Modelling Instrumental Gestures and Techniques:
A Case Study of Piano Pedalling

Beici Liang

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

Media and Arts Technology
School of Electronic Engineering and Computer Science
Queen Mary University of London
United Kingdom

July 2019

Statement of Originality

I, Beici Liang, confirm that the research included within this thesis is my own work, and that any ideas or quotations from the work of other people, published or otherwise, are 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 break any UK law, infringe any third partys copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

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 rests with the author and no quotation from it or information derived from it may be published without the prior written consent of the author.

Signature: Beici Liang

Date: 02/07/2019

Abstract

In this thesis we propose a bottom-up approach for modelling instrumental gestures and techniques, using piano pedalling as a case study. Pedalling gestures play a vital role in expressive piano performance. They can be categorised into different pedalling techniques. We propose several methods for the indirect acquisition of sustain-pedal techniques using audio signal analyses, complemented by the direct measurement of gestures with sensors.

A novel measurement system is first developed to synchronously collect pedalling gestures and piano sound. Recognition of pedalling techniques starts by using the gesture data. This yields high accuracy and facilitates the construction of a ground truth dataset for evaluating the audio-based pedalling detection algorithms.

Studies in the audio domain rely on the knowledge of piano acoustics and physics. New audio features are designed through the analysis of isolated notes with different pedal effects. The features associated with a measure of sympathetic resonance are used together with a machine learning classifier to detect the presence of legato-pedal onset in the recordings from a specific piano. To generalise the detection, deep learning methods are proposed and investigated. Deep Neural Networks are trained using a large synthesised dataset obtained through a physical-modelling synthesiser for feature learning. Trained models serve as feature extractors for frame-wise sustain-pedal detection from acoustic piano recordings in a proposed transfer learning framework.

Overall, this thesis demonstrates that recognising sustain-pedal techniques is possible to a high degree of accuracy using sensors and also from audio recordings alone. As the first study that undertakes pedalling technique detection in real-world piano performance, it complements piano transcription methods. Moreover, the underlying relations between pedalling gestures, piano acoustics and audio features are identified. The varying effectiveness of the presented features and models can also be explained by differences in pedal use between composers and musical eras.

Acknowledgements

First and foremost, I would like to thank my primary supervisor, Prof. Mark Sandler for opening me a door to the field of "music technology". Although I was just an undergraduate student majoring in electronic engineering, he gave me such opportunity to deduce the intricacies of music through computer science. From my early days as a PhD student in the Media and Arts Technology (MAT) programme, he encouraged me to join different meetings and seminars. He helped me get in contact with people at the Centre for Digital Music (C4DM) before I joined this amazing group. Then after the first training year, he supported my research with his always judicious advice throughout the following four years.

I am specially thankful to my second supervisor, Dr. György Fazekas for his impressive competence in all technical matters and constant kindness and patience in answering all the trivial questions I asked. He is not only a supervisor but also a friend to me. I would also like to thank my independent assessor, Dr. Andrew McPherson for the precise feedback on the hardware work. They all grounded my further proceedings. I couldn't ask for a better supervision panel.

I would like to express my profound gratitude to all the people who supported my research in a variety of ways. Thanks to Dr. Mi Tian for her guidelines and tips during my tough starting days; Dr. Siying Wang for sharing her work on score following; Dr. Tian Cheng for helping me get started with automatic piano transcription; Dr. Veronica Morfi for helping me set up servers to run deep learning models; Dr. Keunwoo Choi for his tutorials on designing neural networks; Dr. Ken O'Hanlon and Dr. Sebastian Ewert for the helpful discussions on music signal processing and their efforts on organising seminars; Irán Román and Zhengshan Shi for their summer workshop on deep learning for music information retrieval, where I learned a lot in just two weeks at CCRMA; Kevin MacManus and Daisuke Miura for helping me record datasets on the Disklavier piano at Yamaha Music Europe GmbH; Giulio Moro for his thorough suggestions on designing circuits and recording piano performance; Kok Ho Huen for his always prompt responses when I needed to print posters and find things in the electronics lab; Geetha Bommireddy for research equipment management, and Jonathan Winfield for administrative support. Special thanks to the anonymous reviewers of the papers I have submitted for the helpful comments.

I gratefully acknowledge financial support for my research, publications and conference presentations from various funding parties, including China Scholarship Council (No. 201406250007), EPSRC and AHRC Centre for Doctoral Training in Media and Arts Technology (EP/L01632X/1), and EPSRC Fusing Semantic and Audio Technologies for Intelligent Music Production and Consumption (EP/L019981/1).

Big thanks to my excellent research group C4DM and many others around MAT whom have made my time at QMUL lovely and inspirational: Adán, Adib, Adrien, Alejandro, Alessia, Alo, An, Andrea, Andrew, Angela, Betül, Bhusan, Carlos, Changhong, Chris C, Cornelia, Dalia, Dan, Daniel, Dave M, Delia, Emir, Emmanouil B, Florian, Francisco, Gary, Giacomo, Giulio, Hazar, Inês, Jack, Jacob, James, Janis, Jiajie, Johan, Josh, Ken, Laurel, Leshao, Liam, Liang, Lida, Lin, Lizzie, Lucia, Marco, Marcus, Mariano, Mathieu, Nick, Pablo, Panos, Peter H, Rémi, Rishi, Rod, Rosella, Saumitra, Simon, Sophie, Soomi, Thomas W, Thomas D, Tim, Veronica, Vinod, Will, Yudong, Yukun, Yogi and so on. Special thanks to Simin Yang and Di Sheng for all the various favours over the years inside and outside the office. Also thanks to researchers all over the world for their encouragement and advice on my work: Adrian, Alan, Alfonso, Antonella, Astrid, Aozhi, Bilei, Brecht, Bruno, Bochen, Chaogang, Chen, Chengze, Chris H, Chuanyi, Chunyang, Dave R, David W, Eita, Elaine, Elio, Emilia, Fabio, Gus, Irán, Jan, Jianyu, Jie, Jingting, Jordi, Juanjo, Julie, Junqi, José, Katerina, Keunwoo, Luca, Luwei, Maria, Meinard, Mi, Minwei, Miguel, Parham, Peter F, Qiuqiang, Rachel, Rebecca, Rodrigo, Rong, Sebastian, Shaoduo, Shengchen, Si, Siddharth, Sijia, Siying, Tao, Tian, Wanqi, Wei L, Xiao L, Xiao X, Xinyuan, Yading, Ye, Yi-Hsuan, Yongmeng, Zhengshan, Zijin, Zulfadhli and so on.

Thanks to April, Bingyu, Heng and Qianyun for being awesome flatmates in London. Thanks to Nuo, Haifang, Jiaxuan, Song, Ting, Wei C, Yang X, Xiaojing, Xin, Yuchuan,

Yuheng, Ziyu and all the other friends I met in Tianjin University and Peiyang Chorus. They strengthened my determination to pursue a degree in this field when I was stressed about my future in my early twenties. Thanks to my old buddies, Hongrui, Jingyi, Jingyuan, Shihui, Tianchi, Xu, Yang L and Yuxuan, for constantly cheering me up across thousands of miles. There are many other people that I am not mentioning here stayed besides me in some ways. They are too many to name, and I love them all.

Last but not least, my most heartfelt appreciation goes to my family for their unconditional, unlimited and unyielding support throughout my life. Thanks to Emmanouil Theofanis Chourdakis for clearing up my bad mood with scientific discussions, fun facts, geek jokes and exotic meals. He has been successfully filling my life with joy and love. Thanks to my phenomenal parents, Mrs. Yingchun Mu and Prof. Zhongbao Liang, for their open mind and massive encouragement. They are the source of my confidence, and I am so proud to be their daughter.

License

Table of Contents

List of Figures

List of Tables

List of Abbreviations

MIR	Music Information Retrieval
NIME	New Interfaces for Musical Expression
MIDI	Musical Instrument Digital Interface
DMI	Digital Musical Instrument
CPU	Central Processing Unit
PRU	Programmable Real-time Unit
ADC	Analogue-to-Digital Converter
DAC	Digital-to-Analogue Converter
GMS	Gesture and Motion Signal
GDIF	Gesture Description Interchange Format
SDIF	Sound Description Interchange Format
OSC	Open Sound Control
CSV	Comma-separated Value
AMT	Automatic Music Transcription
IPT	Instrumental Playing Techniques
TF	Time-Frequency
STFT	Short-time Fourier Transform
FT	Fourier Transform

FFT	Fast Fourier Transform
CQT	Constant-Q Transform
SpN	Sinusoids plus Noise
NMF	Non-negative Matrix Factorisation
RMS	Root-Mean-Square
SVM	Support Vector Machine
RBF	Radial Basis Function
C-SVM	SVM with RBF kernel
DT	Decision Tree
DT-SVM	Decision-Tree-based Support Vector Machine
KNN	K-Nearest Neighbour
GNB	Gaussian Naive Bayes
RF	Random Forest
HMM	Hidden Markov Model
EM	Expectation-Maximisation
CART	Classification and Regression Tree
FNN	Feedforward Neural Network
CNN	Convolutional Neural Network
ReLU	Rectified Linear Unit
AUC-ROC	Area Under the Receiver Operating Characteristic Curve
PCB	Printed Circuit Board
DTW	Dynamic Time Warping
GUI	Graphic User Interface
HPSS	Harmonic Percussive Source Separation
AIC	Akaike Information Criterion
2D	Two-dimensional

List of Symbols

f_{Hz}	Frequency in Herz
f_{mel}	Mel-scaled frequency
$\boldsymbol{x}$	A set of input data
i	Data point index
$\boldsymbol{x}_i$	The i-th data point in $\boldsymbol{x}$
y_i	The associated output of $\boldsymbol{x}_i$
$\hat{y}_i$	The predicted output of $\boldsymbol{x}_i$
$h_\theta(.)$	Logistic regression function with input coefficient θ
$g(.)$	Activation function
$J(.)$	Cost function
$\boldsymbol{\omega}$	Weight for the input data
b	Bias term in SVM
$h(\boldsymbol{x})$	The optimal margin classifier in SVM for $\boldsymbol{x}$
$K_{svm}(.)$	Kernel function in SVM
$\phi(.)$	Mapping function in SVM
C	Regularisation parameter in SVM
γ_i	Influence of $\boldsymbol{x}_i$ that reaches in SVM
O	Sequence of observations in HMM

Q	Sequence of states in HMM
ξ	HMM parameters
$P(O\|\xi)$	Probability of the observation sequence O in HMM with parameters ξ
$convolution(.)$	Convolution operation
c	Number of channels in CNN
l_c	Number of convolutional layers in CNN
m_c	Length in frequency axis of CNN's 2D kernel
n_c	Length in time axis of CNN's 2D kernel
m_p	2D Max-pooling's length in frequency
n_p	2D Max-pooling's length in time
TP	True Positive
FP	False Positive
FN	False Negative
N_{tp}	Number of TP
N_{fp}	Number of FP
N_{fn}	Number of FN
κ	Class index
P_1	Precision
R_1	Recall
F_1	F-measure
P_{micro}	Micro-averaged precision
R_{micro}	Micro-averaged recall
F_{micro}	Micro-averaged F-measure
P_b	Precision in boundary detection
R_b	Recall in boundary detection
F_b	F-measure in boundary detection

$D(.)$	Distribution function
μ	Mean of distribution
σ	Standard deviation of distribution
LTGO	Leave-Three-Group-Out
SSS	Stratified Shuffle Split
N	Label of normally played note
NLOH	Label of note played with *non-legato over-half* pedalling
NLH	Label of note played with *non-legato half* pedalling
LOH	Label of note played with *legato over-half* pedalling
LH	Label of note played with *legato half* pedalling
p	Partial index
f_p	Frequency of the p-th partial
F_0	Fundamental frequency
B	Inharmonicity coefficient
f_s	Sampling frequency
n	Frame index
n_{on}	Frame index of note onset
n_{off}	Frame index of note offset
m	Frequency bin index
$s[n]$	The n-th frame of a piano tone signal
$a_p[n]$	Amplitude of the p-th partial at the n-th frame
$f_p[n]$	Frequency of the p-th partial at the n-th frame
$\theta_p[n]$	Initial phase of the p-th partial at the n-th frame
$r[n]$	Residual component at the n-th frame
$RMSE(.)$	Root-mean-square energy function
L	Length of a frame

l	Sample index within a frame
$y_{pd}(.)$	Estimated function modelling the first-partial decay
α	Regression coefficients vector of $y_{pd}(.)$
t	Time in second
t_{on}	Note onset time
t_{off}	Note offset time
t_{dp}	Demarcation point in two-phase linear regression
R^2	Coefficient of determination
y_{rd}	Estimated exponential function modelling the residual decay
a_{rd}	Initial quantity of y_{rd}
λ	Exponential decay constant of y_{rd}
b_{rd}	Bias term in y_{rd}
χ^2_{rd}	The chi-squared statistic measuring the goodness of fit of y_{rd}
N_{peak}	Number of peaks calculated from the differences
F_{diff}	Maximum-amplitude frequency calculated from the differences
$\boldsymbol{S}$	Original spectrogram
$\boldsymbol{V}$	Reconstructed spectrogram by NMF
$\boldsymbol{W}$	Note template based on NMF
$\boldsymbol{W}^a$	Note template for the attack phase
$\boldsymbol{W}^d$	Note template for the decay phase
$\boldsymbol{H}$	Note activation based on NMF
$\boldsymbol{H}^a$	Note activation for the attack phase
$\boldsymbol{H}^d$	Note activation for the decay phase
$\boldsymbol{H}^s$	Spike-shaped note activation
$\boldsymbol{P}^t$	Transient pattern in the attack phase
N_a	Range of the transient in the attack phase

ρ	Pitch index
λ_ρ	Decay rate for the ρ-th pitch
$Thre$	Threshold for deciding note onset candidates
δ	Parameter for deciding $Thre$
$state$	On or off state for a pitch
$D_{KL}(.)$	Kullback-Leibler divergence function
$J_\rho(.)$	Cost function for states of the ρ-th pitch
$SRM(.)$	Sympathetic resonance measure function
SP_n	List of selected partials at the n-th frame
sp	Element of SP_n
m_{sp}	Frequency bin corresponding to sp
M_{sp}	Total number of m_{sp}
P	List of pitch index of transcribed note event
ON	List of onset time frame of transcribed note event
OFF	List of offset time frame of transcribed note event
PF	List of frequency bins corresponding to the estimated partials
Max_{linear}	Maximum value on linear scale
Max_{dB}	Maximum value on decibel scale
$Peak_{loc}$	Peak location
ST_{seg}	List of pedal-segment starting times in second
ET_{seg}	List of pedal-segment ending times in second
T_{ons}	List of pedal-onset times in second
ST_{pedal}	List of onset times in the final pedal detection results
ET_{pedal}	List of offset times in the final pedal detection results
$Thre_{onset}$	Threshold for detecting the pedal onset
$Thre_{segment}$	Threshold for detecting the pedalled segment

Chapter 1

Introduction

1.1 Scope and Motivation

Music performance is not only the realisation of categorical information presented in the score, such as pitch and note duration. It also involves interpretation by the player, leading to expressive performance. During a music performance, the performer can express a musical idea using a sequence of gestures that control the instrument, which in turn, produces sound. Accordingly, the musical idea is transformed into different representation domains: the score, the gesture and the sound domain. In the field of sound and music computing, the translation from score or/and gesture to sound is known as sound synthesis, and the other way round is often referred to as music information retrieval (MIR).

The scope of this thesis is the transcription aspect of MIR, which helps to reveal secrets of artistic expressions from recordings of virtuoso performances. We explore methods for acquiring *instrumental gestures*, which are regarded as the performer's gestures involved in the sound production process. Instrumental gestures are typically continuous. They can be categorised into discrete playing techniques. An accurate acquisition method should be able to reflect the intention of composers written in the score and the interpretation of performers, for instance, the use of pedals in piano performances.

Understanding instrumental gestures is essential as they serve as an integral part of both synthesis and transcription of music. Study on instrumental gestures inevitably requires an interdisciplinary approach with contributions from various fields including biomechanics and human motor control, auditory and visual perception, music performance analysis, music theory, music technology, robotics, human-computer interaction and so on [1]. In the meantime, the sound rendering of instrumental gestures correlates with the physics and acoustics of music instruments. A set of instrumental gestures is usually associated with music instruments of the same category. It could be completely different from another category. The interest of this thesis is narrowed into the study of instrumental gestures on the piano, which is one of the most important instruments in Western music due to its complexity and versatility. In particular, pedalling gestures in classical piano performances are investigated.

The interest in piano pedalling comes from the fact that the author of this thesis is a pianist. Studies on piano performances have abounded in the analysis of hand and finger movements, and recognition of basic units of music such as pitch and note onset. Little work has been done on piano pedalling even if it is regarded as "*the soul of the piano*" by the great 19th-century keyboard virtuoso Anton Rubinstein. To approach physics-based piano synthesis, acoustic effects of the sustain pedal on piano tones have been studied in [2] when the pedal is fully pressed, and in [3] when the half-pedalling technique is used. Yet, the full spectrum of pedalling as an instrumental gesture to convey different timbral nuances has not been adequately and quantitatively explored. Pedalling techniques were even considered "*almost impossible to gain from the audio domain*" in [4]. These challenges set barriers to a full transcription of piano music, which can benefit applications in piano pedagogy, audio-score alignment, musicology and many related domains.

Furthermore, learning to use the piano pedals strongly relies on listening to nuances in the sound. To develop critical listening, merely experimenting with different pedalling techniques can be time consuming and less effective. Instructions with respect to when the pedal should be pressed and for what duration are required. Therefore the main motivation of this thesis is to facilitate the learning process by developing methods of automatic pedalling technique detection either directly from the pedalling gestures or indirectly from audio recordings.

With a deeper understanding of instrumental gestures and piano pedalling introduced in the next two sections, we can break down the motivation into the following research questions, which will be addressed in this thesis:

1. What are the pros and cons of the existing methods for measuring instrumental gestures and detecting the corresponding playing techniques in piano performances?
2. How to design a non-intrusive measurement system that could accurately record how the piano sound is modulated by pedalling?
3. What features can represent different pedalling techniques in order to facilitate audio-based detection?
4. Can automatic detection of piano pedalling be improved by considering acoustics and physics of the piano?
5. How to incorporate all the knowledge to generalise the pedalling technique detection so it performs well on any pianos?

1.2 Instrumental Gestures and Techniques

The term *gesture* can be varied in definitions with respect to research fields. Accordingly, it can be approached from very different perspectives. Zooming into the musical domain, the gesture is considered equivalent to human movements [5]. This is because many musical activities, such as performance, conducting and dancing, involve body movements that evoke meaning, and therefore these movements are called gestures [6].

Musical gestures can be regarded as human movements that go along with sounding music. In a study on the musical gestures of the pianist Glenn Gould, they are divided into three levels "*from purely functional to purely symbolic*" [7]. Gestures of the symbolic level relate to an image of a physical gesture and are perceived by the audience subjectively. In the light of the works in [8–10], the first functional level can be further divided into four categories based on the functions of musical gestures in [6]:

- *Sound-producing gestures* are the ones that are effective in producing sound. They are called *instrumental gestures* in [9], and *effective gestures* in [7]. Based on the typology in [9], *excitation* and *modification* are the two subcategories[1].
- *Communicative gestures* are mainly intended for communication between performer and performer or performer and perceiver. They are called *semiotic gestures* in [11].
- *Sound-facilitating gestures* are the ones that support the sound-producing gestures, but not directly involved in sound production. They are called *accompanying gestures* in [7], *non-obvious performer gestures* in [12], and *ancillary gestures* in [10].
- *Sound-accompanying gestures* are not ancillary to sound production, but intended to follow the music, for instance, tracing the melody of a song or mimicking the sound-producing gestures in the air.

It is noted that the above categories are not mutually exclusive. Multiple functions can simultaneously exist in musical gestures. For example, releasing the sustain pedal in piano performances can be seen as both instrumental and communicative gestures. This is because it can not only mute the current sounding notes, but also indicate the end of a music phrase to other performers. The scope of this thesis focuses on using piano pedals as instrumental gestures. We can categorise the continuous pedalling gestures

[1] These terms presented in [6], i.e., *excitation* and *modification* for the subcategories of *sound-producing gestures*, correspond to *sound-producing gestures* and *sound-modifying gestures* for the subcategories of *instrumental gestures* in [9].

into discrete techniques. More details on pedalling techniques in piano performance are introduced in the following section.

1.3 Pedalling in Piano Performance

Our proposed methods for detecting pedalling techniques are informed by their associated acoustical characteristics in piano sounds. To help understand the intuition behind these methods, this section presents the music background of pianos and pedals. We start with a brief introduction of modern pianos. Due to different mechanisms and acoustics of the three standard pedals, piano sounds can be altered in different ways as introduced in Section 1.3.1. Instrumental gestures on the sustain pedal are the most diverse and commonly used. They can be categorised into a variety of pedalling techniques as detailed in Section 1.3.2.

