, we observed that the proposed model also benefits from the integration of a context-aware attention mechanism, as it demonstrated higher accuracy in classification than a vanilla Hierarchical Recurrent Neural Network (RNN) model. This supports our hypothesis that the incorporation of context-aware attention to gain knowledge of song structure and integrate it into the model structure can lead to the generation of superior representations.

7.2 Further work

7.2.1 Transfer Learning for Performer Identification

Deep learning models, despite their demonstrated capability for learning tasks across various domains, require a significant quantity of data for training to achieve optimal performance. Specifically, a substantial amount of labeled data is required during the training phase, which can be a time-consuming and labor-intensive process. In particular, when dealing with the task of learning the playing styles of different pianists, it is essential to train the model with large datasets that possess sufficient richness and diversity, yet existing performer-oriented solo piano performance datasets are limited and small in size. Furthermore, when applying a pre-trained model to tasks outside of its original training scope, it is imperative to ensure consistency in the feature space and distribution of the test data with that of the training data in order to prevent suboptimal performance.

To address these challenges and further generalize the pianist identification method, transfer learning can be employed. Transfer learning is defined as the enhancement of learning in a new task, referred to as the target task, through the transfer of knowledge from a related task, referred to as the source task, that has already been learned [254]. One common approach in transfer learning is to fine-tune a pre-trained model from the source task to better align with the characteristics of the target task. This is because the learned features in the initial layers of the pre-trained model tend to be more generic, while those in the later layers are more specific to the source task dataset. The fine-tuning process entails making adjustments to the weights of the latter layers only, in order to optimize performance on the target task.

Transfer learning, a technique that leverages knowledge gained from related tasks to improve performance on a new task, has been widely adopted in various fields [255, 256, 257]. It has proven particularly effective in computer vision, as convolutional neural networks (CNNs) excel in capturing rich basic visual information, such as fundamental shapes or prototypical templates of objects, in the early layers. This knowledge can then be transferred to target tasks, such as object detection [258] or person re-identification [259], resulting in improved performance. Although pre-trained models specifically for pianist identification are not available, pre-trained networks trained on large datasets for this task can be used. For example, music auto-tagging models has been employed for tasks such as genre classification [260], violinist identification [261], and musical event classification [168] as they can learn low-level features like tempo, pitch, local harmony, or envelope in the early layers [195, 262]. By using a pre-trained music auto-tagging model, it is possible to transfer this low-level information to the task of pianist identification, which is crucial for understanding the individual style of pianists, as micro-variations in performance are often injected at a very low level.

7.2.2 Exploration of New Features

The proposed features in this study have been demonstrated to effectively represent the individual stylistic characteristics of performers. However, it is imperative to note that further exploration into performance-related features may yield even more profound results. One such avenue for investigation is the melody lead phenomenon [263], which is employed by musicians as a means of accentuating a particular voice above others, and has been shown through previous research [264] to be correlated with both expressiveness and the skill level of the performer. Additionally, while we have employed the use of precise onset times as recorded in MIDI files in our current study, it may be beneficial in exploring the calculation of note onset in relation to its in-tempo position based on predefined tempo within the score, as this could provide further insight into the nuances of performance.

The utilization of various pedals in the piano is a significant aspect of musical performance, allowing performers the ability to add subtle nuances and variations in tone and expression to their playing. This enhances their control over the sound of the piano, enabling the production of more dynamic and nuanced performances. However, the utilization of pedals can vary greatly among performers, with some using them frequently while others abstaining entirely. Nevertheless, the usage of pedal information can be obtained from MIDI files and employed as performance features. For example, Pedal information can be encoded at the note level by analyzing the pedal state at various points within the note, such as the onset, offset, and minimum pedal value between the note onset and offset, as well as between the offset and the next onset [265].

Bibliography

[1] Efstathios Stamatatos and Gerhard Widmer. Automatic identification of music performers with learning ensembles. *Artificial Intelligence*, 165(1):37–56, 2005.

[2] Huan Zhang, Jingjing Tang, Syed Rifat Mahmud Rafee, Simon Dixon, and György Fazekas. Atepp: A dataset of automatically transcribed expressive piano performance. In *23rd International Society for Music Information Retrieval Conference (ISMIR 2022)*, 2022.

[3] Fred Lerdahl and Ray S Jackendoff. *A Generative Theory of Tonal Music, reissue, with a new preface.* MIT press, 1996.

[4] John Rink et al. *Musical performance: A guide to understanding.* Cambridge University Press, 2002.

[5] Roger A Kendall and Edward C Carterette. The communication of musical expression. *Music perception*, 8(2):129–163, 1990.

[6] Caroline Palmer. Music performance. *Annual review of psychology*, 48 (1):115–138, 1997.

[7] Jonathan Dunsby. Guest editorial: performance and analysis of music. *Music Analysis*, pages 5–20, 1989.

[8] Alf Gabrielsson. Perception and performance of musical rhythm. In *Music, mind, and brain*, pages 159–169. Springer, 1982.

[9] Leonard B Meyer. *Emotion and meaning in music.* University of chicago Press, 2008.

[10] Neil P McAngus Todd. The dynamics of dynamics: A model of musical expression. *The Journal of the Acoustical Society of America*, 91(6): 3540–3550, 1992.

[11] Anders Friberg and Johan Sundberg. Does music performance allude to locomotion? a model of final ritardandi derived from measurements of stopping runners. *The Journal of the Acoustical Society of America*, 105(3):1469–1484, 1999.

[12] Marc Leman, Micheline Lesaffre, and Pieter-Jan Maes. Introduction: what is embodied music interaction? In *The Routledge companion to embodied music interaction*, pages 1–10. Routledge, 2017.

[13] Bruno H Repp. Diversity and commonality in music performance: An analysis of timing microstructure in schumann's "träumerei". *The Journal of the Acoustical Society of America*, 92(5):2546–2568, 1992.

[14] Werner Goebl. Skilled piano performance: Melody lead caused by dynamic differentiation. In *Proc. of the 6th Int. Conf. on Music Perception and Cognition.* Citeseer, 2000.

[15] CI Wang. Quantifying pianist style–an investigation of performer space and expressive gestures from audio recordings. *Master's thesis, New York University*, 2013.

[16] Craig Saunders, David R. Hardoon, John Shawe-Taylor, and Gerhard Widmer. Using string kernels to identify famous performers from their playing style. In Jean-François Boulicaut, Floriana Esposito, Fosca Giannotti, and Dino Pedreschi, editors, *Machine Learning: ECML 2004*, pages 384–395, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[17] Craig Saunders, David R Hardoon, John Shawe-Taylor, and Gerhard Widmer. Using string kernels to identify famous performers from their playing style. In *European Conference on Machine Learning*, pages 384–395. Springer, 2004.

[18] Maarten Grachten and Gerhard Widmer. Who is who in the end? recognizing pianists by their final ritardandi. In *ISMIR*, pages 51–56. Citeseer, 2009.

[19] Rafael Ramirez, Esteban Maestre, and Xavier Serra. Automatic performer identification in commercial monophonic jazz performances. *Pattern Recognition Letters*, 31(12):1514–1523, 2010.

[20] Rafael Ramirez, Esteban Maestre, Alfonso Perez, and Xavier Serra. Automatic performer identification in celtic violin audio recordings. *Journal of New Music Research*, 40(2):165–174, 2011.

[21] Yudong Zhao, György Fazekas, and Mark Sandler. Transfer learning for violinist identification. In *Proceedings of European Signal Processing Conference (EUSIPCO)*, 2022.

[22] Nadine Kroher and Emilia Gómez. Automatic singer identification for improvisational styles based on vibrato, timbre and statistical performance descriptors. In *ICMC*, 2014.

[23] Zain Nasrullah and Yue Zhao. Music artist classification with convolutional recurrent neural networks. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[24] Daniel PW Ellis. Classifying music audio with timbral and chroma features. *International Society for Music Information Retrieval Conference*, 2007.

[25] Xulong Zhang, Jiale Qian, Yi Yu, Yifu Sun, and Wei Li. Singer identification using deep timbre feature learning with knn-net. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3380–3384. IEEE, 2021.

[26] Malcolm Budd. *Music and the emotions: The philosophical theories.* Routledge, 2002.

[27] Stephen Davies. Emotions expressed and aroused by music: Philosophical perspectives. *In P. N. Juslin J. A. Sloboda (Eds.), Handbook of music and emotion: Theory, research, applications (pp. 15–43)*, 2010.

[28] Patrik N Juslin. Communicating emotion in music performance: A review and a theoretical framework. *In P. N. Juslin J. A. Sloboda (Eds.), Music and emotion: Theory and research (pp. 309–337)*, 2001.

[29] Jenny Boyd and Holly George-Warren. *Musicians in tune: Seventy-five contemporary musicians discuss the creative process.* Fireside, 1992.

[30] Patrik N Juslin. Five facets of musical expression: A psychologist's perspective on music performance. *Psychology of music*, 31(3):273–302, 2003.

[31] Justin London. Some theories of emotion in music and their implications for research in music psychology. *Musicae scientiae*, 5(1_suppl): 23–36, 2001.

[32] Stephen Davies. *Musical meaning and expression.* Cornell University Press, 1994.

[33] Roger Scruton. *The aesthetics of music.* Oxford University Press, 1997.

[34] Manfred Clynes. Toward a theory of man: Precision of essentic form in living communication. In *Information processing in the nervous system*, pages 177–206. Springer, 1969.

[35] Manfred Clynes. What can a musician learn about music performance from newly discovered microstructure principles (pm and pas). *Action and perception in rhythm and music*, 39, 1987.

[36] Alf Gabrielsson. Music performance research at the millennium. *Psychology of music*, 31(3):221–272, 2003.

[37] H Christopher Longuet-Higgins and Christopher S Lee. The perception of musical rhythms. *Perception*, 11(2):115–128, 1982.

[38] H Christopher Longuet-Higgins and Christopher S Lee. The rhythmic interpretation of monophonic music. *Music Perception*, 1(4):424–441, 1984.

[39] Caroline Palmer. Anatomy of a performance: Sources of musical expression. *Music Perception*, 13(3):433–453, 1996.

[40] John Rink. In respect of performance: The view from musicology. *Psychology of Music*, 31(3):303–323, 2003.

[41] Alexis Kirke and Eduardo R Miranda. An overview of computer systems for expressive music performance. *Guide to computing for expressive music performance*, pages 1–47, 2013.

[42] Alf Gabrielsson. The performance of music. In *The psychology of music*, pages 501–602. Elsevier, 1999.

[43] Eric F Clarke and W Luke Windsor. Real and simulated expression: A listening study. *Music Perception*, 17(3):277–313, 2000.

[44] Patrik N Juslin, Anders Friberg, and Roberto Bresin. Toward a computational model of expression in music performance: The germ model. *Musicae Scientiae*, 5(1_suppl):63–122, 2001.

[45] Sandrine Vieillard, Mathieu Roy, and Isabelle Peretz. Expressiveness in musical emotions. *Psychological research*, 76(5):641–653, 2012.

[46] Alexander Lerch, Claire Arthur, Ashis Pati, and Siddharth Gururani. Music performance analysis: A survey. *20th International Society for Music Information Retrieval Conference*, 2019.

[47] Gerhard Widmer and Werner Goebl. Computational models of expressive music performance: The state of the art. *Journal of new music research*, 33(3):203–216, 2004.

[48] Carlos E Cancino-Chacón, Maarten Grachten, Werner Goebl, and Gerhard Widmer. Computational models of expressive music performance: A comprehensive and critical review. *Frontiers in Digital Humanities*, 5:25, 2018.

[49] Bruno Gingras, Marcus T Pearce, Meghan Goodchild, Roger T Dean, Geraint Wiggins, and Stephen McAdams. Linking melodic expectation to expressive performance timing and perceived musical tension. *Journal of Experimental Psychology: Human Perception and Performance*, 42(4):594, 2016.