1.3.1 Mechanism and Acoustics of Modern Pianos and Pedals

As illustrated in Figure 1.1, a grand piano features keyboard, hammers, dampers, bridges, soundboard, and strings. There are 88 keys on the keyboard of a modern piano, covering a range of notes from *A0* to *C8*. When a key is struck, a complex mechanism transmits this motion to the hammer. The hammer strikes a number of strings, depending on the played note. Most notes have three strings, except for the bass, which ranges from one to two. Almost at the same time of hammer strikes, dampers are lifted away from the strings, which are free to vibrate from this moment. Vibrating strings generate the piano tone, which consists of a fundamental tone and a number of the higher-pitched tones known as partials. When the string vibrations reach the bridge, they are transmitted to the soundboard via a complex coupling mechanism. Because of the string stiffness, partials of piano tones occur at frequencies slightly away from the harmonic positions (integer multiples of the fundamental frequency). This is referred to as *inharmonicity*. Experiments have shown that inharmonicity contributes to the warmth [13], richness and

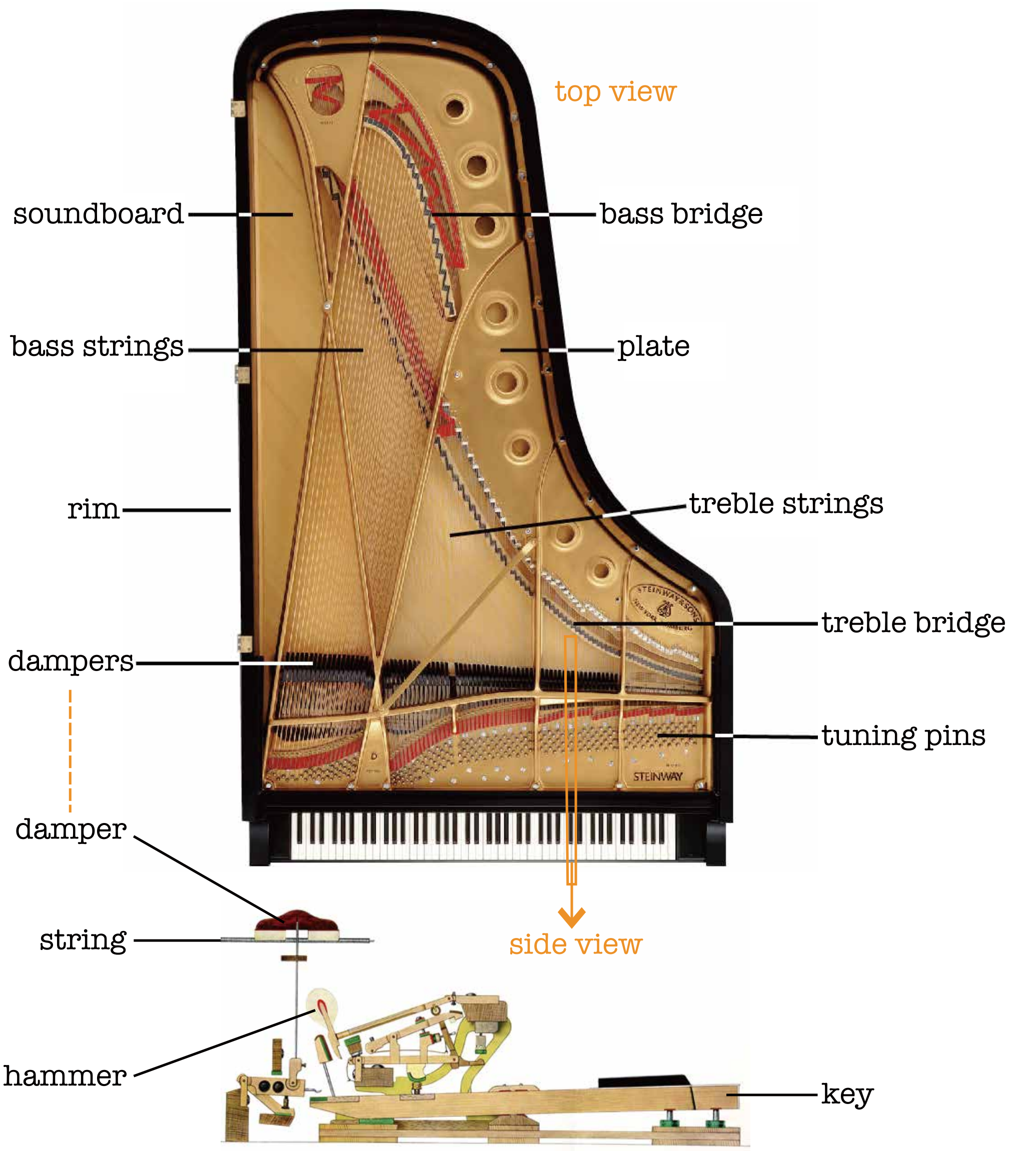


Figure 1.1: Structure of a grand piano.

quality of the piano sound [14].

Due to the hammer struck, a percussive sound is produced at the attack stage of a piano tone. The attack stage is followed by a complicated decay process because of double decay and beating [15]. Piano tone is silenced by dampers, which fall onto the strings when the performer's hands are lifted from the key. It is possible to make piano tones keep sounding if pedals are used. Pedals have existed in pianos since the 18th century when Cristofori introduced a forerunner of the modern soft pedal. It took many decades before piano designers settled on the standardisation of three pedals, following the configuration standardised by Steinway and Sons in the late nineteenth century [16]. From left to right, the three pedals are commonly referred to as the *una corda pedal*, the *sostenuto pedal* and the *sustain pedal*[2].

The una corda pedal of grand pianos functions by shifting the keyboard and hammers to the right such that one less string would be struck. Piano loudness can be decreased when the una corda pedal is pressed. Due to the changes in coupling between the strings, a more significant effect is the change in timbre. Unlike the grand piano, the una corda pedal of upright pianos makes the output sound softer by moving the hammer closer to the strings.

The function of the sostenuto pedal can vary with different pianos. In most modern grand pianos, the sostenuto pedal only sustains dampers that are lifted when the pedal is engaged. This effect leads to an impression of a third hand, because the sostenuto pedal keeps the chosen notes sounding while performers can freely use their hands to play other notes. In some upright pianos, the tone is softened with a piece of felt which is lowered between the hammers and the strings. This makes the pedal act as a "practice" pedal [18] because the loudness is greatly decreased.

Because of the varieties of pedalling techniques on the sustain pedal, it is the most frequently used one among the three standard piano pedals. All dampers are lifted off the strings when the sustain pedal is pressed. This mechanism helps to sustain the current sounding notes and allows strings associated with other notes to vibrate due to coupling via the bridge. A phenomenon known as *sympathetic resonance* [19] is thereby enhanced and embraced by pianists to create a "dreamy" sound effect. In this thesis, we focus on the sustain pedal and investigate its techniques which are detailed in the following section.

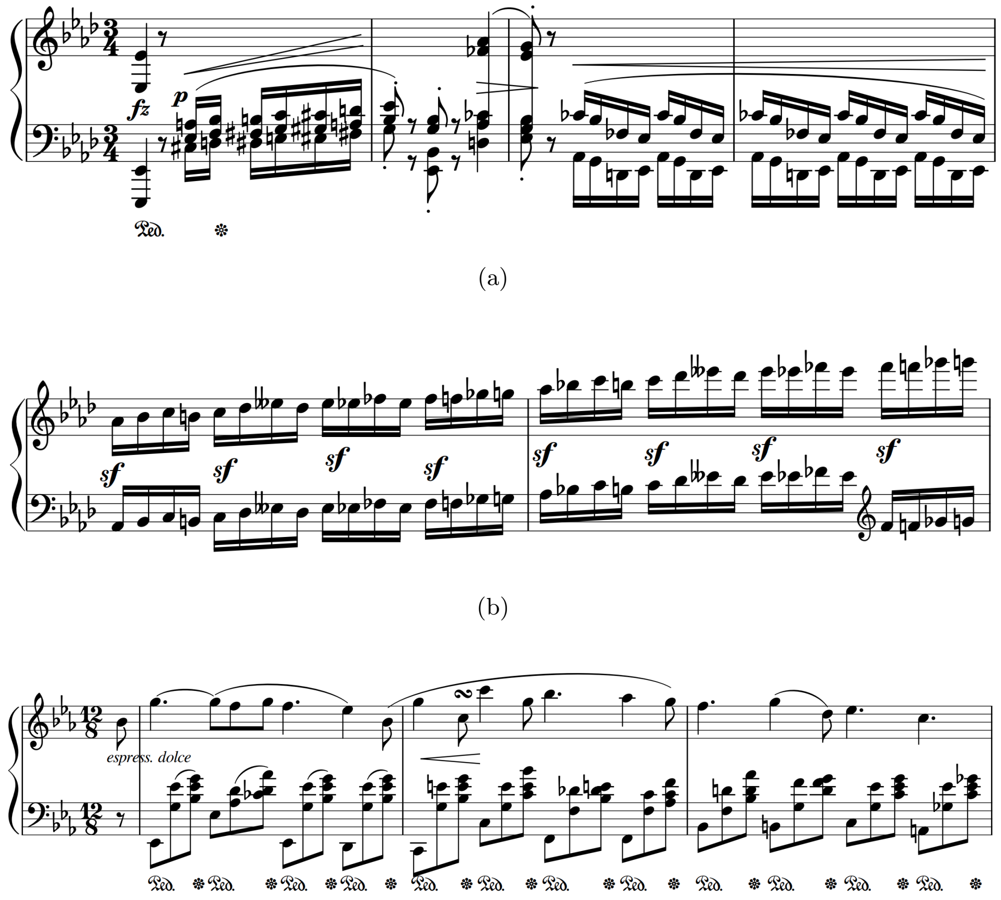
[2] The *una corda pedal* is also known as the *soft pedal*. The *sustain pedal* is also called *damper pedal*, *loud pedal*, or *open pedal* [17].

1.3.2 Pedalling Techniques and Notations

Mirroring the development of the pedals themselves, the notations used for indicating pedalling techniques have likewise changed over the centuries. The use of the pedals was not marked in music scores before the 1790s [20]. Composers like Chopin and Liszt marked pedal onset and offset times actively in their compositions. In contrast, Debussy and Scriabin rarely notated pedalling despite its importance in the interpretation of their works. Yet, they as well as later composers continued to find new sounds through the assumed use of pedals [21]. In general, pedalling techniques can be varied in timing with respect to note onsets and the depth of pedal press [20]. This is especially the case for the sustain pedal. Pianists apply various pedalling techniques on the sustain pedal to colour the resonance subtly, leading to expressive performance. In the rest of this section, pedalling techniques refer to the ones applied on the sustain pedal.

There are three main pedalling techniques related to pedal onset time, i.e., when the pedal should be pressed. *Anticipatory pedalling* can only be applied after silence and before the notes are played. This is primarily used by pianists to produce greater resonance at the commencement of the sound. *Rhythmic pedalling* corresponds to pressing the pedal at the same time as the note onset. This technique supports metrical accentuation, which is an important aspect of Classical-era[3] performance. Pressing the pedal immediately after the note attack is called *legato pedalling*. This enables the performer to produce seamless legato while avoiding blurring the sound with previous sonorities. Legato pedalling is more commonly used than the other two timing-related techniques. Figure 1.2 presents three music excerpts usually played with the three pedalling techniques mentioned above, respectively. The notations under the bar of the music score in Figure 1.2(a) and 1.2(c) can roughly suggest when the sustain pedal should be pressed and released. The actual timing is dependent on the interpretation of performers. In Figure 1.2(b), pedalling notations are not given, but the *sforzando* symbol indicates that it should be played along with rhythmic pedalling for a forceful accent at every beat.

[3] Classical-era extends roughly from the late 18th century to the mid 19th century.



(a)

(b)

(c)

Figure 1.2: Examples of three timing-related pedalling techniques in music excerpts, including (a) anticipatory pedalling in Chopin's Polonaise in A-flat major, Op. 53, (b) rhythmic pedalling in Beethoven's Sonata No. 32 in C minor, Op. 111, and (c) legato pedalling in Chopin's Nocturne Op. 9 No. 2.

As seen in Figure 1.2, "pressed" and "released" are the two conventional positions of the sustain pedal. It is also possible to use the pedal in intermediate positions. When the pedal is kept in an intermediate position, the dampers allow the strings to vibrate to some extent but prevent the strings from vibrating freely. Thus a tone can be heard in full strength while the key is held down. When the key is released, the volume will be reduced but some sound remains. As presented in [22], the German word *Nachklang* can refer to this remainder of sound. The volume of the *Nachklang* depends on the position of the dampers. Scarcely any *Nachklang* remains when the dampers are nearly touching the strings, while practically the full volume of sound remains when the dampers are almost completely removed from the strings. All gradations can be obtained by moving the pedal between these two positions. They are known as part-pedalling techniques that change as a function of the depth of the sustain pedal.

Apart from *full pedal*, pianist Schnabel defined another three levels of part-pedalling, which are referred to as *quarter pedal*, *half pedal* and *three-quarter pedal* [22]. It should be noted that these terms do not refer to specific positions of the pedal, nor even to specific positions of the dampers, but only to the amount of sound which remains when the keys are released. As Schnabel discussed, the only way to distinguish between these pedals or to judge whether they are performed correctly is by hearing the effect created. To test whether a certain position of the pedal produces the effect of *quarter pedal* accurately, play a scale or a succession of different harmonies: there should be no blurring until the last note has been played; play the same passage again without pedal: there should be a significant difference in sound. To test *half pedal*, play single staccato notes or chords: they should sound staccato; play a scale or succession of different harmonies: there should be some blurring. To test *three-quarter pedal*, play a chord and then release the keys: it should sound as if the chord were held out; play and releases the same chord again using full pedal: there should be a marked difference in sound. Many special effects can be created by changing directly from one intermediate position to another or between intermediate and full pedal. For example, a rapid *diminuendo* in a harmonic passage can be achieved by releasing the pedal gradually, thereby passing all intermediate positions.

Throughout the twentieth century, composers became increasingly more precise in providing pedalling instructions to performers. For example, more explicit pedalling notations are presented in Figure 1.3, which denotes pressing the sustain pedal halfway, and then fully before releasing it back to halfway for a moment, and finally keeping it fully pressed until it is released. Even if the pedal notations are given, experts agree that pedalling in the same piano passage can be executed in many different ways [23]. This is influenced by the performer's sense of tempo, dynamics, textural balance, and the settings or milieu in which the performance takes place [21].

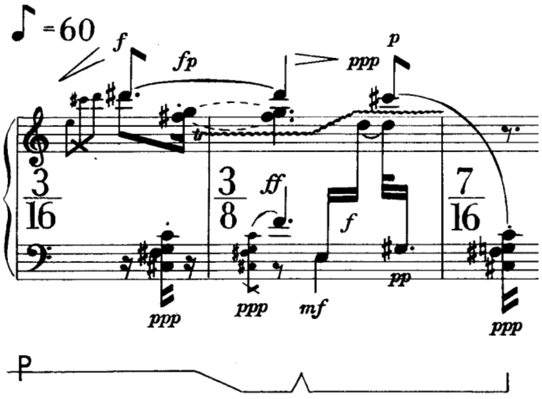


Figure 1.3: Example of explicit pedalling notations in the music excerpt from Stockhausen's Klavierstück IX.

In this thesis, our main focus is to develop automatic methods for detecting pedal onset times and localising the portions with the sustain pedal pressed. A portion between a pedal onset and its corresponding offset is hereafter referred to as *pedalled segment.*

1.4 Thesis Outline and Contributions

With a deeper understanding of instrumental gestures and piano pedalling introduced in Section 1.2 and 1.3, the main contributions of this thesis are made in pedalling technique detection on the sustain pedal. The emphasis of each chapter and relationships between different chapters are intuitively illustrated in Figure 1.4. In detail, the rest of this thesis is outlined as follows.

Chapter 2 presents the technical background of this thesis and reviews related works with an emphasis on piano performance. It starts by introducing three strategies for capturing the instrumental gestures and techniques: direct acquisition, indirect acquisition and multimodal modelling. These strategies are applied to the pedalling technique detection, which is introduced in Chapter 3, 5, 6 and 7. Methods for evaluating the detection performance are also surveyed.

Chapter 3 presents a dedicated system for direct acquisition. The system enables

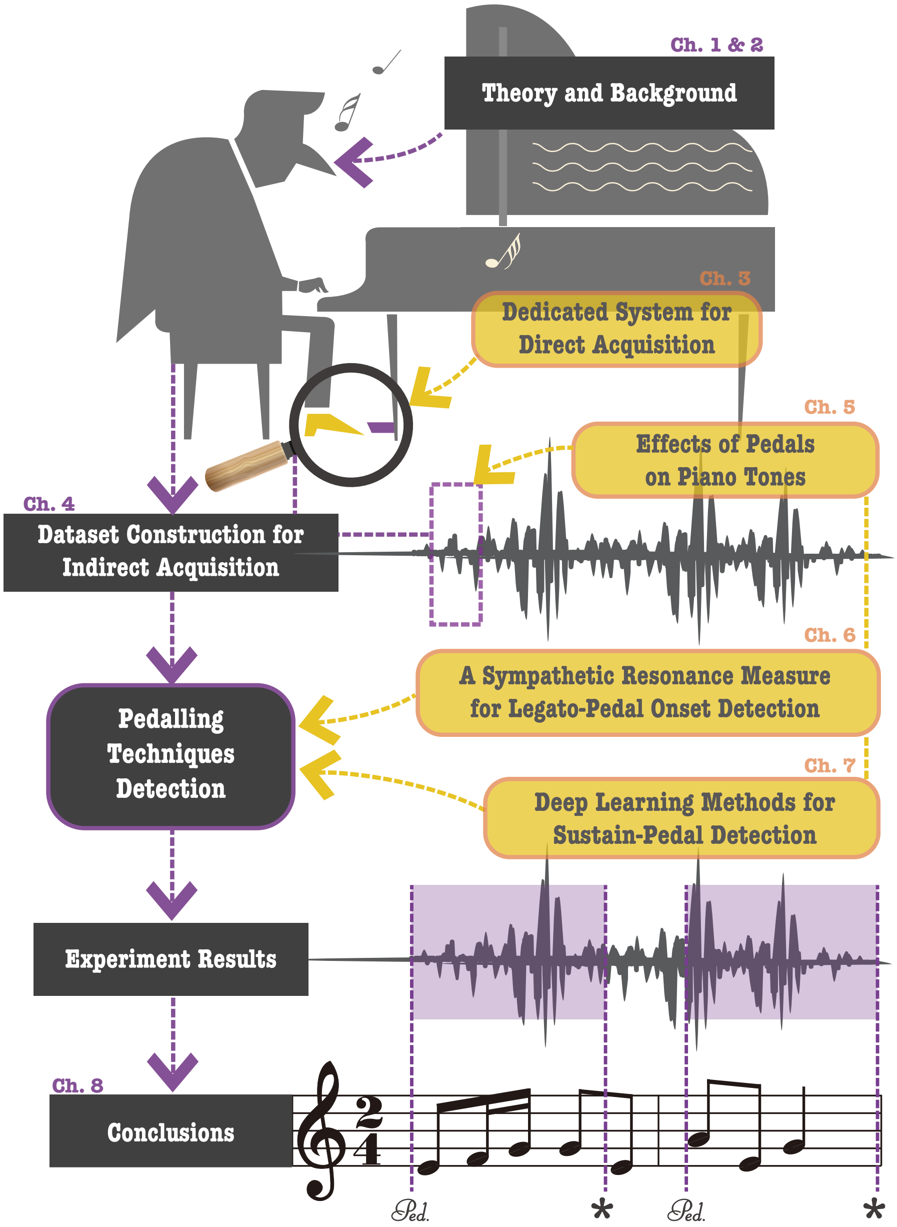

Ch. 1 & 2
Theory and Background
Ch. 3
Dedicated System for Direct Acquisition
Ch. 5
Effects of Pedals on Piano Tones
Ch. 4
Dataset Construction for Indirect Acquisition
Ch. 6
A Sympathetic Resonance Measure for Legato-Pedal Onset Detection
Pedalling Techniques Detection
Ch. 7
Deep Learning Methods for Sustain-Pedal Detection
Experiment Results
Ch. 8
Conclusions
Ped.
*
Ped.
*

Figure 1.4: Outline of the thesis.

recording the pedalling gesture of performers and the piano sound under normal playing conditions. Using the collected gesture data, the task of classifying these data by pedalling techniques is undertaken using signal processing methods and machine learning classifiers. Results can be visualised in an audio-based score following application to show pedalling together with the performer's position in the score. The proposed system obtains high accuracy in classification tasks and can be easily installed on any piano pedals. This allows wider participation compared to other systems, which are often restricted for use in laboratory environments.

Chapter 4 is focused on the three datasets we designed for evaluating indirect acquisition algorithms in this thesis. Using the dedicated system introduced in Chapter 3, how the sustain pedal is played in a piano performance can be automatically annotated. This provides the ground truth for a dataset consisting of acoustic piano recordings. The other two datasets are developed based on MIDI playback under a more controlled recording setup. One is produced using Disklavier to discover the effects of pedals on piano tones. The other one is generated using Pianoteq, providing a large dataset to train deep learning models. These datasets complement the existing piano datasets that are devoted to pitch estimation.

Chapter 5, **6** and **7** propose indirect acquisition methods, which aim to detect pedalling techniques from audio alone. We start with isolated piano tones played with different pedalling techniques. Their spectral and temporal characteristics are discussed and modelled in Chapter 5. Especially the characteristics in the residuals (after the partial components are removed from the original signal) inform us to design features representing sympathetic resonance. These features can be used to indicate the presence of legato pedalling. Therefore legato-pedal onsets become possible to detect in polyphonic piano music using the method proposed in Chapter 6. To facilitate such detection, deep learning models are trained to localise audio frames corresponding to pedalled segments in Chapter 7. The knowledge encoded in the trained models can be further transferred to detect pedalling techniques from audio recorded using other pianos or recording setup.

Chapter 8 concludes this thesis and identifies some directions for future work.

1.5 Associated Publications

This thesis covers the work carried out by the author between September 2015 and April 2019 at Queen Mary University of London. The majority of the work presented in this thesis has been published in peer-reviewed conferences and journals:

- **Direct acquisition:**

  1. **B. Liang**, G. Fazekas, A. McPherson and M. Sandler, "Piano Pedaller: A Measurement System for Classification and Visualisation of Piano Pedalling Techniques," in *Proceedings of the 17th International Conference on New Interfaces for Musical Expression (NIME)*, Copenhagen, Denmark, 2017.

  2. **B. Liang**, G. Fazekas and M. Sandler, "Recognition of Piano Pedalling Techniques Using Gesture Data," in *Proceedings of the 12th International Audio Mostly Conference on Augmented and Participatory Sound and Music Experiences*, London, UK, 2017.

  3. **B. Liang**, G. Fazekas and M. Sandler, "Measurement, Recognition and Visualisation of Piano Pedalling Gestures and Techniques," *Journal of the Audio Engineering Society (JAES)*, volume 66, issue 6, pp. 448-456, 2018.

- **Indirect acquisition:**

  1. **B. Liang**, G. Fazekas and M. Sandler, "Towards the Detection of Piano Pedalling Techniques from Audio Signal," in *Extended Abstracts for the Late-Breaking Demo Session of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, 2017.

  2. **B. Liang**, G. Fazekas and M. Sandler, "Detection of Piano Pedalling Techniques on the Sustain Pedal," in *Proceedings of the 143rd Audio Engineering Society Convention*, New York, USA, 2017.

3. **B. Liang**, G. Fazekas and M. Sandler, "Piano Legato-Pedal Onset Detection Based on a Sympathetic Resonance Measure," in *Proceedings of the 26th European Signal Processing Conference (EUSIPCO)*, Rome, Italy, 2018.

4. **B. Liang**, G. Fazekas and M. Sandler, "Piano Sustain-Pedal Detection Using Convolutional Neural Networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brighton, UK, 2019.

- **Multimodal modelling strategy:**

   1. **B. Liang**, G. Fazekas and M. Sandler, "Transfer Learning for Piano Sustain-Pedal Detection," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)*, Budapest, Hungary, 2019.

Chapter 2

Background and Related Works

2.1 Introduction

This chapter reviews the background, state-of-the-art methods, applications and evaluation approaches for acquiring instrumental gestures and techniques from music performances or audio recordings. Basically we can distinguish two different strategies for data acquisition. We first introduce direct acquisition of instrumental gestures in Section 2.2. The gestures can be captured by various measurement devices and encoded into dedicated formats. The other way is known as indirect acquisition, which is to detect instrumental techniques from the recorded audio signal. This can be approached by signal processing, machine learning and deep learning methods as introduced in Section 2.3. These two strategies can be complementary to each other. In Section 2.4, how to do instrumental technique detection by jointly analysing the sensor and audio data in a multimodal modelling framework is surveyed. Evaluation metrics used for related tasks are introduced in Section 2.4. Finally in Section 2.6, we summarise how the background and related works inform our works in the other chapters of this thesis.

2.2 Direct Acquisition of Instrumental Gestures

2.2.1 Definitions and Applications

Direct acquisition of instrumental gestures involves tracking the main control parameters of an instrument during a music performance, for instance, bow force in violin performance and key velocity in piano performance. These parameters can be directly accessed by digital measurement devices as introduced in Section 2.2.2. Another common approach detailed in Section 2.2.3 is to sense instrumental gestures with hyperinstruments, which are "traditional" musical instruments enhanced with sensors [24]. Typically sensor output needs to be further processed in an embedded system. Multiple streams of instrumental gesture data would arise during the acquisition process. How to represent the gesture data in combination with the musical information is introduced in Section 2.2.4. Direct acquisition of instrumental gestures is essential for applications including:

- **Gesture-sound mappings for interactive music performance**. The New Interfaces for Musical Expression (NIME) community[1] has been advancing this research to aid real-time computer music performance through mapping the gestural data to sound synthesis algorithms [10, 25]. A number of hyperinstruments grow out of the development of gesture-sound mappings. Some representative examples include the E-Sitar [26], the overtone violin [27] and the magnetic resonator piano [28].

- **Quantitative analysis of music performance**. Most musical sounds are the result of the performer's instrumental gestures. Quantitatively assessing these gestures helps to investigate expressiveness in music performance. Overviews of performance analysis techniques with a focus on the quantitative methods can be found in [4, 29–31].

[1] `http://www.nime.org`

- **Construction of a ground truth dataset for evaluating the indirect acquisition algorithms**. The idea of using the direct acquisition to train models for the indirect acquisition was proposed in [32]. The successfully trained model can serve as a "surrogate" sensor that provides gesture information or detects playing techniques from audio signals without using the original sensors, which could be intrusive to the performer to some extent.

The scope of this thesis adopts the idea in the third application for piano pedalling technique detection. In the following sections, we place the emphasis on the direct acquisition of instrumental gestures in piano performances (see [33, 34] for a review of literature on piano touch).

2.2.2 Digital Measurement Devices

The most accurate and commonly used devices for gesture measurement are optoelectronic motion capture systems, such as Vicon and Optotrak. These systems are based on fixed cameras and markers. The motion of the markers can be tracked by the cameras within their capture volume. This has been used to understand mechanisms responsible for motor controls of pianists. Studies are typically performed on finger, hand and arm gestures in piano playing. For instance, temporal control was examined using finger tactile feedback [35], finger motion [36], distinct inter-joint coordination [37], finger kinematics [38] and hand movement efficiency [39]. From the captured gestures, the main differences between concert pianists, piano teachers, and students can be found in the amount of kinetic energy that is used for tone production and extraneous movements [40]. Findings from these gesture analyses also help to raise concerns of overuse syndrome in musicians [41].

Cost and accessibility of the above motion capture systems remain barriers to be used in a more general scenario such as in instrumental lessons. A more affordable option of motion capture is Kinect, which is not restricted to use in laboratory environments.

Kinect can capture pianist movements based on depth sensing. It is markerless but not as accurate as Vicon or Optotrak. Relevant body landmarks such as hand and shoulder positions in piano playing were able to be detected in [42]. This is useful to point out potentially harmful hand postures [43] and help students to correct posture mistakes in virtual piano tutoring [44].

In piano performance, the instrument itself featuring recording and playback functionality also enables instrumental gestures to be captured. Yamaha Disklavier and Bösendorfer CEUS pianos can register key onset/offset, onset velocity, release velocity and movement of three pedals. With this information, pianists' individuality in the performance of five timbral nuances was investigated in [45]. More importantly, the two pianos can provide a fully-automatic and reliable annotation for the generation of datasets, which are needed to develop and evaluate the algorithms for automatic piano transcription. Such datasets include SMD (Saarland Music Data) [46], MAPS (MIDI Aligned Piano Sounds) [47] and MAESTRO (MIDI and Audio Edited for Synchronous TRacks and Organization) [48]. A comparison of the two pianos on their recording and reproducing accuracy can be found in [49].

Since we focus on the pedalling gestures, it is not ideal to use motion capture systems because the infrared light may be blocked by the performer's upper body or the piano. Reflections on the piano surface can introduce more noise into the system. Pedalling gestures are therefore not accurately measured. We opted for a Yamaha Disklavier grand piano to create a dataset in order to investigate different effects of pedals on piano tones (see Section 4.3 for the dataset construction process). Given that many strict specifications that cannot be played by performers are able to be encoded in MIDI files, Disklavier has the ability to playback these MIDI files and help the creation of audio datasets under very controlled settings. There is still a need for the development of a measurement system that can be used on any acoustic pianos and dedicated to pedalling gestures sensing in a non-intrusive way. The following section introduces sensors and embedded systems to facilitate such development.

2.2.3 Sensors and Embedded Systems

Through a comprehensive review of sensors that can measure user interaction in digital musical instruments (DMI), the use of specialised sensors is shown as a significant determinant of classifying musical gestures, allowing for different mappings according to the gesture being performed [50]. Measurements of the key movements using sensors can be traced back to the late 19th century. Binet and Courtier used a rubber tube under the piano keys to determine a continuous key position based on air compression variations [51]. Ortmann reported a more systematic investigation on piano key motion in [52]. Specific velocity profiles of the key can be captured with the help of a tuning fork mounted onto a key. These profiles could be classified into two types of touch: percussive and non-percussive [53].

With the development of integrated circuits, many other sensor modalities have been used to measure piano performance. Inertial sensors, which incorporate accelerometers and gyroscopes, were mounted on pianists' wrists to track arm motion in [54]. With the wearable sensors introduced in [55], the force from hand, wrist, and arm can be measured. However, wearing sensors can be intrusive to pianists, resulting in more or less unrealistic estimates from the music performance. An alternative solution is to augment the piano itself. For example, multi-touch capacitive sensors were added to the surface of each key in order to track the location of finger-key contact in [56].

Minimally intrusive solutions with maximal ecological validity could be optical-based sensing. A Moog PianoBar was modified in [57] to read continuous key position using near-field optical reflectance sensing. This system can be combined with the monocular image-processing based system detailed in [58]. Accordingly, an integrated system enabling the measurement of the small-scale motion of fingers and large-scale movement of hands and arms was developed in [59].

It is noted that none of the above projects considered the inclusion of pedalling techniques as part of gesture sensing. Existing pedal sensing is discrete and only provides on/off information. These problems have motivated us to develop a system that can be portable, self-contained, low-cost and non-intrusive to measure continuous pedalling gesture on any pianos.

To connect multiple sensors and register their data, there have been a number of commercial single-board computers available to use. According to their reliability, performance, and reproducibility, a suitable platform can be selected. Especially for the creation of DMI, the platform should not only provide connectivity to analogue and digital sensors, but also allow on-board audio processing. This is necessary for our pedalling measurement system to synchronously record both the audio and the gesture data. A noteworthy platform is Arduino[2], which has the ability to add expansion boards, enabling fast prototyping. It benefits the applications in music education [60, 61] and the development of hyperinstruments such as the Electrumpet [62] and the Kalichord [63]. However, Arduino is not capable to run audio processing due to its limited central processing unit (CPU). Since Arduino usually handles sensor input communicating via USB-serial, latency and jitter appear because of the serial connection. Raspberry Pi[3] is another popular embedded device with a more powerful CPU. It can be configured to support the audio-oriented environment by the Satellite CCRMA distribution [64]. Latency still exists because the Linux kernel of Raspberry Pi doesn't support real-time processing. This can be addressed by BeagleBone[4], which features a programmable real-time unit (PRU).

Based on a BeagleBone Black with an expansion "cape", Bela[5] was developed with more robust audio performance, which is suitable for building a stand-alone instrument or measurement system. It provides stereo audio input and output, plus several I/O channels with 16-bit analogue-to-digital converters (ADC) and 16-bit digital-to-analogue converters (DAC) for attaching sensors and/or actuators. It combines the resources and advantages of embedded Linux systems with the performance and timing guarantees

[2] `https://www.arduino.cc`
[3] `https://www.raspberrypi.org`
[4] `https://beagleboard.org/bone`
[5] `https://bela.io`

typically available only in dedicated digital signal processing chips and microcontrollers. Consequently, Bela integrates audio processing and sensor connectivity in a single high-performance package for our use. This allows for sampling both digital and analogue input at audio rate, providing jitter-free alignment between audio and sensors. Full technical details can be found in [65, 66]. For these reasons, we integrated Bela into the pedalling measurement system as detailed in Section 3.2, rather than other hybrid microcontroller-plus-computer systems, which typically impose limited sensor bandwidth and may introduce jitter between sensor and audio samples.

2.2.4 Gesture Data Representations

Different gesture data representations have been proposed for storage purposes. There are several motion capture formats that are designed to accompany specific motion capture hardware and focused on full-body motion-capture streams. They are not able to synchronously store other types of data using different resolutions and sampling rates. For a more generic purpose, the Gesture and Motion Signal (GMS) format has been developed for virtual reality multisensory applications in [67]. It provides a binary format for storing low-level sensor data such as position and force. To simultaneously handle a higher level of data such as gesture descriptors, Gesture Description Interchange Format (GDIF) [68] was proposed inspired by Sound Description Interchange Format (SDIF) [69], which was designed to describe properties of audio signals. It is based on existing formats and protocols including XML, Open Sound Control (OSC) and so on. Ideally, GDIF should be possible to store all sorts of data from various systems. A summary of the existing formats representing music-related movement and gesture data can be found in [70].

In the thesis, pedal movement can be represented as a time series, indicating the changes in pedal depth. Considering the simple nature of time series, another two formats instead of the above gesture data formats were used to encode the use of pedals. The two formats are comma-separated values (CSV) and Musical Instrument Digital Interface

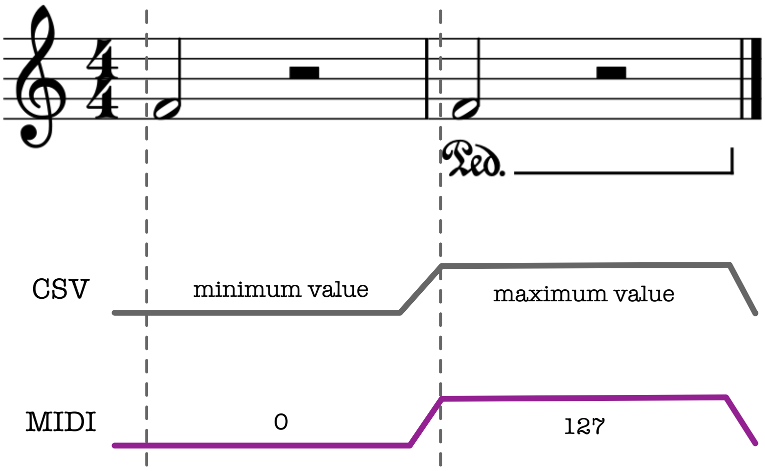


Figure 2.1: Different pedal representations of the same note played without (the first note) or with (the second note) the sustain pedal, including music score, sensor data stored in a CSV file and messages from a MIDI file.

(MIDI). For the pedal movement captured by our measurement system in Chapter 3, it is stored as CSV. Given that the measurement system records the sensor signal at an audio-sample rate using the same master clock, the captured pedal movement can be stored in alignment with audio signals, which are saved in standard pulse code modulation wave files. If pianists perform on Yamaha Disklavier or other pianos with an integrated high-precision MIDI capture and playback system, note events and pedal positions can be recorded as different messages, which are stored in MIDI files. Especially for the sustain pedal, it is handled with the control change message, which consists of a control number of 64 and a control value parameter in the range of $[0, 127]$.

The proposed formats representing the use of pedals can serve as ground truth to evaluate audio-based algorithms of pedalling technique detection. According to a music score with sustain-pedal notations, Figure 2.1 illustrates the corresponding pedal data in CSV and MIDI formats if the performer follows the score.

2.3 Indirect Acquisition for Instrumental Technique Detection

2.3.1 Definitions and Applications

As seen in the previous section, drawbacks arise in developing direct acquisition systems. Certain sensors or devices are required for the system, which could be difficult to develop due to the cost or setup time. The trade-off between accuracy and ecological validity of the system adds more barriers to wider adaptation. Even if an ideal direct acquisition system is developed, usually there is only one version available. This is a common problem in the creation of hyperinstruments, which benefit only one performer who uses the hyperinstrument to acquire data. These problems have motivated researchers to work on indirect acquisition, which is approached by extracting performance-related data from audio signal.

The MIR community has focused on tasks of note-event-related information retrieval, for instance, note onset detection [71], pitch estimation [72], melody extraction [73] and so on. These tasks help to achieve a fully automatic music transcription (AMT) [74] for converting a musical recording into a symbolic representation such as MIDI or music sheet. However, music performance also involves expressive interpretation by the player. Studies on the identification of instrumental playing techniques (IPT) are relatively sparse. IPT can characterise the continuous instrumental gestures which obtain different time-frequency patterns at various scales. Observable patterns that span a certain duration in the time-frequency plot can inform feature design using signal processing methods as introduced in Section 2.3.2. Features that represent IPT characteristics can be used to decide the existence of an IPT by a decision-making mechanism. Machine learning methods are commonly used at this stage. We detail the machine learning algorithms for classification in Section 2.3.3 with a focus on the algorithms used in this thesis. In recent years, deep learning methods have been used to boost the performance for audio-based

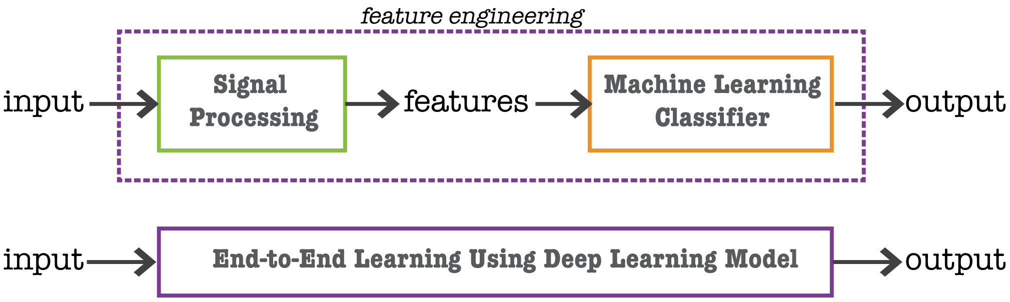

feature engineering
input
Signal Processing
features
Machine Learning Classifier
output
input
End-to-End Learning Using Deep Learning Model
output

Figure 2.2: Diagrams of feature engineering and deep learning approaches.

music classification and tagging [75]. Compared to feature engineering, deep learning allows end-to-end learning with multiple layers combined with nonlinear activations. More technical backgrounds of deep learning are introduced in Section 2.3.4. Figure 2.2 illustrates the main differences in using the above three methods for IPT detection from audio signals.

Automatic identification of IPT is considered as the next milestone in musical instrument recognition in [76]. Recent research has attempted to transcribe IPT on drum [77], erhu [78], guitar [79, 80] and violin [78, 81, 82]. The following sections introduce more details on this literature with respect to the above three methods. We also emphasise the algorithms used for indirect acquisition of piano pedalling techniques in this thesis.

2.3.2 Signal Processing Methods

A number of signal processing techniques can be used for the design of audio features which characterise the time or frequency contexts when an IPT appears. This usually starts with the frequency analysis of an audio signal over time to obtain a time-frequency (TF) representation of the signal. Such two-dimensional representation was initially introduced in [83], where the author demonstrated each point in the representation corresponds to both a limited interval of time and a limited interval of frequency. In computational music analysis, two forms of TF representations are commonly used: *spectrogram* and *Mel-spectrogram*. They provide a visually intuitive content of the input audio signal.

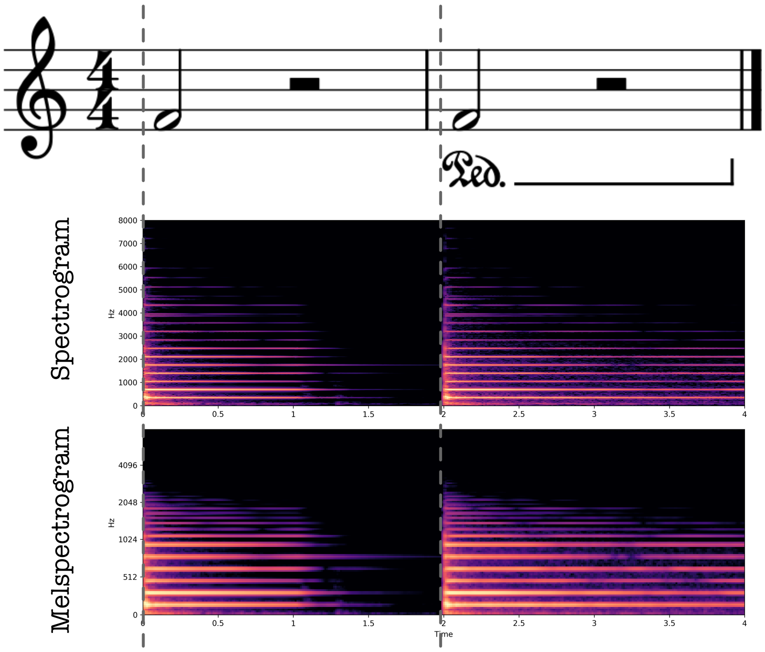


Figure 2.3: Spectrogram (linear frequency) and Mel-spectrogram (Mel-scaled frequency) of the same piano tone played without or with the sustain pedal.

Spectrogram can be obtained by the energy of the short-time Fourier transform (STFT), i.e., its squared modulus. STFT yields frequency information at different frames of a signal. Each frame corresponds to a small section centred around a time instant. This is obtained by multiplying the original signal with a window function. By shifting the window function across time, we can obtain successive frames. Their corresponding frequency information can be computed by the Fourier transform (FT). Therefore the STFT is dependent on not only the signal itself but also the selected window function. A larger size of the window can lead to a better frequency resolution, while reducing the temporal resolution. Considering that the perception of tones in ensemble music is accurate to only 30 to 50 ms [84], the window size is usually set to 5 to 50 ms for analysing music signals. Bell-shaped instead of rectangular window functions such as Hann window (also known as Hanning window) are typically used to reduce the ripple artefacts in the FT of the windowed signal.

There are many variations in visualising a spectrogram to enhance its qualitative properties. In the case of spectrograms for audio signals, the amplitude values are commonly visualised using a decibel scale. This corresponds to the nonlinearity of our ears in sensing sound. Human auditory perception also inspired the design of Mel-spectrogram. The frequency axis of Mel-spectrogram uses Mel-frequencies as calculated in Equation 2.1:

$$f_{mel} = 2595 \log_{10}(1 + f_{Hz}/700), \tag{2.1}$$

where f_{Hz} is frequency in hertz. Compared to spectrogram, Mel-spectrogram is more efficient in size by preserving the most perceptually salient aspects. For this reason, Mel-spectrogram has been widely used for efficient training in deep-learning-based MIR tasks such as music tagging [85] and classification [86]. Figure 2.3 respectively presents the spectrogram and Mel-spectrogram of the same audio signal, which recorded a piano tone played normally and then with the sustain pedal. Similar auditory TF representations include Constant-Q Transform (CQT) [87] and Gammatonegram [88].

As seen in Figure 2.3, TF representations of sounds from pitched instruments mainly consists of sinusoidal and residual components. Sinusoidal components are usually harmonic. Ideally, their frequencies are integer multiples of the fundamental frequency. Residual components contain the energy produced by non-periodic vibration, for instance, from the excitation mechanisms. Therefore the sounds from pitched instruments can be modelled as a sum of a set of sinusoids plus residuals. This musically useful approach was proposed in [89] and the model is denoted as sinusoids plus noise (SpN) model. For the sounds from a specific instrument, physics and acoustics can be taken into account to determine the sinusoidal components. In the case of the piano, sinusoidal components are inharmonic due to the string stiffness. In Chapter 5, we use SpN model to decompose the piano tones played with different pedalling techniques. Audio features can be designed from the sinusoidal and residual components separately to characterise the effects of pedals on piano tones.

Alternatively, models based on non-negative matrix factorisation (NMF) [90] have been applied to the decomposition tasks. NMF has shown comparable performance in automatic music transcription [91–94] and therefore dominated this task during the last two decades. In the context of automatic piano transcription, NMF decomposes the input spectrogram into a product of two non-negative matrices: a dictionary containing the spectral templates that represent spectral energy distribution of 88 notes respectively, and an activation matrix similar to the piano-roll representation that encodes when and how intensely each note is played over time. If the transcription is aimed at a specific piano, the dictionary design can be complemented by different subsets of spectral templates, which represents the attack and decay parts of a piano note, respectively. This further increases the transcription performance as shown in [95, 96]. In Chapter 6, we use the piano transcription method proposed in [96] as an intermediate step to measure the sympathetic resonance based on the residuals.

At this stage, the TF representation after decomposition can emphasise the sinusoidal components only in order to track fundamental frequencies with better performance. The resulting pitch contour can be used as features for the detection of *vibrato* in violin music [78, 81] plus *bend*, *hammer-on*, *pull-off* and *slide* in electric guitar music [80]. For IPT produced by percussive instruments, features extracted from NMF-based activation functions can be used to identify *strike*, *buzz roll*, *flam* and *drag* from drum music [97]. Moreover, low-level temporal/spectral features are usually in combination with the hand-crafted features to form feature vectors (see [98] for an overview of audio features). Finally, the presence of an IPT can be decided with the help of machine-learning classifiers as introduced in the following section.

2.3.3 Machine Learning Methods

With machine learning methods, models can be developed based on sample data (which is known as *training data*) to capture the patterns in order to effectively make decisions or predictions without using explicit instructions. In the training data, a set of data points is denoted as $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_i\}$ and $\boldsymbol{x}_i \in \mathbb{R}^n$ where n corresponds to the dimension of a data point. The associated set of outcomes is denoted as $\{y_1, y_2, ..., y_i\}$. Models trained with *supervised learning* scheme can infer a function to predict y from $\boldsymbol{x}$. While for *unsupervised learning*, training data consists of data points without any target values. The goal becomes to find hidden patterns by *clustering* or *density estimation*.

In this thesis, only supervised learning is used in classification tasks, which aim to assign each input feature vector to one of a finite number of discrete categories. An introduction of the main machine learning methods for classification including Logistic Regression, Support Vector Machine (SVM), Decision Tree (DT) and hidden Markov Model (HMM) is included here as a reference for the following chapters. Because of the scope of this thesis, there are many other classification algorithms such as K-Nearest Neighbours (KNN, first introduced in [99]) and Gaussian naive Bayes (GNB, an extension of naive Bayes [100]) that are not introduced here. A review of classification techniques in the supervised machine learning framework can be found in [101]. When dealing with multi-dimension features, SVM tends to perform better. This is the main reason why SVM is used in different chapters in the thesis. In Chapter 3, SVM also shows its discriminative ability in a multi-class classification task. It obtains better performance compared to HMM, KNN, GNB, DT and Random Forest (RF).