[50] Anders Friberg. pdm: an expressive sequencer with real-time control of the kth music-performance rules. *Computer Music Journal*, 30(1): 37–48, 2006.

[51] Maarten Grachten and Gerhard Widmer. Linear basis models for prediction and analysis of musical expression. *Journal of New Music Research*, 41(4):311–322, 2012.

[52] Carlos Eduardo Cancino Chacón and Maarten Grachten. The basis mixer: a computational romantic pianist. In *Late-Breaking Demos of the 17th International Society for Music Information Retrieval*, 2016.

[53] Ian Simon and Sageev Oore. Performance rnn: Generating music with expressive timing and dynamics. `https://magenta.tensorflow.org/performance-rnn`, 2017.

[54] Katerina Kosta, Oscar F Bandtlow, and Elaine Chew. Practical implications of dynamic markings in the score: is piano always piano? In *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*. Audio Engineering Society, 2014.

[55] Katerina Kosta, Oscar F Bandtlow, and Elaine Chew. A change-point approach towards representing musical dynamics. In *International Conference on Mathematics and Computation in Music*, pages 179–184. Springer, 2015.

[56] Katerina Kosta, Rafael Ramírez, Oscar F Bandtlow, and Elaine Chew. Mapping between dynamic markings and performed loudness: a machine learning approach. *Journal of Mathematics and Music*, 10(2): 149–172, 2016.

[57] Carlos Eduardo Cancino-Chacón, Thassilo Gadermaier, Gerhard Widmer, and Maarten Grachten. An evaluation of linear and non-linear models of expressive dynamics in classical piano and symphonic music. *Machine Learning*, 106(6):887–909, 2017.

[58] Maarten Grachten and Carlos Eduardo Cancino Chacón. Temporal dependencies in the expressive timing of classical piano performances. *The routledge companion to embodied music interaction*, pages 360–369, 2017.

[59] Marco Marchini, Rafael Ramirez, Panos Papiotis, and Esteban Maestre. The sense of ensemble: a machine learning approach to

expressive performance modelling in string quartets. *Journal of New Music Research*, 43(3):303–317, 2014.

[60] Sergio I Giraldo and Rafael Ramirez. A machine learning approach to discover rules for expressive performance actions in jazz guitar music. *Frontiers in psychology*, 7:1965, 2016.

[61] Eric Cheng and Elaine Chew. Quantitative analysis of phrasing strategies in expressive performance: computational methods and analysis of performances of unaccompanied bach for solo violin. *Journal of New Music Research*, 37(4):325–338, 2008.

[62] Ching-Hua Chuan and Elaine Chew. A dynamic programming approach to the extraction of phrase boundaries from tempo variations in expressive performances. In *ISMIR*, pages 305–308, 2007.

[63] Marc Leman, Golan Friedland, and Peter Desain. *Computational models of expressive performance: a review*, pages 1–25. Routledge, 2007.

[64] Neil Todd. A model of expressive timing in tonal music. *Music perception*, 3(1):33–57, 1985.

[65] Neil Todd. A computational model of rubato. *Contemporary Music Review*, 3(1):69–88, 1989.

[66] Neil Todd. Towards a cognitive theory of expression: The performance and perception of rubato. *Contemporary Music Review*, 4(1):405–416, 1989.

[67] Guerino Mazzola and Oliver Zahorka. The rubato performance workstation on nextstep. In *ICMC*. Citeseer, 1994.

[68] Guerino Mazzola, Oliver Zahorka, and Joachim Stange-Elbe. Analysis and performance of a dream. In *Proceedings of the KTH Symposion on Grammars for Music Performance*, pages 59–68, 1995.

[69] Guerino Mazzola. *Geometrie der Töne: Elemente der Mathematischen Musiktheorie*. Springer-Verlag, 2013.

[70] Syed Rifat Mahmud Rafee, Gyorgy Fazekas, and Geraint A Wiggins. Performer identification from symbolic representation of music using statistical models. *International Computer Music Conference.*, 2021.

[71] Yudong Zhao, Changhong Wang, György Fazekas, Emmanouil Benetos, and Mark Sandler. Violinist identification based on vibrato features. In *2021 29th European Signal Processing Conference (EUSIPCO)*, pages 381–385. IEEE, 2021.

[72] John A Sloboda. The communication of musical metre in piano performance. *The quarterly journal of experimental psychology*, 35(2): 377–396, 1983.

[73] Efstathios Stamatatos. Quantifying the differences between music performers: Score vs. norm. In *International Computer Music Conference, ICMC*. Citeseer, 2002.

[74] Michel Bernays and Caroline Traube. Investigating pianist's individuality in the performance of five timbral nuances through patterns of articulation, touch, dynamics, and pedaling. *Frontiers in Psychology*, 5:157, 2014. ISSN 1664-1078. doi: 10.3389/fpsyg.2014.00157. URL `https://www.frontiersin.org/article/10.3389/fpsyg.2014.00157`.

[75] Cynthia CS Liem and Alan Hanjalic. Expressive timing from cross-performance and audio-based alignment patterns: An extended case study. In *ISMIR*, pages 519–524. Citeseer, 2011.

[76] Cynthia Liem, Alan Hanjalic, and Craig Sapp. Expressivity in musical timing in relation to musical structure and interpretation: a cross-performance, audio-based approach. In *Audio Engineering Society Conference: 42nd International Conference: Semantic Audio*. Audio Engineering Society, 2011.

[77] Mi Tian. *A cross-cultural analysis of music structure*. PhD thesis, 2017.

[78] Meinard Müller. *Music Representations*, pages 1–38. Springer International Publishing, Cham, 2021. ISBN 978-3-030-69808-9. doi: 10.1007/978-3-030-69808-9_1. URL `https://doi.org/10.1007/978-3-030-69808-9_1`.

[79] Michael Good. Musicxml for notation and analysis. *The virtual score: representation, retrieval, restoration*, 12(113-124):160, 2001.

[80] Jean-Pierre Briot, Gaëtan Hadjeres, and François-David Pachet. Deep learning techniques for music generation–a survey. *arXiv preprint arXiv:1709.01620*, 2017.

[81] Ole Martin Bjørndalen. Mido - midi objects for python. https://github.com/mido/mido/. Accessed on: 2022-12-22.

[82] MIDI Manufacturers Association (MMA). Official midi specifications. https://www.midi.org/specifications. Accessed on: 2022-12-22.

[83] Patrik N Juslin and Erik Lindström. Emotion in music performance. *The Oxford handbook of music psychology*, pages 377–389, 2009.

[84] Axel Berndt and Tilo Hähnel. Expressive musical timing. In *Audio Mostly 2009: 4th Conf. on Interaction with Sound—Sound and Emotion*, pages 9–16, 2009.

[85] Johan Sundberg, Anders Friberg, and Roberto Bresin. Attempts to reproduce a pianist's expressive timing with director musices performance rules. *Journal of New Music Research*, 32(3):317–325, 2003.

[86] David Huron. *Sweet anticipation: Music and the psychology of expectation*. MIT press, 2008.

[87] Anders Elowsson and Anders Friberg. Modeling the perception of tempo. *The Journal of the Acoustical Society of America*, 137(6): 3163–3177, 2015.

[88] Bruno H Repp. The art of inaccuracy: Why pianists' errors are difficult to hear. *Music Perception*, 14(2):161–183, 1996.

[89] Anders Elowsson and Anders Friberg. Predicting the perception of performed dynamics in music audio with ensemble learning. *The Journal of the Acoustical Society of America*, 141(3):2224–2242, 2017.

[90] Axel Berndt and Tilo Hähnel. Modelling musical dynamics. In *Proceedings of the 5th Audio Mostly Conference: A Conference on Interaction with Sound*, pages 1–8, 2010.

[91] Keiko Teramura, Hideharu Okuma, Yuusaku Taniguchi, Shimpei Makimoto, and Shin-ichi Maeda. Gaussian process regression for rendering music performance. *Proc. ICMPC*, pages 167–172, 2008.

[92] Carlos Eduardo Cancino Chacón and Maarten Grachten. An evaluation of score descriptors combined with non-linear models of expressive dynamics in music. In *International Conference on Discovery Science*, pages 48–62. Springer, 2015.

[93] Sam Van Herwaarden, Maarten Grachten, W Bas De Haas, Hsin-Min Wang, Yi-Hsuan Yang, Jin Ha Lee, et al. Predicting expressive dynamics in piano performances using neural networks. In *Proceedings of the 15th Conference of the International Society for Music Information Retrieval (ISMIR 2014)*, pages 45–52. International Society for Music Information Retrieval, 2014.

[94] Graham Grindlay and David Helmbold. Modeling, analyzing, and synthesizing expressive piano performance with graphical models. *Machine learning*, 65(2):361–387, 2006.

[95] Kenta Okumura, Shinji Sako, and Tadashi Kitamura. Stochastic modeling of a musical performance with expressive representations from the musical score. In *ISMIR*, pages 531–536. Citeseer, 2011.

[96] Tae Hun Kim, Satoru Fukayama, Takuya Nishimoto, and Shigeki Sagayama. Polyhymnia: An automatic piano performance system with statistical modeling of polyphonic expression and musical symbol interpretation. In *NIME*, pages 96–99, 2011.

[97] Colin Lawson and Robin Stowell. *The historical performance of music: an introduction.* Cambridge University Press, 1999.

[98] Samuel Adler and Peter Hesterman. *The study of orchestration*, volume 2. WW Norton New York, NY, 1989.

[99] Roberto Bresin and Giovanni Umberto Battel. Articulation strategies in expressive piano performance analysis of legato, staccato, and repeated notes in performances of the andante movement of mozart's sonata in g major (k 545). *Journal of New Music Research*, 29(3): 211–224, 2000.

[100] Sebastian Flossmann, Maarten Grachten, and Gerhard Widmer. Expressive performance rendering with probabilistic models. In *Guide to*

Computing for Expressive Music Performance, pages 75–98. Springer, 2013.

[101] Dasaem Jeong, Taegyun Kwon, Yoojin Kim, Kyogu Lee, and Juhan Nam. Virtuosonet: A hierarchical rnn-based system for modeling expressive piano performance. In *ISMIR*, pages 908–915, 2019.

[102] Yudong Zhao, György Fazekas, and Mark Sandler. Violinist identification using note-level timbre feature distributions. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 601–605. IEEE, 2022.

[103] David Roxbee Cox. *Principles of statistical inference.* Cambridge university press, 2006.

[104] Karl Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, 185: 71–110, 1894.

[105] Charles M Grinstead and J Laurie Snell. Conditional probability–discrete conditional. *Grinstead & Snell's introduction to probability Orange Grove Texts*, 2009.

[106] Dennis Howitt and Duncan Cramer. *Introduction to statistics in psychology.* Pearson education, 2007.

[107] Robert Freedman, David; Pisani and Roger Purves. *Statistics, Third Edition.* W. W. Norton & Company, 1997.

[108] Frederick J Gravetter and Lori-Ann B Forzano. *Research methods for the behavioral sciences.* Cengage learning, 2018.

[109] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10 (5):293–302, 2002.

[110] Ju-Chiang Wang, Hsin-Min Wang, and Gert Lanckriet. A histogram density modeling approach to music emotion recognition. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 698–702. IEEE, 2015.

[111] Elias Pampalk, Simon Dixon, and Gerhard Widmer. On the evaluation of perceptual similarity measures for music. In *of: Proceedings of the sixth international conference on digital audio effects (DAFx-03)*, pages 7–12, 2003.

[112] Bernard W Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018.

[113] David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[114] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[115] Jun Takagi, Yasunori Ohishi, Akisato Kimura, Masashi Sugiyama, Makoto Yamada, and Hirokazu Kameoka. Automatic audio tag classification via semi-supervised canonical density estimation. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2232–2235. IEEE, 2011.