2.3.3.1 Logistic Regression

Logistic Regression is a straightforward method for binary classification problems, which target to label the input with one of the two classes. The core of logistic regression is the sigmoid function. The function adds nonlinearity to the logistic regression and has the ability to map the input value into the range of [0, 1] using Equation 2.2:

$$g(z) = \frac{1}{1+e^{-z}}, \tag{2.2}$$

which is used in logistic regression as Equation 2.3:

$$h_\theta(\boldsymbol{x}) = g(\boldsymbol{\theta}^\mathsf{T}\boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{\theta}^\mathsf{T}\boldsymbol{x}}}, \tag{2.3}$$

where $h_\theta(\boldsymbol{x})$ is the predicted output, $\boldsymbol{\theta}$ is the coefficient for the input $\boldsymbol{x}$. The coefficient can be estimated from the training data with an objective of minimising the cost function $J(.)$. During the training process, $\boldsymbol{\theta}$ is iteratively updated to decrease the value of $J(\boldsymbol{\theta})$. An optimal $\boldsymbol{\theta}$ has been obtained when the value of $J(\boldsymbol{\theta})$ corresponds to its minimum. This error correcting concept is also shared in updating the parameters of deep learning models. For logistic regression, its cost function can be expressed as Equation 2.4 if cross-entropy loss is used:

$$J(\boldsymbol{\theta}) = -\frac{1}{I}\sum_{i=1}^{I}[y_i \log h_\theta(\boldsymbol{x}_i) + (1 - y_i)\log(1 - h_\theta(\boldsymbol{x}_i))], \tag{2.4}$$

where I is the number of the training data points, and y_i is the actual outcome of the i-th input, i.e., $\boldsymbol{x}_i$. To minimise the cost, gradient descent optimisation algorithms can be used (see [102] for an overview of optimisation algorithms). At this point, learning rate is a hyper-parameter we need to determine with caution. It controls how much we should adjust the coefficient of our model with respect to the loss gradient.

When the training is completed, the resulting logistic regression model can output predicted probabilities when a new input is given. The final step is to assign class labels (0 or 1) to the predicted probabilities using a decision boundary. It can be adjusted according to the objective of the prediction, for example, a higher precision or recall (see Section 2.5 for the explanations of the two terms). We use a trained logistic regression model as a binary classifier to determine the existence of piano legato-pedal onset in Chapter 6.

2.3.3.2 Support Vector Machine

Support Vector Machine (SVM) has been introduced for solving pattern recognition problems [103]. It has gained wide popularity as a machine learning method for multi-class classification and regression tasks. This is mainly because of its discriminative ability compared to other machine learning classifiers in many applications.

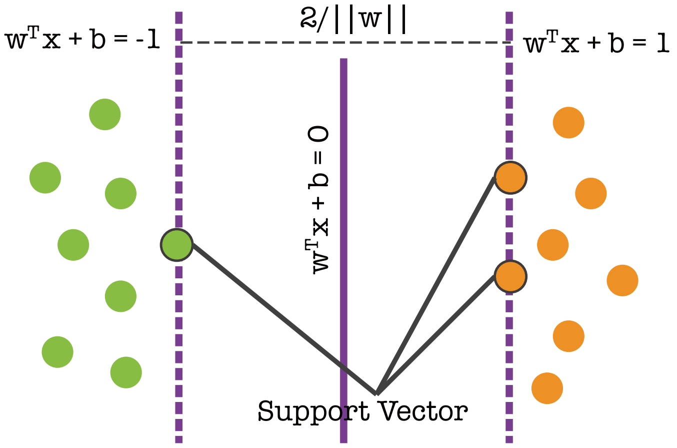


Figure 2.4: A schematic example of SVM and related concepts.

In the training phase, SVM should find a set of hyperplanes that separate the different classes of the training data with the largest distance to the nearest training data points of any class. The distance is known as *margin* ($\boldsymbol{\omega}$) and these nearest data points are *Support Vectors*. Hence SVM is also known as *Maximum Margin Classifier*. Figure 2.4 illustrates these concepts using a two-group classification by SVM. If a data point $\boldsymbol{x}_i$ belongs to a group, then $y_i = -1$. Otherwise y_i is equal to 1. The optimal margin classifier h is formulated as Equation 2.5:

$$h(\boldsymbol{x}) = sign(\boldsymbol{\omega}^\intercal \boldsymbol{x} + b), \tag{2.5}$$

where $\boldsymbol{\omega} \in \mathbb{R}^n$ and the bias term $b \in \mathbb{R}$ are the solutions of the optimisation problem in Equation 2.6:

$$\begin{cases} \underset{\omega,b}{\text{minimise}} & \frac{1}{2}\|\boldsymbol{\omega}\|^2 \\ \text{subject to} & y_i(\boldsymbol{\omega}^\intercal \boldsymbol{x}_i + b) \geqslant 1 \end{cases} \tag{2.6}$$

In the cases that data points are linearly non-separable but nonlinearly separable, SVM can use a kernel function $K_{svm}(\boldsymbol{x}, \boldsymbol{z}) = \phi(\boldsymbol{x})^\intercal \phi(\boldsymbol{z})$, for example, *Radial Basis Function* (RBF). By the mapping ϕ, data points can be mapped into a higher dimensional space, where the data points become linearly separable. However, data points may be still not completely separated by SVM. For solving this, the optimisation problem is reformulated using Equation 2.7:

$$\begin{cases} \underset{\omega,b,\xi}{\text{minimise}} & \frac{1}{2}\|\boldsymbol{\omega}\|^2 + C\sum_{i+1}^{I}\gamma_i \\ \text{subject to} & y_i(\boldsymbol{\omega}^\mathsf{T}\phi(\boldsymbol{x}_i)+b) \geqslant 1-\gamma_i, \end{cases} \tag{2.7}$$

where $C > 0$ is the regularisation parameter that trades off correct classification of training data against maximisation of the margin, and $\gamma_i \geqslant 0$ defines how far the influence of a single training data point reaches. C and γ are the hyper-parameters of RBF-SVM and can be adjusted by cross-validation. SVM was used to classify the IPTs of acoustic guitar [79], electric guitar [80], violin [81] and drum [97]. In this thesis, we use SVM with linear or RBF kernel in Chapter 3, 5 and 7.

2.3.3.3 Decision Tree

Decision Tree (DT) was proposed in [104] and known as Classification and Regression Trees (CART). It also provides a foundation of other algorithms such as Random Forest (RF). DT consists of nodes and branches in a recursive hierarchical structure. With this structure, the correct classification of the training data can be maximised according to the data attributes. Each attribute can be represented as an internal node, which is associated with a test relevant for classification. Classes are represented as leaf nodes. DT branches correspond to each of the possible results. In the testing phase, a new data point can be classified following the nodes and branches to verify the data attributes until it reaches a leaf node. DT has been used to support vibrato detection in [78] using the frequency and amplitude information as two attributes produced by erhu and violin. However, DT is usually not robust to classify input data with a large number of attributes, i.e., high-dimensional data. In Chapter 5, we combine DT with SVM to solve this issue in a multi-class classification task.

2.3.3.4 Hidden Markov Model

Hidden Markov Model (HMM) was initially introduced as a statistical model of a sequence in [105]. It can represent probability distributions over a sequence of *observations* $O = \{O_1, O_2, ..., O_T\}$. Each observation ($O_t$) is dependent on *emission probabilities* of the *state* (Q_t), which is hidden from the observer. Each state in the sequence $Q = \{Q_1, Q_2, ..., Q_T\}$ also has a set of *transition probabilities*, indicating the probability of moving to another state.

There are three fundamental problems that have been approached by HMM[6]:

1. Given observation sequence O, how to compute $P(O|\xi)$, the probability of the observation sequence for the given model with parameters ξ?

2. How to determine the best state sequence Q for the given observation sequence O?

3. How to adjust model parameters ξ to maximise the probability of the observation sequence, i.e., maximising $P(O|\xi)$?

Training HMM as a classifier involves the problem 3. This can be solved by an iterative Expectation-Maximisation (EM) algorithm [106], which is also known as the Baum-Welch algorithm. Model parameters are iteratively re-estimated until $P(O|\xi)$ reaches its local maximum likelihood. Optimal values of state transition and emission probabilities can be obtained.

Using the trained HMM for prediction is in accordance with solving the problem 2. Since different state sequences can produce the same O, the best one should obtain a maximum likelihood. This requires finding the maximum over all possible state sequences. The Viterbi algorithm [107] provides an efficient solution to decode the given O into the optimal Q. This solution to the problem 2 has been successfully in automatic speech recognition, where the audio signal is regarded as O and a string of text is the Q (see [108]

[6] The idea of solving three fundamental problems by HMM was introduced by Jack Ferguson of Institute for Defence Analysis in lectures and writing.

for a tutorial). In Chapter 3, we compare HMM with SVM in the task of recognising pedalling techniques from the gesture data.

2.3.4 Deep Learning Methods

As seen in the previous sections, machine learning methods involve feature design and classifier selection. Feature design requires careful engineering and domain knowledge to design a feature extractor able to transform the raw input into a suitable representation. The selected classifier can then detect the patterns in the representation in order to categorise the raw input. To automatically discover the representations needed for classification, deep learning methods have been making major advances by composing multiple layers. These layers gradually obtain representation at a higher and more abstract level. Deep learning has turned out to be very effective in discovering complex structures in high-dimensional data. It has been successfully applied as the state-of-the-art method in speech recognition, visual object recognition, and many other domains [109]. In this section, we use a feedforward neural network (FNN), the first and simplest type of deep learning model [110], as an example to explain the terminologies used for designing and training deep learning models. An introduction of the convolutional neural network (CNN), one particular type of FNN, and transfer learning framework is also included here as a reference for Chapter 7.

2.3.4.1 Feedforward Neural Network

With the FNN architecture, the trained model can map a fixed-size input to a fixed-size output. In the case of IPT classification, the input can be waveform or TF representations of audio excerpts, and the output can be a probability for each of several categories. Using the architecture and symbols illustrated in Figure 2.5, the forward pass in a neural net

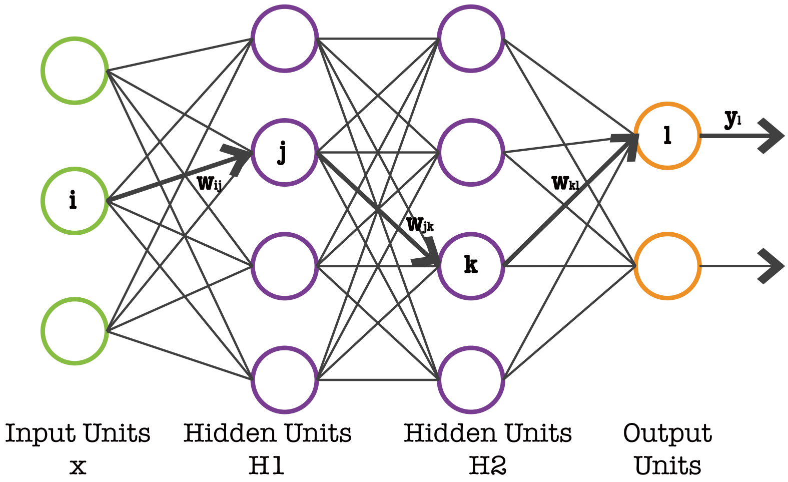


Figure 2.5: A schematic example of FNN with two hidden layers.

with two hidden layers can be computed using Equation 2.8:

$$
\begin{aligned}
y_j &= g(z_j), \quad z_j = \sum_i w_{ij} x_i, && i \in \text{Input} \\
y_k &= g(z_k), \quad z_k = \sum_j w_{jk} y_j, && j \in \text{H1} \\
y_l &= g(z_l), \quad z_l = \sum_k w_{kl} y_k, && k \in \text{H2}
\end{aligned}
\tag{2.8}
$$

where z is the total input to each unit, y is the outputs of the units, and $g(.)$ is the activation function applied to z to get the corresponding y. Bias terms for obtaining z is omitted here for simplicity.

The activation function is analogous to the activation of biological neurons. It is non-linear, enabling the network to learn more intricate patterns. The sigmoid function as previously introduced in logistic regression was used as the activation function in early neural network works. With an increasing number of layers, an issue known as "gradient vanishing" appears [111]. To solve this problem, the Rectified Linear Unit (ReLU) [112] was introduced as an alternative activation function. It can be formulated using Equation 2.9:

$$g(z) = max(0, z) \tag{2.9}$$

It is noted that the activation function for the output layer should have the same output range to the range of the ground truth. For instance, if the ground truth corresponds to a probability, the sigmoid function is suitable because its output is in the range of [0, 1]. If the outputs of a network need to be interpretable as posterior probabilities for a categorical target variable, softmax function is preferred because those outputs can not only range in [0, 1] but also sum to one (see [113, p. 184] for details of softmax used as an activation function).

After designing the FNN, we can train the FNN by adjusting the weights w over multiple layers using backpropagation [114]. This is an iterative process that involves computing the gradient of the loss function with respect to the units in multiple layers using the chain rule. The loss function is decided by the difference between the ground-truth output and the predicted output with respect to the current weights. The goal is to minimise the absolute of difference such that optimal weights are obtained. Furthermore, there are a number of optimisation methods to improve the convergence of backpropagation which suffers from a slow convergence rate and yields suboptimal solutions. For instance, *Adam* optimiser [115] is proposed to compute adaptive learning rates for each parameter.

2.3.4.2 Convolutional Neural Network

Convolutional Neural Network (CNN) is also known as convolutional networks proposed in [116]. It is a particular type of deep FNN that uses convolution in place of general matrix multiplication in the hidden units. The convolution operation is typically denoted using Equation 2.10:

$$convolution(t) = (x * w)(t), \tag{2.10}$$

where the function x is referred to as the *input*, the weighting function w as the *kernel* and the output as the *feature map* in convolutional network terminology. Compared to FNN, CNN is easier to train and generalises better in computer vision tasks. This is

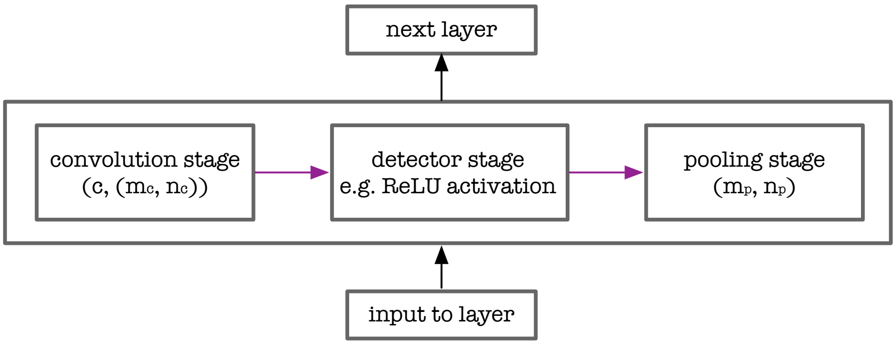


Figure 2.6: Typical components in a layer of spectrogram-based CNN (each stage can be also regarded as a layer).

mainly because the properties of natural signals shed light on the ideas behind CNN, i.e., local connections, shared weights, pooling and the use of many layers. Accordingly, the components of a typical CNN are a series of two types of layers: convolutional layers and pooling layers.

For spectrogram-based CNN in audio-related tasks, Figure 2.6 presents the components as different stages in a typical layer of a CNN with specifications using Keras[7]-style grammar for the sake of clarity. TF representations are commonly used as 2D inputs to layer. Then the convolutional stage is specified by the number of channels (c) and the 2D kernel denoted by its lengths in the frequency (m_c) and time (n_c) axes. This stage performs convolutions in parallel such that a set of linear activations are produced. In the detector stage, each activation is passed through a non-linearity such as a ReLU. In the pooling stage, the output is modified by merging semantically similar features into one. Max-pooling [117] is a common operation that computes the maximum within a neighbourhood. For spectrogram-based CNN, the neighbourhood can be specified by its length in frequency (m_p) and time (n_p). Thereby the representation is reduced in dimension and invariant to small shifts or distortions.

A complete CNN should include a stack of the above three stages, followed by one or more fully-connected layers. Backpropagation with optimisers through the complete CNN is as simple as through an FNN. All the weights are updated, resulting in a trained

[7] `https://keras.io`

CNN. CNN has become the most widely used model for music classification and tagging tasks [75]. It is used in Chapter 7 to train a binary classifier for distinguishing the music excerpts with the sustain-pedal effect.

2.3.4.3 Transfer Learning

FNN, CNN and other deep learning models trained by supervised learning can achieve good performance on many tasks. However, the training process requires extremely large labelled datasets. Meanwhile, test data usually obtain the same feature space and the same distribution as the training data. In real-world scenarios, these conditions may not hold. If a model trained from one domain of interest could be adapted to a new task in another domain of interest, performance of the new task would be maintained by adjusting the pre-trained model. This has motivated the development of transfer learning, which was defined in [118] as "*the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned*". Here the related task that has been learned is denoted as *source task*, and the new task as *target task*. According to different situations between the source and target domains and tasks, there are different categories of transfer learning techniques (see [119] for a survey on transfer learning).

In the field of deep learning, it is more common to tune the pre-trained model from the source task to solve the target task. This is effective because the source task usually used a large corpus to train the model, allowing to make accurate predictions on a large number of classes. The pre-trained model can learn hierarchical feature representations. In other words, features are more generic in the early layers of the pre-trained model and more dependent on the source-task dataset in later layers. Therefore tuning the pre-trained model involves adjusting the weights of the later layers only. This transfer learning technique has been successfully implemented for image classification [120].

Alternatively, as proposed in [121], features can be extracted from a CNN model, which was trained for music tagging in the source task. These features were then used to train SVMs as classifiers, which solved many target tasks, such as genre and vocal/non-vocal classification. For our thesis, we believe the transfer learning strategy is suited to the challenges in detecting the sustain pedal from polyphonic piano music recorded in different acoustic and recording conditions. In Chapter 7, it is used to transfer the knowledge learned from a large synthesised dataset for the sustain-pedal detection from acoustic piano recordings.

2.4 Multimodal Modelling Strategy

Given the fact that music is largely distributed through audio formats, music analysis predominantly focuses on audio signals. However, music performance is multimodal, leading to multifaceted music content existing in different representations [122, 123]. In recent years, with the development of deep learning techniques for action recognition in video, audiovisual analysis of music performances becomes an emerging area to the music signal processing community (see [124] for an overview). Integrating visual modality contributes to extracting the information that is challenging to retrieve from the audio alone. Studies have shown that audiovisual analysis benefits the tasks such as automatic music transcription from string ensembles [125], vibrato detection from string instruments [126] and audio source separation [127]. Apart from video, a huge amount of other music-related data is available, including sheet music, album covers and so on. Due to the rapid growth of music data, there is a need for cross-modal music retrieval and applications to bridge the gap between various music representations (see [128] for an overview of key methodologies).

For the studies on instrumental gestures and techniques, multifaceted content that appears in music performances promotes the development of systems that allows for multimodal recordings. Both direct and indirect acquisitions as discussed in the previous sections are used in order to capture comprehensive parameters, such as timing and dynamics in music performance. For instance, a complete set of bowing parameters in violin performance was extracted in [129]. The parameters were obtained by combining motion capture system, dedicated sensors and recorded sound. This allows detailed studies of musically relevant aspects of bow control and coordination of bowing parameters. A similar combination strategy enables real-time estimates for pitch and bowing technique detection during violin performance as discussed in [130]. This multimodal method has also facilitated the creation of DMI through decoupling the synthesis of an instrument's sound from the instrumental gestures [131] [132].

In this thesis, our datasets detailed in Chapter 4 are composed of three modalities: MIDI data, sensor signals, and audio signals. Inspired by the methods proposed in [32, 82], we propose indirect acquisition methods for pedalling technique detection based on the training of models with a previously recorded dataset of piano performances, which contains synchronised streams of piano audio signals and pedal controls measured with sensors or MIDI devices.

2.5 Evaluation Methods

2.5.1 Evaluation Metrics

To evaluate the performance of our proposed detection methods for the sustain pedal, we use classification evaluation metrics which have been universally used to assess event detection algorithms. Both the ground-truth annotations (y_i) and the estimations ($\hat{y}_i$) are typically binarised with label "0" or "1" to represent if the i-th frame is played without or with the sustain pedal. By default only the positive label is evaluated using precision (P_1), recall (R_1) and F-measure (F_1). They are defined as:

$$
\begin{aligned}
P_1 &= \frac{N_{tp}}{N_{tp} + N_{fp}}, \\
R_1 &= \frac{N_{tp}}{N_{tp} + N_{fn}}, \\
F_1 &= 2 \times \frac{P_1 \times R_1}{P_1 + R_1},
\end{aligned}
\tag{2.11}
$$

where N_{tp}, N_{fp} and N_{fn} are the numbers of *true positives* (TP), *false positives* (FP) and *false negatives* (FN), respectively. A TP is an outcome where the prediction correctly returns the positive class, i.e., $\hat{y}_i = y_i = 1$. A FP is an outcome where the prediction incorrectly returns the positive class, i.e., $\hat{y}_i = 1$ while $y_i = 0$. Similarly, a FN is an outcome where the prediction incorrectly returns the negative class, i.e., $\hat{y}_i = 0$ while $y_i = 1$.

F-measure is a harmonic mean of precision and recall with respect to the positive label. If we also consider the negative label or extend the above binary metrics to multi-class problems, there are a number of ways to average binary metric calculations across the set of classes. Macro-averaged F-measure (F_{macro}) is computed as an arithmetic mean of the per-class F-measures. Macro-averaging gives equal weight to each class. Since our datasets obtain imbalanced occurrence counts of the labels, micro-averaged F-measure (F_{micro}) provides a better overview of the performance. This is because F_{micro} assigns each sample-class pair an equal contribution to the overall metric by counting the total TP, FP and FN across different classes. In this thesis, F_{micro} is opted to represent the overall performance using Equation 2.12:

$$
\begin{aligned}
P_{micro} &= \frac{\sum_{\kappa} N_{tp}^{\kappa}}{\sum_{\kappa} N_{tp}^{\kappa} + \sum_{\kappa} N_{fp}^{\kappa}}, \\
R_{micro} &= \frac{\sum_{\kappa} N_{tp}^{\kappa}}{\sum_{\kappa} N_{tp}^{\kappa} + \sum_{\kappa} N_{fn}^{\kappa}}, \\
F_{micro} &= 2 \times \frac{P_{micro} \times R_{micro}}{P_{micro} + R_{micro}},
\end{aligned}
\tag{2.12}
$$

where κ is the class index.

Apart from classification evaluation metrics that compare the annotations and estimations frame by frame, boundary detection metrics [133] can be used to evaluate the detection of pedalled segment boundaries. This regards an estimated boundary as correctly detecting a ground-truth boundary if it is within a tolerance window, i.e., reasonably temporal distance away from the closest annotated boundary. Each ground-truth boundary can be detected by at most one estimated boundary. Boundary detection performance can be then measured with standard precision, recall, and F-measure.

For the purpose of model comparison, the machine learning community commonly uses AUC-ROC score (or simply AUC, representing area under the receiver operating characteristic curve) [134]. A ROC curve can present the binary classification performance of the models at all classification thresholds based on their corresponding precision and recall. It is plotted with the TP rate on the y-axis against the FP rate on the x-axis. The entire area underneath the ROC curve provides an aggregate measure of performance across all possible classification thresholds. Hence AUC is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. We opt for AUC scores to compare the CNN models which are trained with balanced datasets in Chapter 7. The model with the highest AUC score based on the validation set obtains the best discriminative ability to decide the presence of the sustain pedal.