[116] Michael I Mandel and Daniel PW Ellis. Song-level features and support vector machines for music classification. *International Society for Music Information Retrieval Conference (ISMIR 2005)*, 2005.

[117] Prashant Lahane and Arun Kumar Sangaiah. An approach to eeg based emotion recognition and classification using kernel density estimation. *Procedia Computer Science*, 48:574–581, 2015.

[118] Yi-Hsuan Yang and Homer H Chen. Prediction of the distribution of perceived music emotions using discrete samples. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(7):2184–2196, 2011.

[119] Rajeev Rajan and Hema A Murthy. Music genre classification by fusion of modified group delay and melodic features. In *2017 Twenty-third National Conference on Communications (NCC)*, pages 1–6. IEEE, 2017.

[120] Douglas A Reynolds. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663), 2009.

[121] Chandanpreet Kaur and Ravi Kumar. Study and analysis of feature based automatic music genre classification using gaussian mixture model. In *2017 International Conference on Inventive Computing and Informatics (ICICI)*, pages 465–468. IEEE, 2017.

[122] Ju-Chiang Wang, Yi-Hsuan Yang, Hsin-Min Wang, and Shyh-Kang Jeng. Modeling the affective content of music with a gaussian mixture model. *IEEE Transactions on Affective Computing*, 6(1):56–68, 2015.

[123] Roshni Ajayakumar and Rajeev Rajan. Predominant instrument recognition in polyphonic music using gmm-dnn framework. In *2020 International Conference on Signal Processing and Communications (SPCOM)*, pages 1–5. IEEE, 2020.

[124] Jana Eggink and Guy J Brown. A missing feature approach to instrument identification in polyphonic music. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 5, pages V–553. IEEE, 2003.

[125] George Tzanetakis, Andrey Ermolinskyi, and Perry Cook. Pitch histograms in audio and symbolic music information retrieval. *Journal of New Music Research*, 32(2):143–152, 2003.

[126] Simon Dixon, Fabien Gouyon, Gerhard Widmer, et al. Towards characterisation of music via rhythmic patterns. In *ISMIR*, 2004.

[127] Pedro Cano, Markus Koppenberger, and Nicolas Wack. Content-based music audio recommendation. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 211–212, 2005.

[128] Jonathan T Foote. Content-based retrieval of music and audio. In *Multimedia Storage and Archiving Systems II*, volume 3229, pages 138–147. SPIE, 1997.

[129] Juan P Bello. Measuring structural similarity in music. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(7):2013–2025, 2011.

[130] Bob L Sturm. A survey of evaluation in music genre recognition. In *International Workshop on Adaptive Multimedia Retrieval*, pages 29–66. Springer, 2012.

[131] Zhouyu Fu, Guojun Lu, Kai Ming Ting, and Dengsheng Zhang. A survey of audio-based music classification and annotation. *IEEE transactions on multimedia*, 13(2):303–319, 2010.

[132] Mathieu Barthet, György Fazekas, and Mark Sandler. Music emotion recognition: From content-to context-based models. In *International symposium on computer music modelling and retrieval*, pages 228–252. Springer, 2013.

[133] Arthur Flexer, Dominik Schnitzer, Martin Gasser, and Gerhard Widmer. Playlist generation using start and end songs. In *ISMIR*, volume 8, pages 173–178, 2008.

[134] Brian McFee, Luke Barrington, and Gert Lanckriet. Learning content similarity for music recommendation. *IEEE transactions on audio, speech, and language processing*, 20(8):2207–2218, 2012.

[135] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[136] Bent Fuglede and Flemming Topsoe. Jensen-shannon divergence and hilbert space embedding. In *International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings.*, page 31. IEEE, 2004.

[137] Jacob Goldberger, Shiri Gordon, Hayit Greenspan, et al. An efficient image similarity measure based on approximations of kl-divergence between two gaussian mixtures. In *ICCV*, volume 3, pages 487–493, 2003.

[138] Evelyn Fix and Joseph Lawson Hodges. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique*, 57(3):238–247, 1989.

[139] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[140] Per-Erik Danielsson. Euclidean distance mapping. *Computer Graphics and image processing*, 14(3):227–248, 1980.

[141] Vladimir Vapnik. Pattern recognition using generalized portrait method. *Automation and remote control*, 24:774–780, 1963.

[142] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

[143] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[144] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

[145] Jing Liu and Lingyun Xie. Svm-based automatic classification of musical instruments. In *2010 International Conference on Intelligent Computation Technology and Automation*, volume 3, pages 669–673. IEEE, 2010.

[146] Gursimran Kour and Neha Mehan. Music genre classification using mfcc, svm and bpnn. *International Journal of Computer Applications*, 112(6), 2015.

[147] Wei Chun Chiang, Jeen Shing Wang, and Yu Liang Hsu. A music emotion recognition algorithm with hierarchical svm based classifiers. In *2014 International Symposium on Computer, Consumer and Control*, pages 1249–1252. IEEE, 2014.

[148] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. Convolutional recurrent neural networks for music classification. In *2017 IEEE International conference on acoustics, speech and signal processing (ICASSP)*, pages 2392–2396. IEEE, 2017.

[149] Emmanouil Benetos, Simon Dixon, Zhiyao Duan, and Sebastian Ewert. Automatic music transcription: An overview. *IEEE Signal Processing Magazine*, 36(1):20–30, 2018.

[150] Jean-Pierre Briot, Gaëtan Hadjeres, and François-David Pachet. *Deep learning techniques for music generation*, volume 1. Springer, 2020.

[151] Yann LeCun et al. Generalization and network design strategies. *Connectionism in perspective*, 19(143-155):18, 1989.

[152] Maria V Valueva, NN Nagornov, Pavel A Lyakhov, Georgii V Valuev, and Nikolay I Chervyakov. Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. *Mathematics and computers in simulation*, 177:232–243, 2020.

[153] Yi-Tong Zhou and Rama Chellappa. Computation of optical flow using a neural network. In *ICNN*, pages 71–78, 1988.

[154] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning (adaptive computation and machine learning series). *Cambridge Massachusetts*, pages 321–359, 2017.

[155] Juhan Nam, Keunwoo Choi, Jongpil Lee, Szu-Yu Chou, and Yi-Hsuan Yang. Deep learning for audio-based music classification and tagging: Teaching computers to distinguish rock from bach. *IEEE signal processing magazine*, 36(1):41–51, 2018.

[156] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323 (6088):533–536, 1986.

[157] Jun Han and Claudio Moraga. The influence of the sigmoid function parameters on the speed of backpropagation learning. In *International workshop on artificial neural networks*, pages 195–201. Springer, 1995.

[158] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010.

[159] Shiv Ram Dubey, Satish Kumar Singh, and Bidyut Baran Chaudhuri. A comprehensive survey and performance analysis of activation functions in deep learning. *CoRR*, abs/2109.14545, 2021. URL `https://arxiv.org/abs/2109.14545`.

[160] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[161] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[162] Feynman T Liang, Mark Gotham, Matthew Johnson, and Jamie Shotton. Automatic stylistic composition of bach chorales with deep lstm. In *ISMIR*, pages 449–456, 2017.

[163] Jacek Grekow. Music emotion recognition using recurrent neural networks and pretrained models. *Journal of Intelligent Information Systems*, 57(3):531–546, 2021.

[164] Sebastian Böck, Florian Krebs, and Gerhard Widmer. Accurate tempo estimation based on recurrent neural networks and resonating comb filters. In *ISMIR*, pages 625–631, 2015.

[165] Chun Pui Tang, Ka Long Chui, Ying Kin Yu, Zhiliang Zeng, and Kin Hong Wong. Music genre classification using a hierarchical long short term memory (lstm) model. In *Third International Workshop on Pattern Recognition*, volume 10828, pages 334–340. SPIE, 2018.

[166] Gianluca Micchi. A neural network for composer classification. In *International Society for Music Information Retrieval Conference (ISMIR 2018)*, 2018.

[167] Juri Opitz and Sebastian Burst. Macro f1 and macro f1. *arXiv preprint arXiv:1911.03347*, 2019.

[168] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. Transfer learning for music classification and regression tasks. *arXiv preprint arXiv:1703.09179*, 2017.

[169] S Madeh Piryonesi and Tamer E El-Diraby. Data analytics in asset management: Cost-effective prediction of the pavement condition index. *Journal of infrastructure systems*, 26(1):04019036, 2020.

[170] Annette M Molinaro, Richard Simon, and Ruth M Pfeiffer. Prediction error estimation: a comparison of resampling methods. *Bioinformatics*, 21(15):3301–3307, 2005.

[171] Percy Goetschius. *Lessons in music form: A manual of analysis of all the structural factors and designs employed in musical composition.* Oliver Ditson Company, 1904.

[172] Braginsky Alexander and Yamaha Corporation. International piano-e-competition. Available: http://www.piano-e-competition.com/, 2002. Last checked on Oct 01, 2020.

[173] Werner Goebl and Roberto Bresin. Measurement and reproduction accuracy of computer-controlled grand pianos. *The Journal of the Acoustical Society of America*, 114(4):2273–2283, 2003.

[174] Bruno Gingras and Stephen McAdams. Improved score-performance matching using both structural and temporal information from midi recordings. *Journal of New Music Research*, 40(1):43–57, 2011.

[175] Chun-Ta Chen, Jyh-Shing Roger Jang, and Wenshan Liou. Improved score-performance alignment algorithms on polyphonic music. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1365–1369. IEEE, 2014.

[176] Eita Nakamura, Nobutaka Ono, Shigeki Sagayama, and Kenji Watanabe. A stochastic temporal model of polyphonic midi performance with ornaments. *Journal of New Music Research*, 44(4):287–304, 2015.

[177] Eita Nakamura, Kazuyoshi Yoshii, and Haruhiro Katayose. Performance error detection and post-processing for fast and accurate symbolic music alignment. In *ISMIR*, pages 347–353, 2017.

[178] Eita Nakamura, Nobutaka Ono, Yasuyuki Saito, and Shigeki Sagayama. Merged-output hidden markov model for score following of midi performance with ornaments, desynchronized voices, repeats and skips. *algorithms*, 21:8, 2014.

[179] Eita Nakamura, Nobutaka Ono, and Shigeki Sagayama. Merged-output hmm for piano fingering of both hands. In *ISMIR*, pages 531–536, 2014.

[180] Eric F Clarke. Generative principles in music performance, 1988.

[181] Caroline Palmer. Mapping musical thought to musical performance. *Journal of experimental psychology: human perception and performance*, 15(2):331, 1989.

[182] Efstathios Stamatatos. A computational model for discriminating music performers. In *Proceedings of the MOSART Workshop on Current Research Directions in Computer Music*, pages 65–69, 2001.

[183] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 03 1951. doi: 10.1214/aoms/1177729694. URL `https://doi.org/10.1214/aoms/1177729694`.

[184] Hiroaki Sasaki, Yung-Kyun Noh, Gang Niu, and Masashi Sugiyama. Direct density derivative estimation. *Neural Computation*, 28(6):1101–1140, 2016. doi: 10.1162/NECO_a_00835. URL `https://doi.org/10.1162/NECO_a_00835`. PMID: 27140943.

[185] J. R. Hershey and P. A. Olsen. Approximating the kullback leibler divergence between gaussian mixture models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, volume 4, pages IV–317–IV–320, 2007.

[186] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[187] Beici Liang, György Fazekas, and Mark Sandler. Transfer learning for piano sustain-pedal detection. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE, 2019.

[188] Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[189] Federico Simonetta, Carlos Cancino-Chacón, Stavros Ntalampiras, and Gerhard Widmer. A convolutional approach to melody line identification in symbolic scores. *arXiv preprint arXiv:1906.10547*, 2019.

[190] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[191] John Cristian Borges Gamboa. Deep learning for time-series analysis. *arXiv preprint arXiv:1701.01887*, 2017.