2.5.2 Cross-Validation

To prepare data for the experiments on machine/deep learning models, a straightforward split is to separate the dataset into training and test set. Then we can fit the model on the training set and measure the model performance on the test set using the evaluation metrics introduced in the previous section. Due to a simple training/test split, the model could be biased to the characteristics of the data points within the training set, leading to over-fitting problems. In this case, we cannot simply assess the quality of the model based on its performance on the test set.

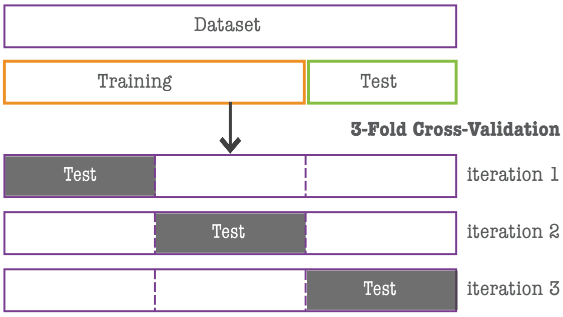


Figure 2.7: Visualisation of the differences between a straightforward train/test split and three-fold cross-validation.

A more robust method for evaluating models is cross-validation. It involves splitting the dataset into k folds of approximately equal size. For each unique fold, it is regarded as a test set. Then the model is trained on the remaining $k - 1$ folds and evaluated on the test set. This procedure is known as *k-fold cross-validation.* We can obtain k samples of model evaluation metrics in the end. The average score across the evaluation metrics from each validation fold can be used as a performance measure of the model. Figure 2.7 presents the behaviour differences between a straightforward train/test split and a three-fold cross-validation. It is noted that if the data in the test set has never been used in cross-validation, the test set is also known as a holdout dataset.

In practice, there are a number of variations on the k-fold cross-validation procedure. One commonly used variation is *stratified shuffle split*, where folds are formed by preserving the percentage of samples for each class. Therefore stratified randomised folds are returned. Another variation we used in this thesis is *leave-k-group-out cross-validation*, where dataset is split into groups according to a domain. For instance, samples can be grouped by which music piece they are associated with. Consequently, each training set consists of all the samples except the ones related to k specific groups.

2.6 Summary

This chapter presented the technical background for this thesis. It first introduced direct acquisition of instrumental gestures with an emphasis on the ones in piano performances. An introduction of some commonly used devices, sensors, embedded systems, and gesture data representations was provided to inform the development of our dedicated system for pedalling gesture measurement in Chapter 3. It also contributed to the construction of the datasets detailed in Chapter 4. In terms of the indirect acquisition, we summarised the existing works on IPT detection and presented signal processing, machine learning and deep learning methods that can be used for pedalling technique detection from audio signals. Multimodal modelling strategy was introduced as a way of jointly using the data of different modalities to improve the performance of indirect acquisition. Finally, we introduced evaluation methods for pedalling technique detection tasks including the metrics and cross-validation schemes.

Chapter 3

Dedicated System for Direct Acquisition

3.1 Introduction

This chapter introduces a novel measurement system dedicated to the direct acquisition of pedalling data. Pedalling gestures can be captured by the measurement system developed in Section 3.2, where the sensor data can be simultaneously recorded alongside the piano sound under normal playing conditions. Using the collected gesture data, a reliable method for pedalling techniques recognition is devised in Section 3.3. This comprises of two separate tasks: pedal onset/offset detection and classification by pedalling techniques. We compared Support Vector Machines (SVM), hidden Markov models (HMM) and other common classification models for the classification task. The recognition results can be represented using novel pedalling notations and visualised in an audio-based score following application described in Section 3.4.

The proposed measurement system can be used to annotate how the sustain pedal is used during piano performance. Thereby the ground truth for a dataset consisting of piano recordings is obtained. This contributes the dataset construction introduced in the following Chapter 4. This chapter incorporates material from our publications: "Piano Pedaller: A Measurement System for Classification and Visualisation of Piano Pedalling Techniques" published in NIME [135], "Recognition of Piano Pedalling Techniques Using Gesture Data" published in *Audio Mostly Conference* [136] and "Measurement, Recognition and Visualisation of Piano Pedalling Gestures and Techniques" published in *Journal of the Audio Engineering Society* [137].

3.2 Measurement System Development

The proposed measurement system synchronously records the pedalling gestures and the piano sound at an audio sampling rate and a high resolution, with the ability to be deployed on common acoustic pianos. Figure 3.1 illustrates the schematic overview of the system, consisting of a sensor and circuit system to collect pedal depth data, as well as an audio recorder and a portable single-board computer to capture both data sources simultaneously.

Near-field optical sensing was used to measure the continuous pedal position with the help of a reflective photomicrosensor (Omron EE-SY1200[1]). This includes an LED and a phototransistor in a compact package. The sensor was mounted in the pedal bearing block, pointing down towards the sustain pedal. This configuration avoids interference with pianists. One of the major considerations in selecting this optical sensor is that its response curve is monotonic within the optimal sensing distance (0.7mm to 5mm) shown in Figure 3.2. As the sustain pedal is pressed so that the pedal-sensor distance is increased, the pedal reflects less of the optical beam projected by the sensor emitter, thus decreasing the amount of optical energy reaching the detector. However, when the sustain pedal is too close to the sensor, the current will drop off. We ensured that the measurement made use of the linear region of the sensor and remained in the optimal sensing range. This was calibrated by measuring the distance between the sensor and

[1] Detailed specifications of Omron EE-SY1200 can be found in `https://omronfs.omron.com/en_US/ecb/products/pdf/en-ee_sy1200.pdf`.

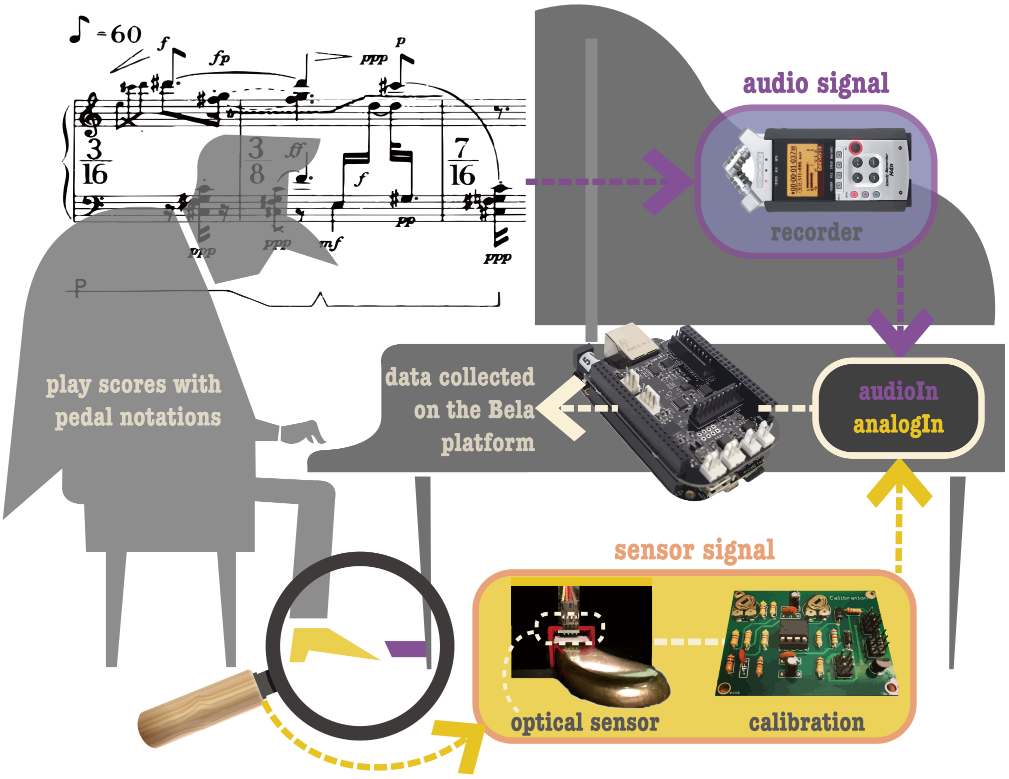


Figure 3.1: Schematic overview of the dedicated measurement system.

the pedal. Then the output voltage of the sensor was amplified and scaled to a suitable range through a custom-built Printed Circuit Board (PCB) which employed a modified version of the circuit described in [57]. Another consideration is the reflectivity of the object being measured. A removable white sticker was affixed on the top of the sustain pedal in order to reflect enough light for the measurement to be robust.

Figure 3.3 shows an overall schematic of the photomicrosensor and PCB circuits. The collector of the phototransistor in the photomicrosensor `OPTO1` attaches directly to the inverting input of an operational amplifier `IC1A`. The voltage at this point is fixed by `IC1A` feedback at V_{ref}, which is equal to 3V produced by resistors `R2` and `R3`. This helps to mitigate the effects of parasitic capacitance in the transistor. `R4` and `R5` set the resting voltage $V_{ref} - (5V - V_{ref})R5/R4 = 0.33V$. `C1` filters high-frequency noise and ensures stability. With this configuration, the output voltage of the circuit is proportional to the incoming light and roughly follows the inverse square of the pedal-sensor distance.

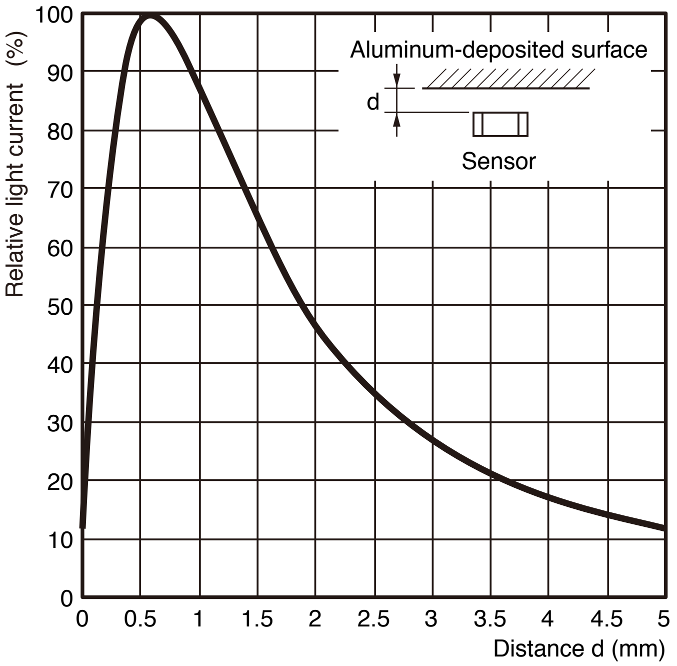


Figure 3.2: Voltage output response curves for EESY1200 using its official datasheet as a reference.

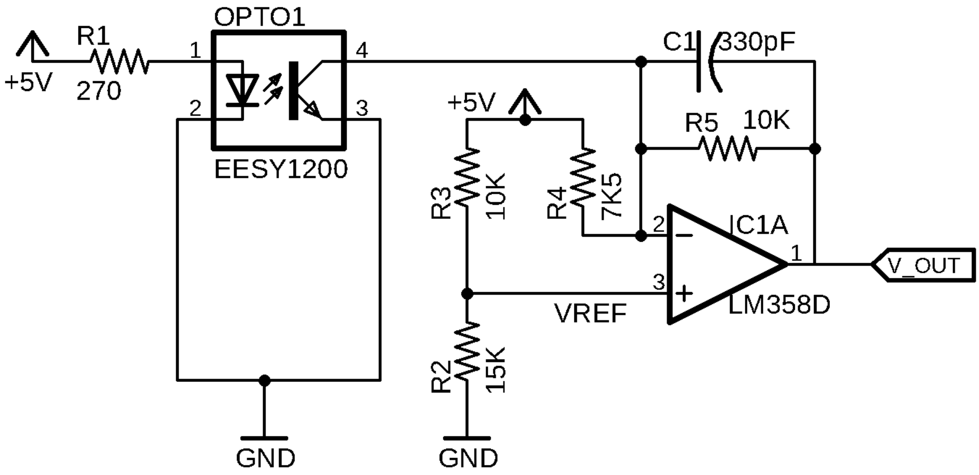


Figure 3.3: Overall schematic of the photomicrosensor and PCB circuits.

The output of the circuit was then recorded using the analogue input of Bela[2], which is an open-source embedded system based on the *BeagleBone Black* single-board computer [138] (see Section 2.2.3 for more details). We opted for using Bela because of the need to synchronously capture audio and sensor data using a high sampling rate with-

[2] `http://bela.io`

out any jitter. The sensor was therefore recorded at 22.05KHz. The piano sound was simultaneously recorded at 44.1kHz on the recorder with high quality and then fed to the audio input of Bela. Finally, both the sensor and audio data were captured with the same master clock and logged into the internal memory of Bela.

3.3 Sensor-Based Recognition

Given the gesture data from the measurement system, methods for pedalling techniques recognition are examined by the dataset introduced in Section 3.3.1. The recognition consists of *when* and *which* technique is employed. "When" refers to the pedal onset and offset times, which can be detected using signal processing algorithms in Section 3.3.2. "Which" refers to the level or class of pedal depth in Section 3.3.3. We aim to classify this into the quarter, half, three-quarter or full pedalling technique. As we mentioned in Section 1.3.2, pianists vary their use of pedalling techniques with the music piece, the acoustics and physics or the piano, and the room acoustics of the performance venue. When any of the above conditions are changed, an automatic adaptation of pedalling techniques is required. Manually setting the thresholds to classify the level of part-pedalling is therefore inefficient. We decided to use supervised learning methods to train SVM or HMM classifiers in a data-driven manner. We employed the *Scikit-learn* [139] and *hmmlearn*[3] libraries to construct our SVM and HMM separately. The performance of these two classifiers are presented in Section 3.3.4.

3.3.1 Dataset

The measurement system was deployed on the sustain pedal of a Yamaha baby grand piano situated in the studios at Queen Mary University of London. Ten well-known passages of Chopin's piano music were selected to form our dataset. These pieces were chosen because of the expressive nature of Chopin's compositions, as well as because

[3] `http://hmmlearn.readthedocs.org`

Table 3-A: The number of pedalling instances annotated in the 10 selected passages from Chopin's music.

Music Passages	**Pedalling Techniques**			
	1/4	1/2	3/4	Full Pedal
Op.10 No.3	14	13	7	5
Op.23 No.1	7	17	8	29
Op.28 No.4	17	24	5	24
Op.28 No.6	9	27	5	17
Op.28 No.7	2	10	3	1
Op.28 No.15	7	34	4	22
Op.28 No.20	9	12	11	17
Op.66	6	21	10	11
Op.69 No.2	2	15	10	24
B.49	3	51	8	17
Sums	76	224	71	167

Chopin was among the first composers to consistently call for the use of pedals in piano pieces.

The author[4] performed the passages using music scores which had been annotated with pedalling techniques by the author in advance. Pedal onset and offset times were marked in several versions of Chopin's published scores. We adopted the version that most publishers accept. In these scores the pedal markings always coincide with the phrase markings. When the sustain pedal is pressed, the suggested pedal depth was also notated. This was roughly in accordance with the dynamics changes and metric accents, since more notes will remain sounding when the key is released in case the sustain pedal is pressed to a deeper level. We provided an annotated Chopin's score as an example in Appendix A.1. Because different techniques may not be used in equal proportion in real world performances, there was no intended coverage of the four different levels of pedal depth. Consequently the number of instances of each pedalling technique in the music passages we recorded remains unbalanced as can be observed in Table 3-A.

The gesture data were labelled every 0.02 seconds according to the notated scores to obtain a basic ground truth dataset. In order to evaluate to what extent the author

[4] The author is an active expert pianist with more than 15 years of classical piano training.

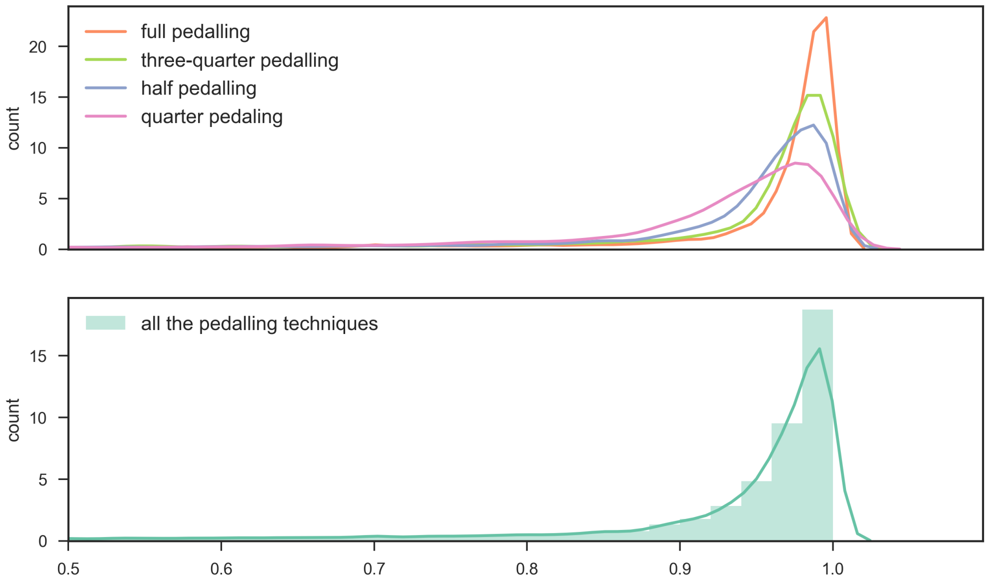


Figure 3.4: Distribution of normalised pedal data within segments labelled with different pedalling techniques.

followed the instructions provided in the scores, we computed descriptive statistics, visualised the data and examined how well it matches the notation. We first grouped the gesture data that were consecutively labelled with the same pedalling technique into one segment. To visualise the data distribution, the gesture data within a segment were normalised to a range of [0, 1]. As seen in Figure 3.4, distribution of the normalised data in segments with the same label of pedalling technique has similar ranges of mean and standard deviation. Therefore mean and standard deviation were extracted to characterise the pedalling technique used within each segment. Figure 3.5 presents the value of the parameters calculated from actual sensor data of each pedalling instance. We can observe fairly well-defined clusters within the data with respect to pedal markings, and also notice that the clusters are approximately linearly separable with some exceptions of half and quarter pedal. We also visually examined the consistency of pedal use with the markings and confirmed that the interpretation of the author was largely consistent with the pedalling notations provided in the notated scores.

The dataset was developed using ten passages played by one pianist on the same piano under the same recording configuration. It was limited in the diversification of

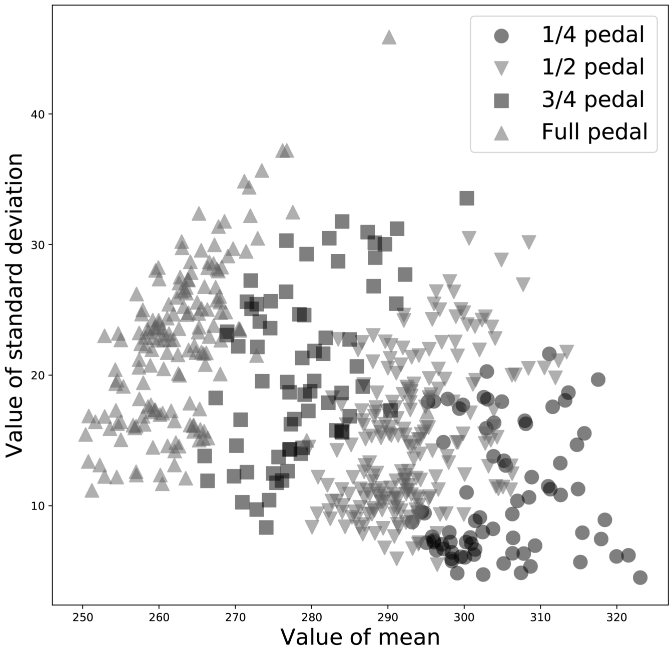


Figure 3.5: Scatter plot of the value of parameters calculated from actual sensor data of each pedalling instance.

data sources. Yet, there are 558 pedalling instances in total to examine the performance of our proposed methods. A good result can indicate that the measurement system is useful to automatically characterise a pianist's pedalling techniques given a specific piano performance environment. For the reproducibility of this study, we made the dataset available online[5], including the annotated scores, associated audio recordings and gesture data, as well as their corresponding labels of pedalling techniques.

3.3.2 Onset and Offset Detection

Figure 3.6 presents the process of segmenting the pedal data using the detected onset and offset times. The value of raw gesture data corresponds to the movement trajectory of the sustain pedal. The smaller the value, the deeper the pedal was pressed. The Savitzky-Golay filter was used to smooth the raw data. It is a particular type of low-pass filter well-adapted for smoothing noisy time series data [140]. The Savitzky-Golay

[5] `http://doi.org/10.5281/zenodo.3237929`

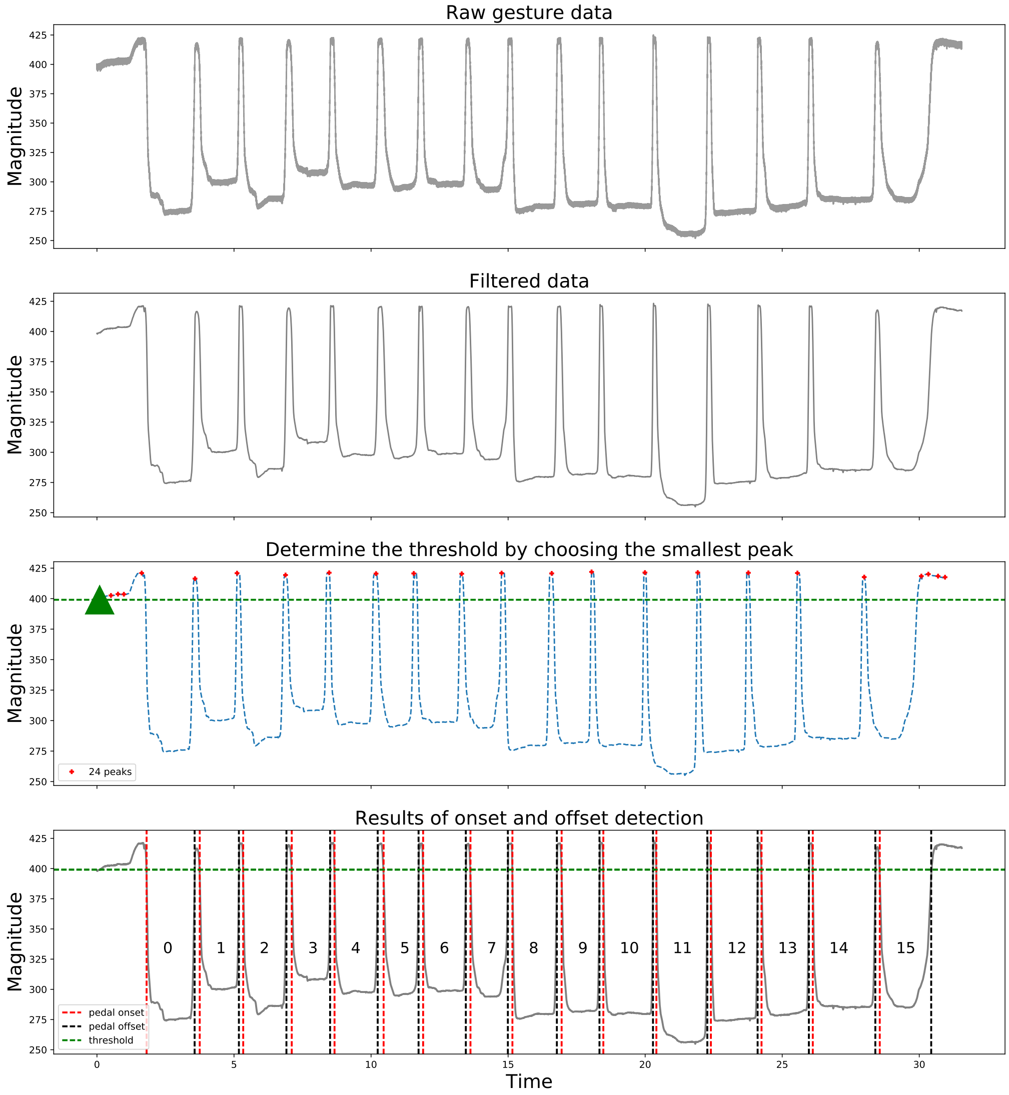


Figure 3.6: Process of pedal onset and offset detection using gesture data.

filter has the advantages of preserving the features of the distribution such as maxima and minima, which are often flattened by other smoothing techniques such as moving average or simple low-pass filtering. Thus it is often used to process time series data collected from sensors such as electrocardiogram processing [141]. Furthermore, filtering could avoid spurious peaks in the signal, which would lead to the false detection of pedalling onsets or offsets.