[192] Xiaowu Zou, Zidong Wang, Qi Li, and Weiguo Sheng. Integration of residual network and convolutional neural network along with various activation functions and global pooling for time series classification. *Neurocomputing*, 367:39–45, 2019.

[193] Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pages 1578–1585. IEEE, 2017.

[194] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[195] Keunwoo Choi, George Fazekas, Mark Sandler, and Jeonghee Kim. Auralisation of deep convolutional neural networks: Listening to learned features. In *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR*, pages 26–30, 2015.

[196] Jan Schlüter and Sebastian Böck. Improved musical onset detection with convolutional neural networks. In *2014 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 6979–6983. IEEE, 2014.

[197] Jordi Pons, Thomas Lidy, and Xavier Serra. Experimenting with musically motivated convolutional neural networks. In *2016 14th international workshop on content-based multimedia indexing (CBMI)*, pages 1–6. IEEE, 2016.

[198] Simon Dixon. Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30(1):39–58, 2001.

[199] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[200] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[201] Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F Schmidt, Jonathan Weber, Geoffrey I Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery*, 34(6):1936–1962, 2020.

[202] Zhiguang Wang and Tim Oates. Imaging time-series to improve classification and imputation. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[203] Werner Goebl and Simon Dixon. Analysis of tempo classes in performances of mozart sonatas. In *Proceedings of VII International Symposium on Systematic and Comparative Musicology and III International Conference on Cognitive Musicology*, pages 65–76, 2001.

[204] Asmir Tobudic and Gerhard Widmer. Relational ibl in classical music. *Machine Learning*, 64(1):5–24, 2006.

[205] Bruno H Repp. The dynamics of expressive piano performance: Schumann's "träumerei"revisited. *The Journal of the Acoustical Society of America*, 100(1):641–650, 1996.

[206] Gerhard Widmer, Simon Dixon, Werner Goebl, Elias Pampalk, and Asmir Tobudic. In search of the horowitz factor. *AI Magazine*, 24(3): 111–111, 2003.

[207] Qiuqiang Kong, Bochen Li, Xuchen Song, Yuan Wan, and Yuxuan Wang. High-resolution piano transcription with pedals by regressing onset and offset times. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3707–3717, 2021.

[208] Curtis Hawthorne, Ian Simon, Rigel Swavely, Ethan Manilow, and Jesse Engel. Sequence-to-sequence piano transcription with transformers. *arXiv preprint arXiv:2107.09142*, 2021.

[209] Michel Bernays and Caroline Traube. Expressive production of piano timbre: touch and playing techniques for timbre control in piano performance. In *Proceedings of the 10th Sound and Music Computing Conference (SMC2013)*, pages 341–346. KTH Royal Institute of Technology Stockholm, Sweden, 2013.

[210] Andrew Robertson. Decoding tempo and timing variations in music recordings from beat annotations. In *ISMIR*, pages 475–480, 2012.

[211] Bruno H Repp. Acoustics, perception, and production of legato articulation on a digital piano. *The Journal of the Acoustical Society of America*, 97(6):3862–3874, 1995.

[212] Craig Stuart Sapp. Comparative analysis of multiple musical performances. In *ISMIR*, pages 497–500, 2007.

[213] Maarten Grachten, Werner Goebl, Sebastian Flossmann, and Gerhard Widmer. Phase-plane representation and visualization of gestural structure in expressive timing. *Journal of New Music Research*, 38 (2):183–195, 2009.

[214] Curtis Hawthorne, Erich Elsen, Jialin Song, Adam Roberts, Ian Simon, Colin Raffel, Jesse Engel, Sageev Oore, and Douglas Eck. Onsets and frames: Dual-objective piano transcription. *arXiv preprint arXiv:1710.11153*, 2017.

[215] Curtis Hawthorne, Ian Simon, Rigel Swavely, Ethan Manilow, and Jesse Engel. Sequence-to-Sequence Piano Transcription with Transformers. *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR)*, 2021.

[216] Curtis Hawthorne, Erich Elsen, Jialin Song, Adam Roberts, Ian Simon, Colin Raffel, Jesse Engel, Sageev Oore, and Douglas Eck. Onsets and frames: Dual-objective piano transcription. *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018*, pages 50–57, 2018. doi: 10.5281/zenodo.1492341.

[217] Katerina Kosta, Oscar F Bandtlow, and Elaine Chew. MazurkaBL: Score-aligned loudness, beat, and expressive markings data for 2000

chopin mazurka recordings. In *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'18*, pages 85–94, 2018. ISBN 978-1-5251-0551-7.

[218] Tian Cheng, Simon Dixon, and Matthias Mauch. Modelling the decay of piano sounds. In *Proceedings of International Conference on Acoustic, Speech and Signal Processsing (ICASSP)*. IEEE, 2015. ISBN 9780992862633.

[219] B Series. Method for the subjective assessment of intermediate quality level of audio systems. *International Telecommunication Union Radiocommunication Assembly*, 2014.

[220] Katerina Kosta, Oscar F Bandtlow, and Elaine Chew. Mazurkabl: score-aligned loudness, beat, expressive markings data for 2000 chopin mazurka recordings. In *Proceedings of the fourth International Conference on Technologies for Music Notation and Representation (TENOR)(Montreal, QC)*, pages 85–94, 2018.

[221] Wilbert Jan Heeringa. *Measuring dialect pronunciation differences using Levenshtein distance.* PhD thesis, University Library Groningen][Host], 2004.

[222] Ning Chen, Wei Li, and Haidong Xiao. Fusing similarity functions for cover song identification. *Multimedia Tools and Applications*, 77(2): 2629–2652, 2018.

[223] Emilia Gomez. *Tonal Description of Music Audio Signals.* PhD thesis, University of Pompeu Fabra, 2006.

[224] Joan Serrà, Xavier Serra, and Ralph G Andrzejak. Cross recurrence quantification for cover song identification. *New Journal of Physics*, 11, 2009. doi: 10.1088/1367-2630/11/9/093017.