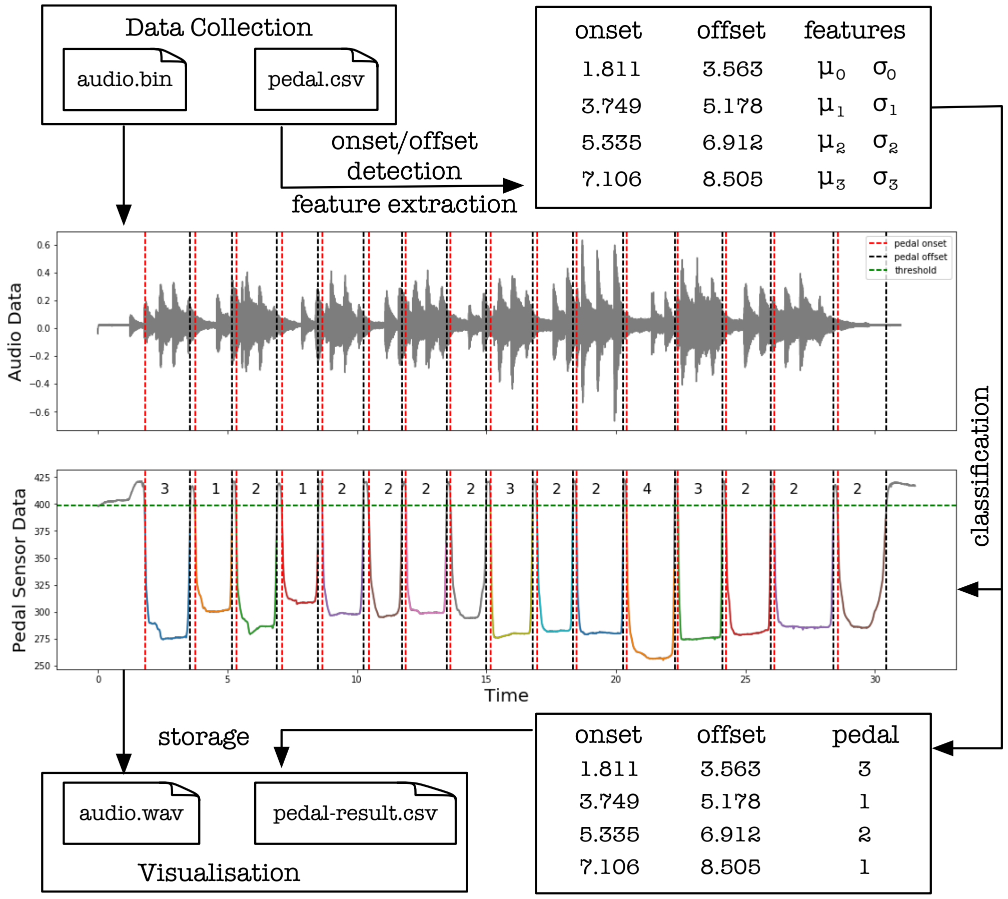
Using the filtered data, pedalling onset and offset times were detected by comparing the data with a threshold (horizontal dashed line in Figure 3.6). This threshold is selected by choosing the minimum value from a peak detection algorithm, i.e., the smallest peak (represented by the triangle in Figure 3.6). Peaks are selected from local maxima in the the filtered data. The smallest peak implies the pedal depth that can distinguish pedal-on/off state by the performer. The moment when the value of data crosses the threshold with a negative slope is considered as the onset time, while a positive slope indicates the offset time. In this manner, each pedalled segment was defined by the gesture data between the onset time and its corresponding offset time. For example, there are 16 segments detected in Figure 3.6. However, the robustness of this method is dependent on the parameter settings used in peak-picking. With a less optimal parameter setting, more false positives of peaks could be returned, leading to an inaccurate selection of the threshold.

3.3.3 Feature Extraction and Classification

Figure 3.7 illustrates the overall classification procedure. After we defined the pedalled segments by the gesture data between the detected onset and offset times, Gaussian parameters were extracted from every segment to aid classification. This was motivated by the observation that the data in each segment largely fits the normal distribution as discussed in Section 3.3.1. Using statistical aggregates as features can not only reduce the dataset size and improve computational efficiency, but also enables a focus on higher level information that represents each instance of pedal use. The statistical features used as input to the classifier were computed based on the Gaussian assumption and parameterised by Equation 3.1, where μ is mean of the distribution and σ is standard deviation.

$$D(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \tag{3.1}$$

To classify the segments using their extracted features, a subset of our dataset was used to train the classifiers. The training data consists of gesture data of pedalled segments with labels of pedalling techniques. Labels 1 to 4 respectively corresponds to



onset	offset	features	
1.811	3.563	μ_0	σ_0
3.749	5.178	μ_1	σ_1
5.335	6.912	μ_2	σ_2
7.106	8.505	μ_3	σ_3

onset	offset	pedal
1.811	3.563	3
3.749	5.178	1
5.335	6.912	2
7.106	8.505	1

Figure 3.7: Process of classifying the gesture data within segments into 4 pedalling techniques.

the quarter, half, three-quarter and full pedalling technique. Although the pedal position is measured in a continuous space, classification of pedalling as discrete events coincides with the interpretation by pianists and may benefit applications such as transcription and visualisation, where discrete symbols corresponding to a recognised technique are easier to read than a continuous pedal depth curve. The recognition results remained synchronised with the audio data. These were then used as the inputs of our visualisation application presented in Section 3.4.

In terms of classifiers, the SVM algorithm was chosen because it was originally devised for classification problems which involve finding the maximum margin hyperplane that separates two classes of data [142]. If the data in the feature space are not linearly separable, they can be projected into a higher dimensional space and converted into a separable problem. More technical backgrounds about SVM were presented in the previous Section 2.3.3.2. For our SVM-based classification, we compared SVMs with different kernels and parameters in order to select one with the best discriminative capacity to categorise the extracted aggregate statistical features into pedalling techniques. SVM essentially learns an optimal threshold for classification from the features in training data, avoiding the use of heuristic threshold and may also account for possible non-linearity in the data.

The second method we employed was an HMM-based classification. As introduced in Section 2.3.3.4, HMM is a statistical model that can be used to describe the sequence of observable events that depend on hidden states which are not directly observable. In our framework, the observations are the features from gesture data and the hidden states are the four pedalling techniques to be classified. In our dataset which consists of Chopin's music, the levels of pedal depth among the segments were changed constantly. We assumed that learning the transition probability of the hidden states could reveal musicological meanings in terms of the extensive use of part-pedalling techniques for an expressive performance. The structure of our HMM was designed as a fully connected model with four states, where states may exhibit self-transition or transition into any of the three other states. Gaussian emissions were used to train the probabilistic parameters. Our HMM-based classification was done by finding the optimal state sequence associated with the given observation sequence. The hidden state sequence that was most probable to have produced a given observation sequence can be computed using Viterbi decoding.

3.3.4 Evaluation Results

Our ground truth dataset introduced in Section 3.3.1 contains labels for the pedal depth denoting the pedalling technique employed within each segment where the pedal is used. The performance of the classifiers was compared using this dataset by conducting leave-one-group-out cross-validation 2.5.2.

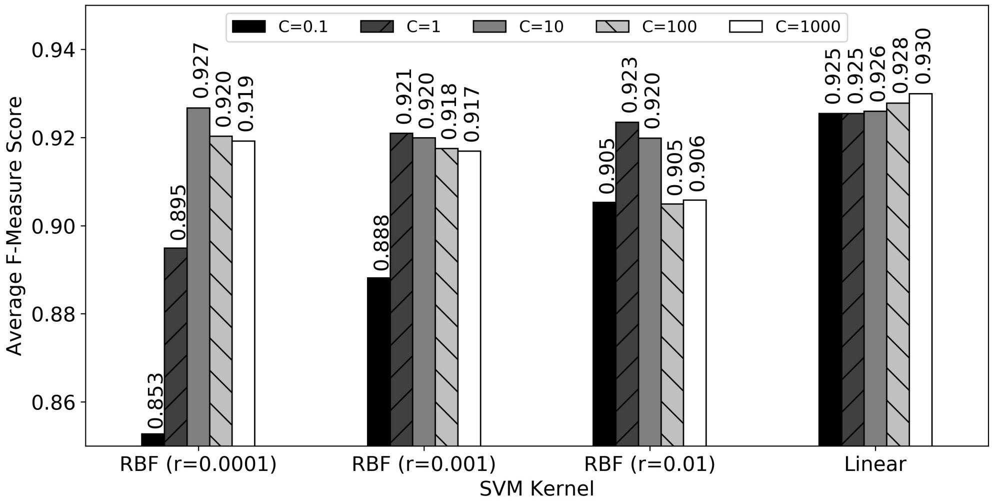


Figure 3.8: Average F-measure score from the leave-one-group-out cross-validation using SVM classifiers with different kernels (RBF and linear) and parameters (γ and C).

Table 3-B: F-measure scores of SVM and HMM from each validation.

Music passages	**HMM F-score**	**SVM F-score**
Op.10 No.3	0.744	0.969
Op.23 No.1	0.902	0.924
Op.28 No.4	0.914	0.976
Op.28 No.6	0.759	0.959
Op.28 No.7	0.688	0.893
Op.28 No.15	0.627	0.943
Op.28 No.20	0.816	0.906
Op.66	0.938	0.970
Op.69 No.2	0.804	0.881
B.49	0.823	0.879
Mean	0.801	0.930

In the leave-one-group-out scheme, samples were grouped in terms of music passages. Classifiers were validated in each music passage where the data need to be classified, while the rest of the passages constitute the training set. Figure 3.8 presents the average F-measure scores for SVM classifiers with different kernels and parameters. The highest score was achieved by a linear-kernel SVM with the regularisation parameter $C = 1000$.

This largely confirms that the pedalling data for most pieces are linearly separable in the feature space we employed. We adopted this SVM model and compared it with HMM. Table 3-B shows the F-measure scores of the evaluation. We can observe that

Table 3-C: Average F-measure scores from two cross-validation strategies using different machine learning techniques.

	KNN	**GNB**	**DT**	**RF**	**SVM**
LTGO	0.916	0.910	0.905	0.910	0.925
SSS	0.941	0.926	0.930	0.944	0.945

SVM outperformed HMM in every music passage, while a mean F-measure score of 0.801 and 0.930 was obtained for the HMM and SVM respectively.

We hypothesise that the lower score of the HMM is resulting from the fact that it was trained in a non-discriminative manner. The HMM parameters were estimated by applying the maximum likelihood approach using the samples from the training set and disregarding the rival classes.

Furthermore, one pedalling technique being followed by a certain another one may be unnecessary or adds very little value when the individual pedal events are separated from each other by long offset phases. For this reason, the learning criterion was not related to factors that may yield improvement of the recognition accuracy directly. While this does not allow us to dismiss potential dependencies between pedalling techniques, our simple HMM model was not able to capture and exploit such dependencies. The reported results can possibly be improved using the hidden Markov SVM proposed in [143] as a discriminative learning technique for labelling sequences based on the combination of the two learning algorithms. An alternative or richer parametrisation of the data instead of Gaussian parameters may also benefit the classification.

To take a detailed look at the SVM-based classification, we present a confusion matrix showing the cross-validation results with the highest average F-measure score in Figure 3.9. It can be observed that the ambiguities between adjacent pedalling techniques can lead to misclassification. In most cases, however, pedalling techniques can be discriminated from one another well.

To avoid a potential over-fitting problem that the leave-one-group-out scheme may cause, we checked the results with two other cross-validation strategies, namely, leave-

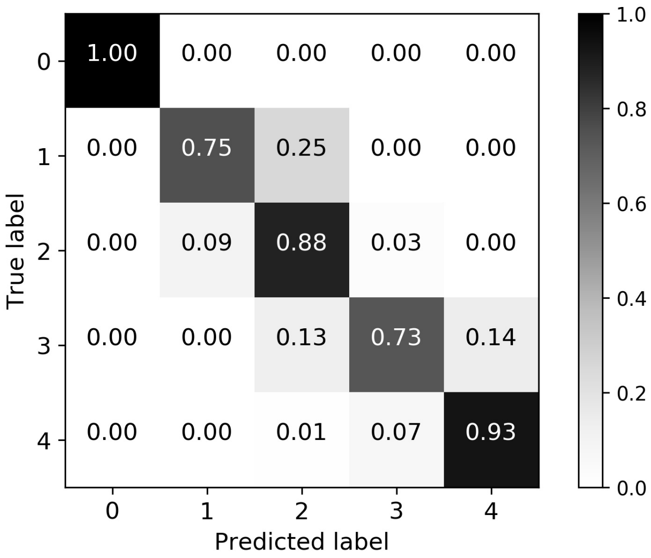


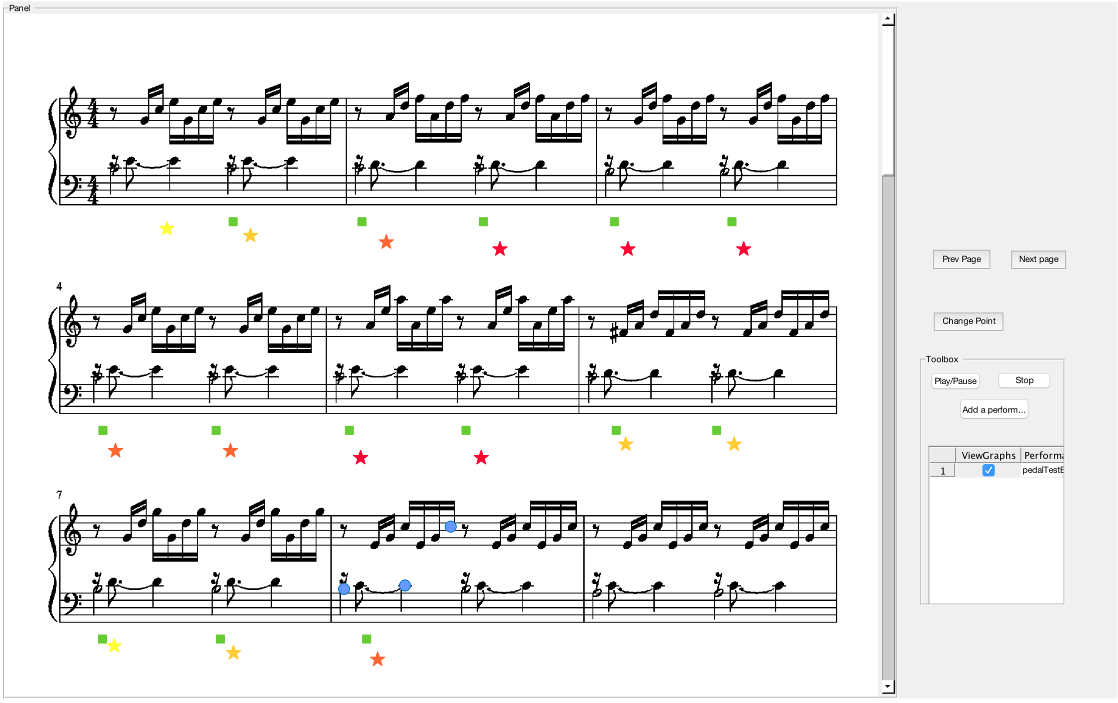
Figure 3.9: Normalised confusion matrix according to the cross-validation results using SVM.

three-group-out (LTGO) and 10-iteration stratified shuffle split (SSS). Introduction of different cross-validation strategies can be seen in the previous Section 2.5.2. For this, the test size was set to 0.3. The SVM model shows a mean F-measure score of 0.925 and 0.945 for these two strategies separately. To confirm SVM's classification ability, we also compared it with a range of common machine learning classifiers, including K-Nearest Neighbours (KNN), Gaussian naive Bayes (GNB), decision tree (DT) and random forest (RF)[6]. The average F-measure scores of these classifiers obtained from the LTGO and SSS cross-validation are presented in Table 3-C. SVM still obtains the highest scores.

3.4 Visualisation Application

In order to demonstrate a practical application of our study, a piano pedalling visualisation application was developed that can present the recognition results in the context of music scores. This may be useful for piano pedagogy or practice as well as musicological

[6] Given the scope of our thesis, not all the machine learning techniques we tested were detailed here. Technical backgrounds of the main techniques were introduced in Section 2.3.3. A review of classification techniques in the supervised machine learning framework can be found in [101].



	ViewGraphs	Performa
1	☑	pedalTestE

Figure 3.10: Screenshot of the visualisation system.

performance studies. We devised a simple notation system for pedalling that indicates pedal depth and timing. The application employed a score following implementation [144] implemented in Matlab, which aligns the music score with the audio recording of the same piece. Asynchronies between the piano melody and the accompaniment were handled by a multi-dimensional variant of the dynamic time warping (DTW) algorithm [145] in order to obtain better alignments. We extended this implementation to align the pedalling recognition results of the same piece, given the detected onset and offset times and the classified pedalling technique.

A screenshot of this system is shown in Figure 3.10. The graphical user interface (GUI) allows the user to select a music score first. After importing the audio recording and the corresponding pedalling recognition results, they can be displayed by clicking the Play/Pause button. The GUI used the following markups for display purposes: blue circles show what notes in the score are sounding aligned with the audio; stars indicate pedal onsets while squares indicate pedal offset. Four different levels of colour saturation plus the vertical location of the star delineate the four pedalling techniques. The levels are increased with the recognised pedal depth class.

The recognition and the score alignment are completed offline so that our visualisation application allows the player to review the pedalling techniques used in a recording. This could be used in music education, for instance guiding students on how to use the pedals in practice after class. We obtained only informal feedback on the application so far. It was suggested that the visualisation should be implemented as a real-time application to enable its use during live piano performance. This could also be used to trigger other visual effects in the performance, as pedalling is partly related to music phrasing. Because of the relatively high latency of the Matlab GUI, it was also recommended to implement our application using another platform.

3.5 Summary

This chapter presents a novel measurement system which was designed to directly capture the pedalling gestures along with the piano sound. The temporal locations of pedalling events were identified using onset and offset detection through signal processing methods. The employed pedalling technique was then recognised using supervised machine learning based classification. SVM- and HMM-based classifiers were trained and compared to assess how well we can separate the data into quarter, half, three-quarter or full pedalling techniques. In our evaluation, SVM outperformed the HMM-based method and achieved an average F-measure score of 0.930. A practical use case was exemplified by our visualisation application, where the recognition results are presented together with the corresponding piano recording in a score following system. Since the system can track how the pedalling techniques were used in a piano recording, it can be used to build a dataset with ground truth in order to facilitate the algorithm design for the detection from audio alone.

Chapter 4

Datasets Construction for Indirect Acquisition

4.1 Introduction

Detection of pedalling techniques from audio recordings is necessary in the cases where installing sensors on the piano is not practical. To evaluate such detection methods presented in the following chapters, this chapter presents the evaluation datasets. Most public annotated piano datasets are constructed for research on multi-pitch estimation [146] or lack isolated-note recordings from the same piano [46]. We therefore built our own datasets, which are constructed using different pianos and motivated by various detection strategies.

The measurement system proposed in the previous Chapter 3 is used to annotate how the sustain pedal is used during piano performance. Accordingly, the ground truth for a dataset consisting of ten well-known passages of Chopin's music is produced in Section 4.2. This dataset is used to evaluate the audio-based detection algorithms, which are regarded as the indirect acquisition of pedalling techniques. To facilitate the design of indirect acquisition algorithms, two other datasets are created from a more

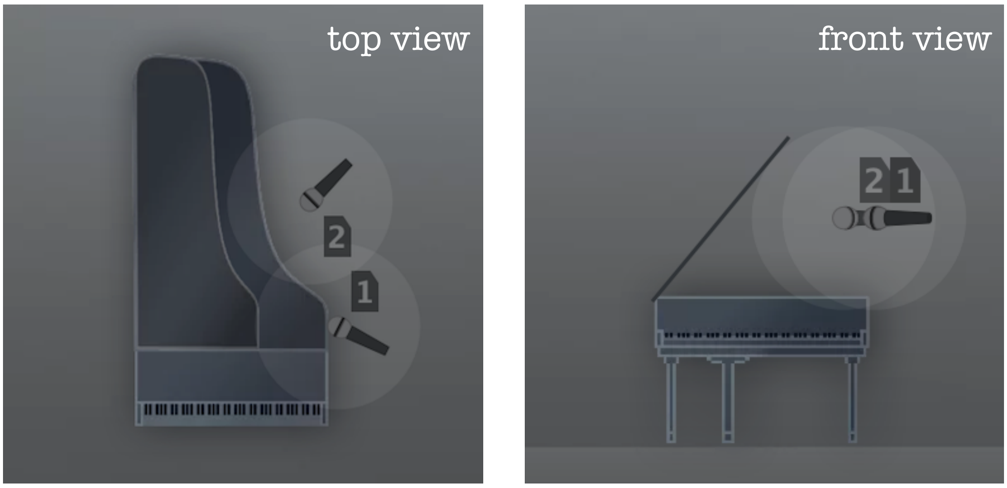


Figure 4.1: Top and front views of microphone positions kept constant during the recording.

controlled recording setting. They are based on MIDI rendering using Disklavier and Pianoteq presented in Section 4.3 and Section 4.4, respectively.

4.2 Acoustic Piano Recording

The same ten passages used in Section 3.3.1 were selected and recorded using better equipment. The author played the passages by her own interpretation instead of using the music scores with specific pedalling annotations provided in advance. This can include a wider range of pedalling techniques. Ten passages were performed using a Yamaha baby grand piano situated in the studios at Queen Mary University of London. The audio was recorded at 44.1 kHz and 24 bits using the spaced-pair stereo microphone technique. A pair of Earthworks QTC40 omnidirectional condenser microphones were positioned about 50 cm above the strings as illustrated in Figure 4.1. The positions were kept constant during the recording.

Meanwhile, movement of the sustain pedal was recorded along with the audio with the help of the measurement system proposed in Chapter 3. After the pedal onset and offset detection for each passage, audio data between the pedal onset time and its

Table 4-A: Occurrence counts of label *on* and *off* at every 0.1 seconds obtained from the ground truth of the dataset consisting of acoustic piano recordings.

Music Passages	**Occurrence Counts**	
	on	*off*
Op.10 No.3	849	268
Op.23 No.1	722	355
Op.28 No.4	995	322
Op.28 No.6	788	289
Op.28 No.7	291	66
Op.28 No.15	611	306
Op.28 No.20	783	274
Op.66	660	197
Op.69 No.2	591	186
B.49	1111	441
Average	740	270

corresponding offset time were annotated with *on*. The rest of the audio data were labelled as *off*. We can accordingly obtain audio data with *on* or *off* labels representing the sustain pedal was pressed or released during the piano performance.

Since it is uncommon to press the sustain pedal for less than 0.1 seconds, audio data were annotated every 0.1 seconds. At this annotation resolution, occurrence counts of label *on* and *off* for each passage are presented in Table 4-A. We can observe that the sustain pedal is widely used for the interpretation of Chopin's music. This dataset is used to evaluate the performance of a transfer learning method using pre-trained deep learning models in the task of localising the pedalled segments from acoustic piano recordings (see Section 7.4.2 for details). We made the dataset available online[1], including music scores of the ten passages, their associated audio data and pedalling annotations.

4.3 Disklavier Rendering

To exploit the piano acoustics with different sustain-pedal effects, a dataset across different tones and velocities is required. It is unrealistic to ask pianists to play tones

[1] `http://doi.org/10.5281/zenodo.3243529`

with various pedalling techniques, while keeping the velocity constant, or vice versa. We encoded specifications for notes with different pedalling conditions in standard MIDI files. A Disklavier, introduced in Section 2.2.2, was used to playback these MIDI files and generate required audio data.

As we discussed in Section 1.3.2, pedalling techniques on the sustain pedal can be varied by pedal-onset timing and pedal depth. The dataset includes ten categories of notes: piano notes played normally without pedal (denoted as *normal* hereafter), notes played with three pedal-onset timings in conjunction with three possible pedal depths (*anticipatory + full*, *anticipatory + three-quarter*, *anticipatory + half*, *rhythmic + full*, *rhythmic + three-quarter*, *rhythmic + half*, *legato + full*, *legato + three-quarter*, *legato + half*). Hence we have three by three conditions for notes played with pedal in addition to the *normal* case. The quarter pedal was not included here because the Disklavier pedal was unstable when rendering notes with the quarter-pedal effect. We recorded individual notes from low to high frequency range of the instrument (from note *E1* to *#G7*) at the velocity of *piano*, *mezzo-forte* and *forte* respectively. The interval between each note is four semitones and each note was played for 2 seconds in the ten configurations described above. This leads to 600 notes in total. Individual recording of each note has note onset time at 0.5 seconds and note offset time at 2.5 seconds.

For notes with the pedal effects, the pedal onset time for *anticipatory*, *rhythmic* and *legato* pedalling techniques is set to 0.5 seconds before, the same time as, and 0.5 seconds after the note onset, respectively. The sounds of repeated notes (same note played repeatedly with an accelerated speed), trills (rapid alternation between two adjacent notes), chords and arpeggio (a group of notes from a chord played one after the other in an ascending/descending order) with different pedalling techniques are also provided in the dataset. Detailed MIDI specifications for rendering the above sounds are presented in Appendix A.2.

The individual note recordings are used in Chapter 5 to design audio features that characterise different pedalling techniques. For the detection of pedalling techniques in

Table 4-B: The number of legato-pedal onsets and duration of each piece.

Piece	**#Pedal Onset**	**Duration** (minutes:seconds)
Beethoven Op.31 No.2-3	84	06:50
Chopin Op.10 No.3	108	04:57
Brahms Op.10 No.1	110	05:08
Ravel Jeux d'eau	88	05:24
Total Number	390	22:19

polyphonic music, we used the same Disklavier to render MIDI files of music pieces. Using the same instrument was informed by the performance improvement in the piano transcription task where a specific piano was used to employ knowledge about the physics and acoustics of the instrument [147]. We believe having access to the recordings of a specific piano's tones is a reasonable assumption for many performance scenarios. Here four well-known piano pieces by different composers were selected, including the third movement of the Piano Sonata No. 17 (Op. 31 No. 2-3) composed by Beethoven in 1801-02, Étude Op. 10 No. 3 composed by Chopin in 1832, the Ballades Op. 10 No. 1 composed by Brahms in 1854, and Jeux d'eau composed by Ravel in 1901. The SMD dataset [46] already had the MIDI files of these four pieces, which were performed by professional pianists on a Disklavier. We used the Disklavier that had been used to render piano tones with different pedalling techniques to obtain audio rendering of the selected four MIDI files.

Audio data of the four pieces are used in Chapter 6 to evaluate the onset detection of legato pedalling. Ground truth can be directly obtained from the corresponding MIDI file. As we introduced in Section 2.2.4, movement of the sustain pedal is represented by integers (0-127) in controller number 64 in MIDI data. We define the sustain-pedal onset happens as the controller value changes from less than 64 to equal to or more than 64. If the sustain pedal is pressed immediately after playing a note, legato pedalling is used and its onset is denoted as *legato-pedal onset*. Table 4-B lists the number of the legato-pedal onset and duration of each piece.

The recording was carried out at the Yamaha recording studio in Milton Keynes, United Kingdom, in March 2017. The instrument was a Yamaha Disklavier grand piano which was tuned directly prior to the recording session. The audio of the piano tones and the four pieces were recorded at a sampling rate of 44.1 kHz and a resolution of 24 bits, using the same microphone setup presented in Section 4.2. Both the MIDI files and their corresponding recordings of Disklavier rendering are available online[2].

4.4 Pianoteq Rendering

In Chapter 7, to train deep learning models, a large dataset is required to capture the acoustic nuances when the sustain pedal is pressed. A dataset consisting of *pedal* and *no-pedal* versions of same music excerpts was constructed. This can facilitate the learning of pedalling-related features that are invariant to note event changes.

To prepare excerpts in pairs, 1567 MIDI files publicly available from the Minnesota International Piano-e-Competition website were downloaded from the competition year 2002, 2004, 2006, 2008, 2009 and 2011[3]. They were recorded using a Yamaha Disklavier piano from the performance of skilled competitors. Considering the amount of MIDI data, to obtain the corresponding high-quality audio in an efficient way, Pianoteq 6 PRO[4] was used instead of recording the Disklavier MIDI playback. It is a physically modelled virtual instrument approved by Steinway & Sons. Audio can be exported with different settings in Pianoteq. We employed the Steinway Model D grand piano instrument and the close-miking recording mode, which is similar to the microphone technique used for our physical recordings[5]. Audio with or without sustain-pedal effect was then generated with a sampling rate of 44.1 kHz and a resolution of 24 bits through preserving or removing the sustain-pedal message in the MIDI.

[2] `http://doi.org/10.5281/zenodo.3242149`

[3] By the time the author built this dataset, only the MIDI files from these six years were available online (`http://www.piano-e-competition.com`). By June 2019, MIDI files of the year 2013, 2014, 2015, 2017 and 2018 were added to the website.

[4] `https://www.pianoteq.com/pianoteq6`

[5] Models for Yamaha pianos are not provided in Pianoteq. Detailed configurations of the close-miking recording mode, such as how far the microphone is away from the piano in centimetres, are not specified in Pianoteq. Therefore we inferred that "similar" microphone technique was applied.

We used audio data generated from the year 2011 Competition as the test set, which includes 175 pieces by 28 different composers from Baroque to Modern period. Data from other years of the competition were used to form the train/validation set, which covers 1392 pieces by 84 different composers. The process of clipping paired excerpts from audio data in the train/validation set are detailed in Section 7.2.1. We use the paired excerpts to train deep learning models as binary classifiers, which can distinguish *pedal* versus *no-pedal* excerpts in the experiments presented in Section 7.3. The trained model can be used to complete the frame-wise detection in Section 7.4, which was evaluated using the pieces in the test set and the dataset introduced in the previous Section 4.2.

In total, over 300 gigabytes of audio data were generated from the MIDI files. Since it is difficult to download this amount of data from online storage, we provide a guide to generate the dataset by Pianoteq in Appendix A.3.

4.5 Summary

This chapter presents three evaluation datasets for the audio-based pedalling technique detection tasks investigated in the following chapters. The first dataset consists of recordings of 10 passages of Chopin's music. These recordings were annotated with *on* or *off* labels, indicating the sustain pedal is pressed or released at every 0.1 seconds.

The other two datasets with paired audio and MIDI recordings were constructed in a more controlled setting. The two datasets help to develop detection methods based on signal processing and deep learning respectively. The first one contains recordings of isolated notes with various pedalling techniques and four pieces by different composers, using the MIDI capture and playback system of a Yamaha Disklavier grand piano. This dataset is designed for discovering the effects of pedals on piano tones in Chapter 5, which informs the methods of pedal onset detection in polyphonic music presented in Chapter 6. The second contains a large number of music pieces in *pedal*/*no-pedal* versions generated by MIDI rendering using a commercial virtual instrument. This dataset is used for training convolutional neural networks to localise the audio frames played with the sustain pedal in Chapter 7. The trained models can be used to extract features adapted to the acoustic characteristics of a real piano in order to enhance the detection performance. This is examined using the 10-passage dataset of Chopin's music.

Chapter 5

Effects of Pedals on Piano Tones

5.1 Introduction

As seen in Section 1.3.2, pedalling techniques can vary in the timing and depth of pedal press. This is especially the case for the sustain pedal. In this chapter, we investigate effects of different pedalling techniques on piano tones. This serves as a starting point for more complex use cases, such as pedalling technique detection in polyphonic piano music.

For the purpose of physics-based piano sound synthesis, there have been studies on the analysis of isolated notes when the sustain pedal is fully pressed. The main features of the sustain-pedal effect were outlined in [2]. They are the decay time, amplitude beating and energy of the residuals, which are obtained through removing the sinusoidal components from the note. The values of these features are increased when the note is played with the sustain pedal. Similarly, the decay of piano tones when half pedalling is used was analysed in [3]. Energy of the residuals was used in [148] to separate notes played with or without the sustain pedal through autoregressive modelling of the estimated residuals and then selecting a threshold to define the two classes.

However, the works mentioned above didn't consider the full spectrum of pedalling techniques, for instance, the effect of pedal timing (see Section 1.3.2 for the details of timing-related techniques). Inspired by the observations in the existing works, we design and exploit audio features based on the analysis of both the partials and residuals of isolated notes without or with different pedalling techniques. To examine the effectiveness of the proposed features, they are used in decision-tree-based support vector machines (DT-SVMs) to classify the notes by pedalling techniques of different timing and depth when the sustain pedal is pressed.

This chapter is organised as follows. The dataset used in this chapter is introduced in Section 5.2. Signal analysis to design audio features and machine learning methods to address the problem of pedalling technique detection on isolated notes are presented in Section 5.3. Experiment is introduced in Section 5.4, including the evaluation results and discussions. Finally, we summarise this chapter in Section 5.5. This chapter incorporates material from "Detection of Piano Pedalling Techniques on the Sustain Pedal" by Liang, Fazekas and Sandler originally published in *Proceedings of the 143rd Audio Engineering Society Convention* [149].

5.2 Dataset

Audio recordings of isolated notes played with different pedalling techniques are used in this chapter. They are from the dataset created using Disklavier as introduced in Section 4.3. The pedalling techniques encoded in the MIDI data for Disklavier to playback represent nine categories, which include three timings (*anticipatory*, *rhythmic* and *legato*) in conjunction with three possible depths (*half*, *three-quarter* and *full*). By observing the temporal and spectral characteristics of these recordings, representative audio features can be designed to distinguish various pedalling techniques.

Because of the physical mechanism of Disklavier itself, audio rendering is not able to accurately reflect all the specifications encoded in the input MIDI files. Both *three-*

Table 5-A: Description of the pedalling techniques considered in this chapter.

Label	**Techniques**	**Descriptions**	**#Notes**
N	*normal*	Notes played without the sustain pedal.	33
NLOH	*non-legato over-half*	Fully or three-quarterly depress the sustain pedal before or at the same time as the note attack.	132
NLH	*non-legato half*	Halfway depress the sustain pedal before or at the same time as the note attack.	66
LOH	*legato over-half*	Fully or three-quarterly depress the sustain pedal after the note attack.	66
LH	*legato half*	Halfway depress the sustain pedal after the note attack.	33

quarter and *full* pedalling lift the damper high enough to prevent it from interacting with the strings. This leads to the same effects of pedal depth on the isolated notes. Hence we consider *three-quarter* and *full* pedalling as the same category designated as *over-half*. Similarly, *anticipatory* and *rhythmic* pedalling have nearly the same timing effects on the notes. We designate *anticipatory* and *rhythmic* pedalling collectively as *non-legato*. Therefore four instead of nine categories of pedalling techniques are obtained.

Meanwhile, the bass strings are attached to a separate bridge, which inhibits the energy from leaking to the middle and treble strings. Thus the bass tones are not altered substantially when they are played with the sustain pedal, compared to the situation where the sustain pedal is not engaged. Likewise, this happens to a part of the treble region. This is because the strings associated with notes higher than *G6* are always free to vibrate because there are no more dampers above these strings. Strings in the middle region are more likely to be affected by pedalling techniques. We therefore focus on modelling the pedal effects of tones from the middle region. These physical configurations and mechanism of the sustain pedal are usually consistent across modern grand pianos. Representative results should be obtained, although tones of one grand piano are used for this study. We believe it is possible to apply our proposed method on other grand pianos.

From the original dataset, 330 recordings representing the notes from the middle

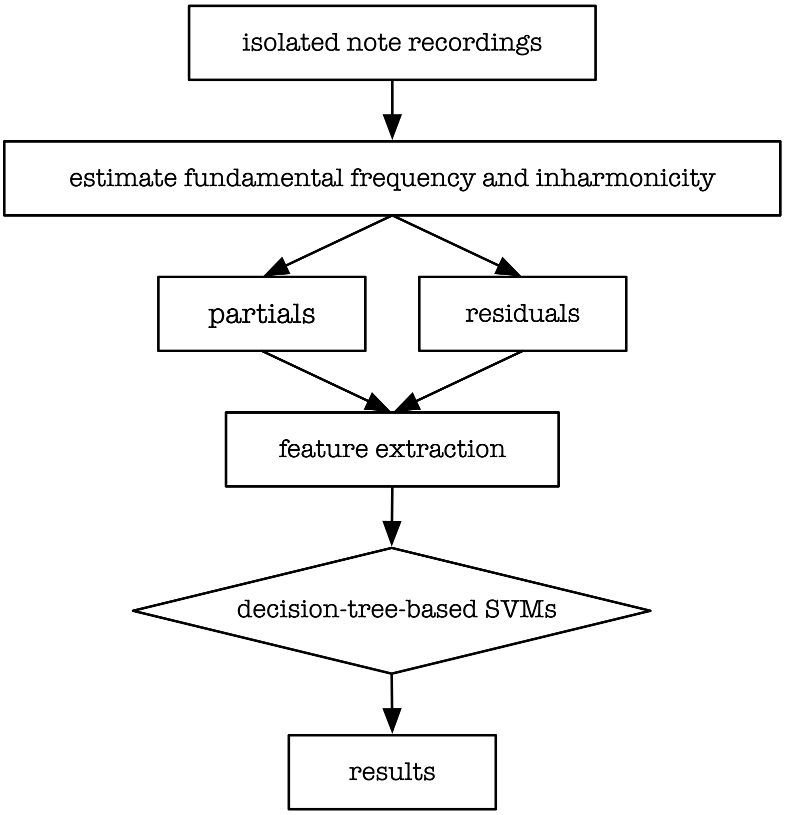


Figure 5.1: Schematic overview of the proposed method for pedalling technique detection informed by the effects of pedals on piano tones.

region of piano (from *C3* to *E6*) are used to form the dataset. They all have the same note onset time at 0.5 seconds and offset time at 2.5 seconds. Table 5-A lists the four pedalling techniques plus notes played without the sustain pedal that are considered in this chapter.

5.3 Methods

Figure 5.1 illustrates the overview of the proposed method which has three main stages. The first stage is decomposition using sinusoids plus noise (SpN) model as introduced in Section 2.3.2. To obtain the sinusoidal components (partials), we first find the frequencies of partials of each note, taking inharmonicity into account. Then the residuals are obtained by subtracting the partials from the original sound. The next stage is feature extraction. We design features through modelling variation in partials and residuals due to pedal use. Finally, DT-SVM is used to classify notes by pedalling techniques. Performance of the classification task can indicate the discriminative power of the proposed features. More details about each step are provided in the following sections.

5.3.1 Partials Plus Residuals Decomposition

5.3.1.1 Partials Estimation

Perfect harmony is characterised by all partial frequencies being integer multiples of the fundamental frequency. Due to string stiffness, partials of piano notes occur at frequencies higher than the expected harmonics. The theoretical partial frequencies can be computed using Equation 5.1:

$$f_p = pF_0\sqrt{1 + Bp^2}, \qquad p \in \mathbb{N} \tag{5.1}$$

where p is the partial index and f_p is the corresponding frequency, F_0 is the fundamental frequency and B is the inharmonicity coefficient. The values of F_0 and B are varied from note to note [13]. To estimate F_0 and B, we implemented the method proposed in [150] with Python code available online[1]. We opted for this method since it compares favourably with some other existing algorithms, such as ones proposed in [151] and [152]. This method has been used to help model the decay of piano sound in [153].

The estimation method employs a non-negative matrix factorisation (NMF) framework as introduced in Section 2.3.2. Each partial of a piano tone is represented using the main lobe magnitude spectrum of a Hanning window. Partials form the spectrum of each note. Such representation serves as a model that is iteratively updated to fit the observed spectrum. The cost function is defined by using the Kullback-Leibler divergence [154]. Moreover, it incorporates inharmonicity into the regularisation term, which

[1] `https://github.com/beiciliang/estimate-f0-inharmonicity`

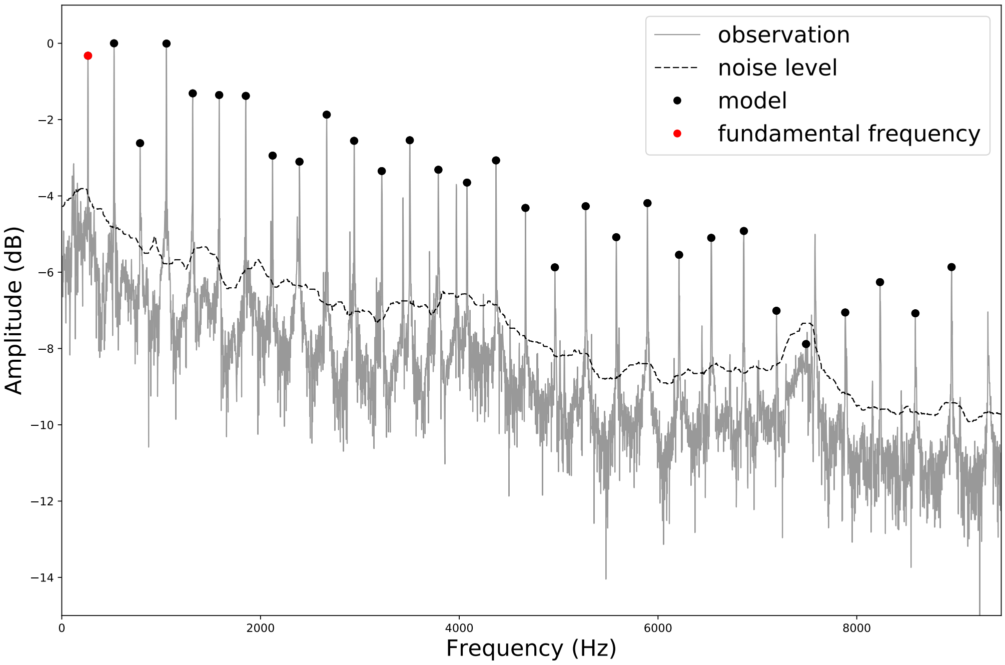


Figure 5.2: First 30 estimated partial frequencies of note *C4* (the fundamental frequency is around 262Hz).

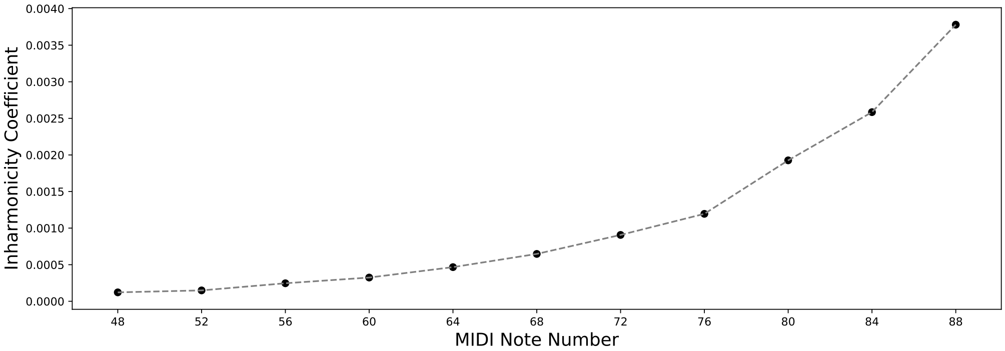


Figure 5.3: Inharmonicity coefficient along the piano compass estimated for the 11 *normal* notes played at the *mezzo-forte* velocity.

is defined as a sum of the mean square error between the estimated partial frequencies from the model and these given by the inharmonicity relation in Equation 5.1. Finally, the central frequency of each partial f_p can be optimised. All of the updated partial frequencies are used to update the inharmonicity coefficient B of the note.

In practice, to avoid most of the potential outliers during the partial selection, there was a pre-processing stage that estimated the noise level adaptive to the magnitude spectrum. This can separate spectral peaks corresponding to the partials from noise. We used the noise level estimation method proposed in [155], which assumes that the noise is generated by a filtered white Gaussian noise. Accordingly, the noise spectral magnitude in a given narrow band should follow a Rayleigh distribution. Parameters of the distribution were decided using a 300Hz median filtering on the magnitude spectrum. Then only partials above the noise level were tracked and optimised. For instance, the dash line in Figure 5.2 represents the estimated adaptive noise level. Finally, the detected frequencies of the first 30 partials of the note *C4* are indicated using black dots, while the fundamental frequency (around 262Hz) is highlighted using a red dot. In Figure 5.3, the estimated inharmonicity coefficient along the piano compass (11 *normal* notes played at the velocity of *mezzo-forte*) is presented.

Based on the above estimation methods, we can obtain a set of partials for each note. We set the value of maximum f_p up to $f_s/3$ where f_s is the sampling frequency (44100Hz). Ideally $f_s/3$ can cover 11 partials of note *E6*, which is the highest pitch in our dataset. Therefore the frequencies of at least 10 partials are obtained for each note in the dataset.