[225] Jordi Pons, Oriol Nieto, Matthew Prockup, Erik Schmidt, Andreas Ehmann, and Xavier Serra. End-to-end learning for music audio tagging at scale. *arXiv preprint arXiv:1711.02520*, 2017.

[226] Jort F Gemmeke, Daniel P W Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio

events. In *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[227] Francesco Foscarin, Andrew Mcleod, Philippe Rigaux, Florent Jacquemard, and Masahiko Sakai. ASAP : a dataset of aligned scores and performances for piano transcription. In *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[228] Qiuqiang Kong, Bochen Li, Jitong Chen, and Yuxuan Wang. GiantMIDI-Piano : A large-scale midi dataset for classical piano music. *Transactions of the International Society for Music Information Retrieval*, 5(1):87–98, 2022.

[229] Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. Enabling factorized piano music modeling and generation with the Maestro dataset. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–12, 2019.

[230] Zhengshan Shi, Craig Stuart Sapp, Kumaran Arul, Jerry McBride, and Julius O Smith. SUPRA: Digitaizing the stanford university piano roll archive. In *Proceeding of the 20th International Society on Music Information Retrieval (ISMIR)*, 2019.

[231] Verena Konz, Wolfgang Bogler, and Vlora Arifi-M. Saarland music data. *Late-Breaking and Demo Session of the International Conference on Music Information Retrieval (ISMIR)*, 2011.

[232] Mack T Henderson. *Rhythmic organization in artistic piano performance.* PhD thesis, note on label mounted at head of title.Iowa(1933)., 1937.

[233] Gino Brunner, Andres Konrad, Yuyi Wang, and Roger Wattenhofer. Midi-vae: Modeling dynamics and instrumentation of music with applications to style transfer. *arXiv preprint arXiv:1809.07600*, 2018.

[234] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American*

chapter of the association for computational linguistics: human language technologies, pages 1480–1489, 2016.

[235] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[236] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. In *29th AAAI conference on artificial intelligence*, 2015.

[237] Werner Goebl and I Band. *Numerisch-klassifikatorische Interpretationsanalyse mit dem". Bösendorfer Computerflügel"*, 1999.

[238] Syed Rifat Mahmud Rafee, Gyorgy Fazekas, and Geraint A. Wiggins. Performer identification from symbolic representation of music using statistical models. In *Proceedings of the International Computer Music Conference (ICMC)*, 2021.

[239] Iman Malik and Carl Henrik Ek. Neural translation of musical style. *arXiv preprint arXiv:1708.03535*, 2017.

[240] Akira Maezawa. Deep piano performance rendering with conditional vae. In *19th International Society for Music Information Retrieval Conference (ISMIR) Late Breaking and Demo Papers*, 2018.

[241] Sageev Oore, Ian Simon, Sander Dieleman, Douglas Eck, and Karen Simonyan. This time with feeling: Learning expressive musical performance. *Neural Computing and Applications*, 32(4):955–967, 2020.

[242] Ian Simon and Sageev Oore. Performance rnn: Generating music with expressive timing and dynamics, 2017, 2017.

[243] Dasaem Jeong, Taegyun Kwon, and Juhan Nam. Virtuosonet: A hierarchical attention rnn for generating expressive piano performance from music score. In *Neurips 2018 workshop on machine learning for creativity and design*, 2018.

[244] Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. Hierarchical multiscale recurrent neural networks. *ICLR*, 2017.

[245] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[246] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[247] Jake Grigsby, Zhe Wang, and Yanjun Qi. Long-range transformers for dynamic spatiotemporal forecasting. *arXiv preprint arXiv:2109.12218*, 2021.

[248] Neo Wu, Bradley Green, Xue Ben, and Shawn O'Banion. Deep transformer models for time series forecasting: The influenza prevalence case. *arXiv preprint arXiv:2001.08317*, 2020.

[249] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11106–11115, 2021.

[250] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

[251] Trieu H. Trinh, Andrew M. Dai, Minh-Thang Luong, and Quoc V. Le. Learning longer-term dependencies in rnns with auxiliary losses, 2018.

[252] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M Dai, Matthew D Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer. *arXiv preprint arXiv:1809.04281*, 2018.

[253] Kristy Choi, Curtis Hawthorne, Ian Simon, Monica Dinculescu, and Jesse Engel. Encoding musical style with transformer autoencoders. In *International Conference on Machine Learning*, pages 1899–1908. PMLR, 2020.

[254] Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global, 2010.

[255] Stevo Bozinovski. Reminder of the first paper on transfer learning in neural networks, 1976. *Informatica*, 44(3), 2020.

[256] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[257] Ankit Narendrakumar Soni. Application and analysis of transfer learning-survey. *International Journal of Scientific Research and Engineering Development*, 1(2):272–278, 2018.

[258] Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285–1298, 2016.

[259] Mengyue Geng, Yaowei Wang, Tao Xiang, and Yonghong Tian. Deep transfer learning for person re-identification. *arXiv preprint arXiv:1611.05244*, 2016.

[260] Beici Liang and Minwei Gu. Music genre classification using transfer learning. In *2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 392–393. IEEE, 2020.

[261] Yudong Zhao, György Fazekas, and Mark Sandler. Transfer learning for violinist identification. In *2022 30th European Signal Processing Conference (EUSIPCO)*, pages 239–243. IEEE, 2022.

[262] Keunwoo Choi, George Fazekas, and Mark Sandler. Explaining deep convolutional neural networks on music classification. *arXiv preprint arXiv:1607.02444*, 2016.

[263] Werner Goebl. Melody lead in piano performance: Expressive device or artifact? *The Journal of the Acoustical Society of America*, 110(1): 563–572, 2001. doi: 10.1121/1.1376133.

[264] Caroline Palmer. On the assignment of structure in music performance. *Music Perception*, 14(1):23–56, 1996.

[265] Dasaem Jeong, Taegyun Kwon, Yoojin Kim, and Juhan Nam. Score and performance features for rendering expressive music performances. In *Music encoding conference*, pages 1–6. Music Encoding Initiative Vienna, Austria, 2019.