5.3.1.2 Decomposition

As introduced in Section 2.3.2, an efficient modelling technique for music sounds is the sinusoids plus noise (SpN) model [89], which can separate the signal into a sum of a set of sinusoids (partials) plus noise (residuals). A music signal is commonly segmented into short-time frames for analysis due to the time-varying frequencies, amplitudes and phases of the sinusoids. It is also based on the assumption that the sinusoids are constant in a single analysis frame. The SpN model for $s[n]$, the n-th frame of a piano tone signal s, can be formulated as:

$$s[n] = \sum_{p} a_p[n] \cos(2\pi f_p[n]n + \theta_p[n]) + r[n], \tag{5.2}$$

where $a_p[n]$, $f_p[n]$, and $\theta_p[n]$ are the amplitude, frequency, and initial phase of the p-th sinusoid, respectively. These parameters are considered to be fixed within the n-th frame. $r[n]$ is the residual component.

According to the properties of piano tones introduced in Section 1.3.1, the frequencies of the partials for a single piano note are stable. This leads to the partial frequencies can be fixed across frames between the note onset (n_{on}) and corresponding offset (n_{off}). This has been used to facilitate the sinusoidal modelling of piano tones in [156]. We can reformulate Equation 5.2 as:

$$s[n] = \sum_p a_p[n] \cos(2\pi f_p n + \theta_p[n]) + r[n], \qquad n \in [n_{on}, n_{off}] \tag{5.3}$$

Separation driven by partial frequency estimation is typically faster and more robust than the joint estimation of the SpN parameters. Based on this strategy, methods have been proposed for the separation of harmonic sounds in [157] and [158]. In our case, partial frequencies for each note have been estimated in the previous Section 5.3.1.1, which takes the piano inharmonicity into account. Accordingly, the amplitudes and phases for the given frequencies can be estimated from the STFT of the signal s.

Given the estimations of the partial frequencies, amplitudes and phases, the sinusoidal component of a piano tone, i.e., $\sum_p a_p[n] \cos(2\pi f_p n + \theta_p[n])$, can be derived. The residual component $r[n]$ is obtained by subtracting the sinusoids from the original piano tone.

5.3.1.3 Observations

To design effective features for the detection of different pedalling techniques, we observe the evolution of the decomposed partials and residuals separately, using note *C4* played at the same *forte* velocity as an example. For the partials of a piano signal, changes in the first partial are the most representative. Evolution of the first partial can be represented by tracking its amplitude ($a_1[n]$) along the time axis. Figure 5.4 shows the

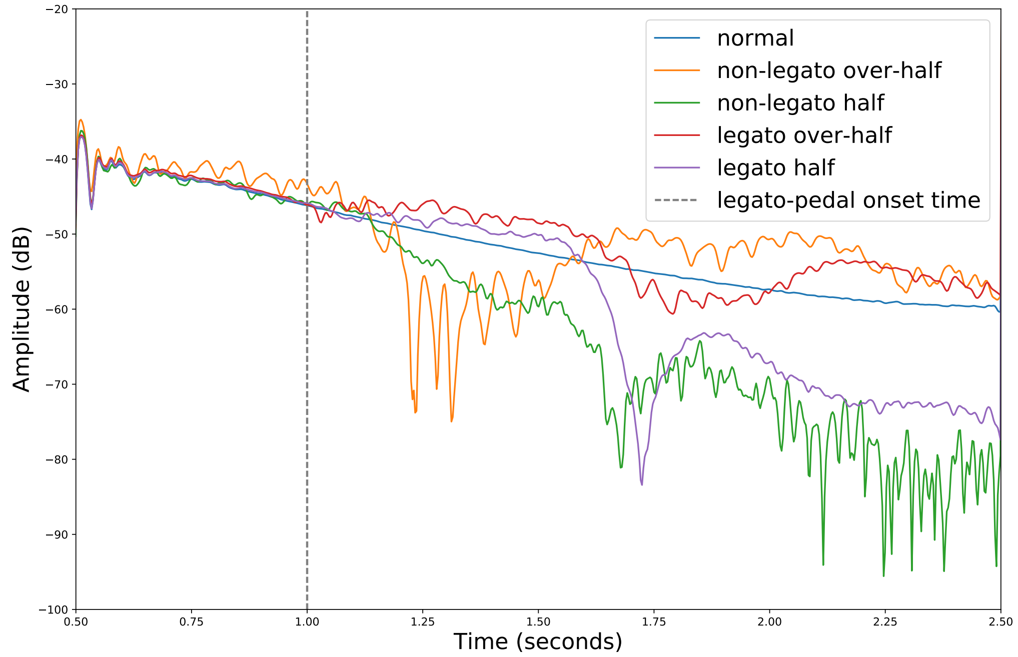


Figure 5.4: Evolution of the amplitude of the first partial for note *C4* played with different pedalling techniques.

evolution of the amplitude of the first partial for note C4 played with different pedalling techniques from the note onset time (0.5 seconds) to the offset time (2.5 seconds).

As seen in Figure 5.4, only the note played normally gradually decays (the blue line). Amplitude beating appears immediately after the pedal is pressed. For instance, notes played with *legato over-half* pedalling (the red line) and *legato half* pedalling (the purple line) obtain a linear decay similar to the blue line during the first 0.5 seconds, and then decay with more beatings after the legato-pedal onset which takes place at 1.0 seconds. For note played with *non-legato over-half* pedalling (the orange line) and *non-legato half* pedalling (the green line), amplitude beating start to appear at the beginning. If we fit the evolution of the first partial using a linear or double decay model, the amount of variability in the model outcomes can reflect the extent of amplitude beatings. This motivates us to design features that can distinguish the note played normally versus played with a pedalling technique.

Residuals of a piano sound mainly consist of the strike of the hammers on the strings

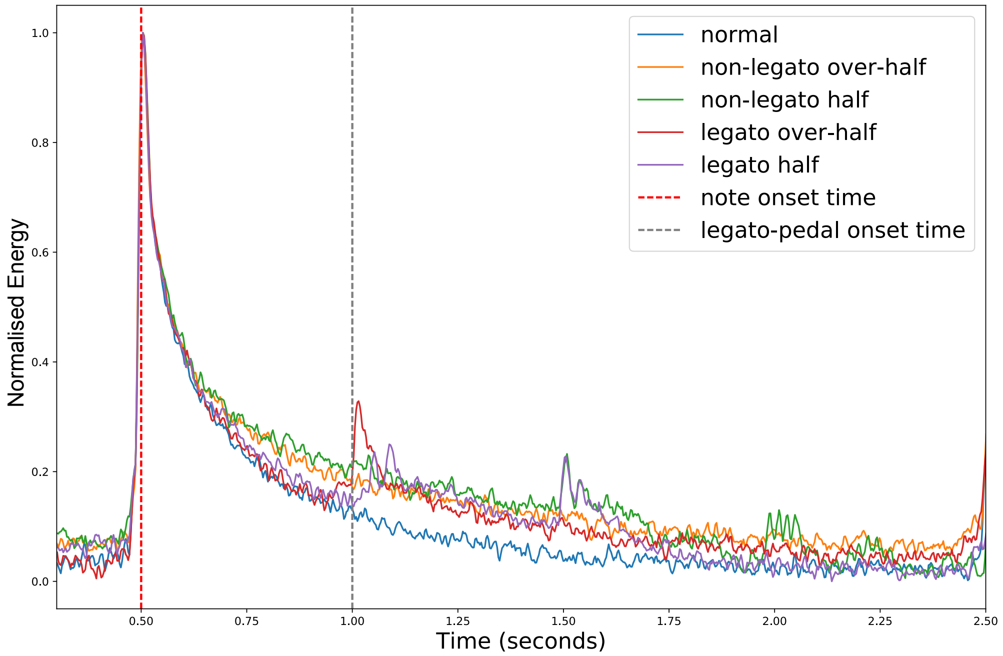


Figure 5.5: Evolution of the normalised RMS energy of the residuals for note *C4* played with different pedalling techniques.

due to the note onset, the interaction between the dampers and the strings, and the resonance of the sound board. This can be reflected on the evolution of the residuals, which is obtained by measuring the root-mean-square (RMS) energy of residuals for each frame using Equation 5.4:

$$RMSE(r[n]) = \sqrt{\frac{1}{L}\sum_{l} |r_n[l]|^2}, \qquad l = 0, ..., L-1, \tag{5.4}$$

where L is the frame length, and l is the sample index within the current n-th frame.

Figure 5.5 shows the evolution of residuals when different pedalling techniques are used to play note *C4*. Each evolution was normalised to the range of [0,1]. The first spike is due to the note onset at 0.5 seconds. Other spikes are led by legato-pedal onset at 1.0 seconds, and by half pedalling that results in more frequent frictions between the dampers and the strings. The evolution after the note onset is roughly subject to exponential decay, which the parameters can be also used to characterise pedalling techniques.

According to the above observations, we can extract features from the evolution of partials and residuals separately in the following section.

5.3.2 Feature Extraction

As introduced in Section 1.3.1, vibrations of the strings result in a variety of decay patterns of piano notes. The decay of partials can be classified into three types: linear decay, double decay and curve decay. For the notes in the middle region of the piano played normally, the decay of the first partial largely fits the type of linear or double decay. To model these two types of decay, we used the multivariate adaptive regression splines algorithm proposed in [159], which combines recursive partitioning and spline fitting for flexible regression. Double decay can be modelled as a two-phase linear regression using Equation 5.5:

$$y_{pd}(t) = \begin{cases} \alpha_{10} + \alpha_{11}t, & t_{on} < t < t_{dp} \\ \alpha_{20} + \alpha_{21}t, & t_{dp} < t < t_{off} \end{cases} \tag{5.5}$$

where y_{pd} is the estimated function modelling the first-partial decay; $\alpha = \{\alpha_{10}, \alpha_{11}, \alpha_{20}, \alpha_{21}\}$ is the vector of regression coefficients; t_{on} and t_{off} are the note onset time and the offset time respectively; and t_{dp} is the demarcation point which satisfies the linear constrain expressed in Equation 5.6:

$$\alpha_{10} + \alpha_{11}t_{dp} = \alpha_{20} + \alpha_{21}t_{dp} \tag{5.6}$$

To check the value of the regression fit, the coefficient of determination is a useful statistic. It was first introduced in [160] and denoted as R^2 in statistics. It quantifies the amount of variability in the outcomes that are replicated by the regression model. A measure of how well y_{pd} fits the observed data can be provided: the better the model fits the data, the closer the value of R^2 is to 1. In this context, R^2 is defined in Equation 5.7 as follows:

$$R^2 = 1 - \frac{\sum_n (y_{pd}[n] - a_1[n])^2}{\sum_n (a_1[n] - \bar{a}_1)^2}, \tag{5.7}$$

where $y_{pd}[n]$ is the modelled outcome at the n-th frame and $\bar{a}_1$ is the mean of the observations $a_1[n]$. We calculated R^2 using y_{pd} that models the first partial evolution as a linear and a double decay respectively, and then retained the R^2 with a larger value. More amplitude beatings occur when a note is played with the sustain pedal as indicated in Figure 5.4. This leads to a lower value of R^2 compared to the value obtained using the note played without the sustain pedal. Therefore R^2 is applied as one of the features to distinguish notes played with or without the sustain pedal.

For the residual components, feature extraction starts with modelling the evolution of $RMSE(r[n])$ between the note onset and offset time as an exponential decay using Equation 5.8:

$$y_{rd}(t) = a_{rd}e^{-\lambda t} + b_{rd}, \qquad t_{on} < t < t_{off} \tag{5.8}$$

where y_{rd} is the estimated function modelling the residual decay; a_{rd}, λ and b_{rd} are the coefficients obtained using non-linear least-squares fitting [161]. The value of λ and b_{rd} can be changed significantly with various pedalling techniques as indicated in Figure 5.5. They were thereby selected as features. The chi-square statistic χ^2_{rd} was calculated to measure the goodness of the exponential-decay fit by Equation 5.9:

$$\chi^2_{rd} = \sum_n \frac{(RMSE(r[n] - y_{rd}[n]))^2}{y_{rd}[n]} \tag{5.9}$$

The value of χ^2_{rd} is larger in the cases of notes played with half or legato pedalling, which reveals more spikes in the evolution of $RMSE(r[n])$. Therefore χ^2_{rd} can be useful for detecting the *non-legato over-half* pedalling technique.

At this point, more features should be extracted to refine the detection of the pedalling techniques including *non-legato half*, *legato half* and *legato over-half*. For this purpose, the differences between the fitted exponential decay model and the observed data of residuals were used. A peak detection algorithm employed the differences to obtain the number of peaks N_{peak}, which aims to facilitate the detection of legato pedalling. This is based on the evident spike led by the legato-pedal onset shown as the red line around 1.0

seconds in Figure 5.5. A fast Fourier transform (FFT) algorithm also operated on the differences, from which the value of the frequency F_{diff} with the maximum amplitude was used to promote the detection of half pedalling. This is effective because more frequent frictions between piano dampers and strings appear when half pedalling is used, resulting in more oscillations in the differences and the increasing value of F_{diff}.

To sum up, six features (R^2, λ, b_{rd}, χ^2_{rd}, N_{peak}, F_{diff}) can be extracted from each note. They are used for detecting whether a note is played normally or not initially. If so, the data of $a_1[n]$ during the first 500ms, i.e., before the legato-pedal onset time, will be saved as a_1^{ref}. This could inform the pedalling detection for rest instances of the same pitch by calculating the chi-squared $\chi^2_{a_1}$ between their observed data $a_1[0:500ms]$ and a_1^{ref}. In the case of legato pedalling, $a_1[0:500ms]$ are more similar with the a_1^{ref} compared with the cases of *non-legato* pedalling according to the evolution before the legato-pedal onset in Figure 5.4. Thus the value of $\chi^2_{a_1}$ is smaller for the notes played with legato pedalling. After the *normal* notes are distinguished, $\chi^2_{a_1}$ is added as an additional feature for further pedalling technique detection.

5.3.3 Decision-Tree-Based Support Vector Machines

As introduced in Chapter 2, decision tree (DT) is a non-parametric supervised learning method which can be used for classification. Its goal is to create a tree-like model that predicts the value of a target variable by learning simple decision rules inferred from the data features [162]. Support vector machines (SVMs) were originally designed for binary classification [163]. For multi-class problems, k classes are separated through building k one-versus-rest classifiers. This leads to the existence of unclassifiable regions. For example, the shaded regions in Figure 5.6 cannot be classified into any of the three classes using the decision function $f_k(x)$ learned from SVMs, where $k = 1, ..., K$.

We can combine DT and SVMs to create decision-tree-based support vector machines (DT-SVMs) for solving multi-class problems. DT-SVMs have been proved to effectively

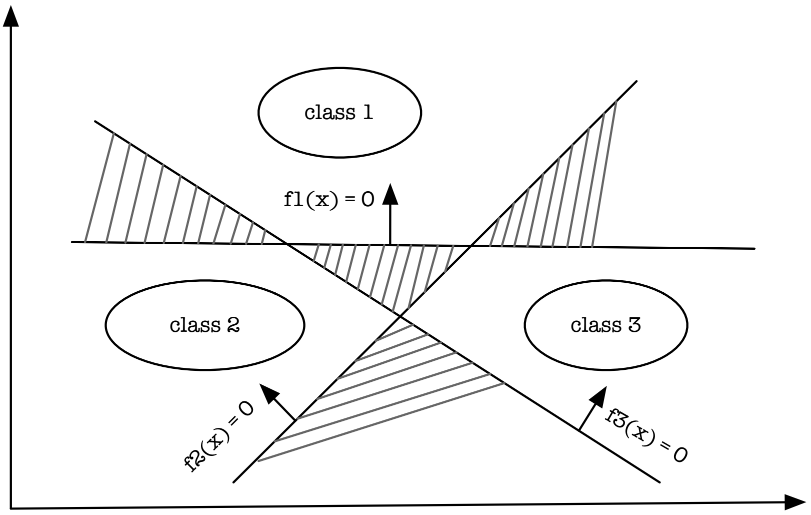
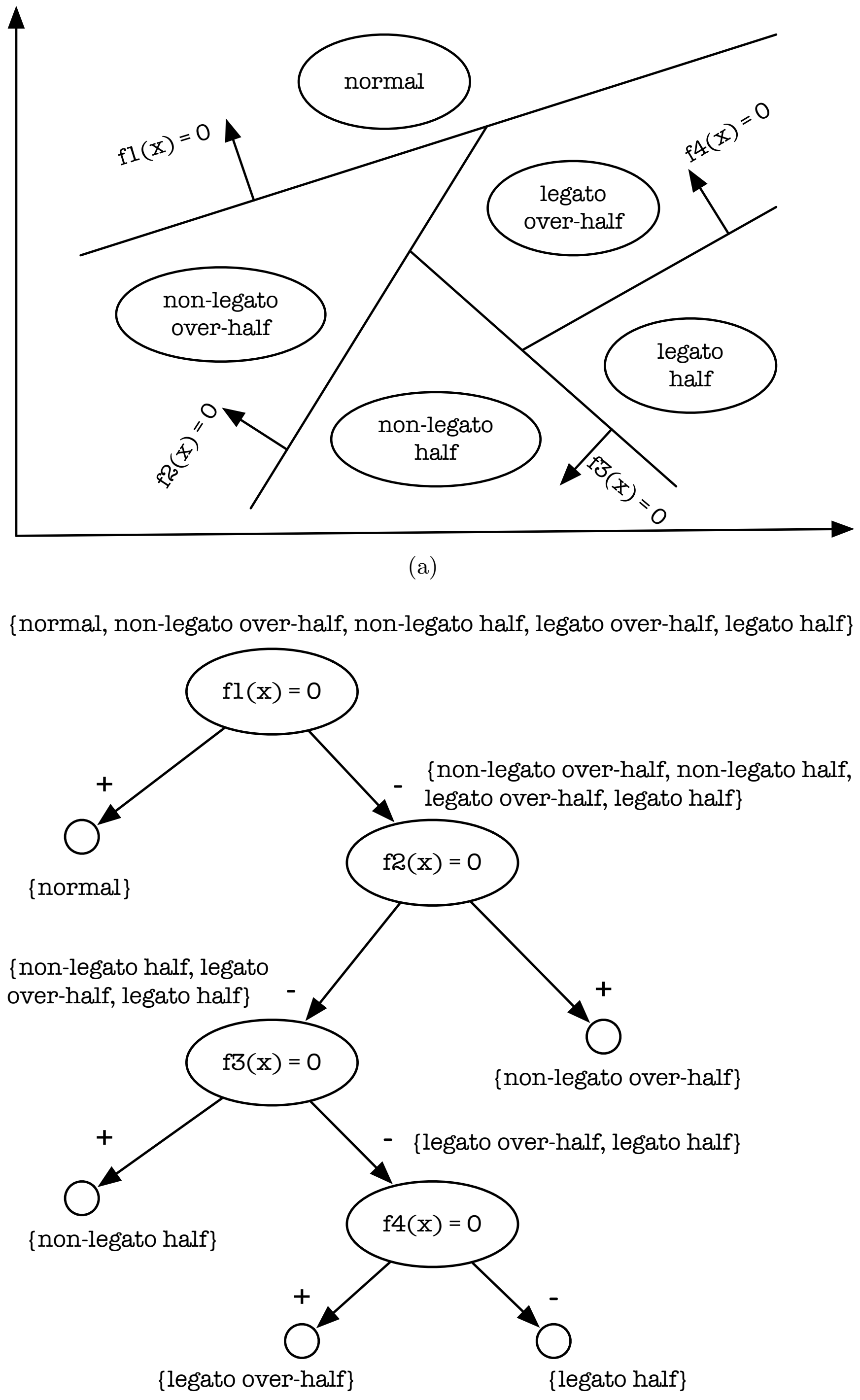


Figure 5.6: The existence of unclassifiable regions using SVMs.

resolve the existence of unclassifiable regions and obtain a higher generalisation capacity than using conventional SVMs or DT in [164]. The structure of DT-SVMs also fits our proposed idea of recursively separating a pedalling technique from others. The classes that are easily bifurcated need to be split at the upper node of the decision tree. In our case, we firstly determined the hyperplane that separates *normal* notes from the notes played with the sustain pedal in the feature space. If multiple classes of pedalling techniques are in the separated subspace, another hyperplane that separates the classes will be determined. This procedure is repeated until there is only one class remains in the separated region. Figure 5.7(a) shows a hypothetical division of the feature space for our pedalling technique classification, and Figure 5.7(b) expresses this using the decision tree.

5.4 Experiments

In the experiment, assessment of the devised features is presented and the proposed DT-SVM model is compared with different classifiers in the pedalling technique classification task. We first detail the configurations to set up the experiment, then present and discuss the experiment results.


normal
f1(x) = 0
legato over-half
f4(x) = 0
non-legato over-half
legato half
f2(x) = 0
non-legato half
f3(x) = 0
{normal, non-legato over-half, non-legato half, legato over-half, legato half}
f1(x) = 0
+
-
{non-legato over-half, non-legato half, legato over-half, legato half}
{normal}
f2(x) = 0
{non-legato half, legato over-half, legato half}
-
+
f3(x) = 0
{non-legato over-half}
+
-
{legato over-half, legato half}
{non-legato half}
f4(x) = 0
+
-
{legato over-half}
{legato half}

(a)

(b)

Figure 5.7: Schematic structure of the proposed DT-SVMs, including (a) hypothetical division of the feature space and (b) expression by the decision tree.

Table 5-B: The number of note instances in mid-lower and mid-upper regions for cross-validation and testing.

Split	**Regions**	
	mid-lower	mid-upper
Train	45	45
Valid	15	15
Test	30	15

Note: the number of note instances in the train/valid split here is referred to as the partition for each fold in the cross-validation.

5.4.1 Experimental Setup

Based on the physical structure of the Disklavier piano, the stress bar between the note *B4* and *C5* can separate middle-register notes into two regions (see Figure 1.1 as a reference). The two regions are denoted as *mid-lower* (from note *C3* to *B4*) and *mid-upper* (from note *C5* to *E6*). Using the dataset introduced in Section 5.2, the total number of note instances in the two regions is 180 and 150, respectively. Various pedalling techniques can lead to subtle acoustic differences between the two regions. Accordingly, extracted features that represent effects of pedals on piano tones of mid-lower region are unlike the ones from the mid-upper region. We constructed classifiers for the two regions respectively.

To present the effectiveness of our proposed DT-SVM classifier, we considered a multi-class SVM [165] with Radial Basis Function (RBF) kernel (designated as C-SVM) and a DT for comparison. An eight-fold cross-validation scheme was adopted for the evaluation of these classifiers (see Section 2.5.2 for more details on k-fold cross-validation). During the testing phase, the learned classifier selected from the one with the best validation performance can output the detected pedalling technique for the new notes. We used the *Scikit-learn* library [139] to construct the classifiers. All features were scaled into the range of [0, 1] using the standard min-max scaling approach.

As seen in Table 5-A, the number of note instances are unequal between different labels. To balance the distribution of notes over the key velocity and the labels, Table 5-B

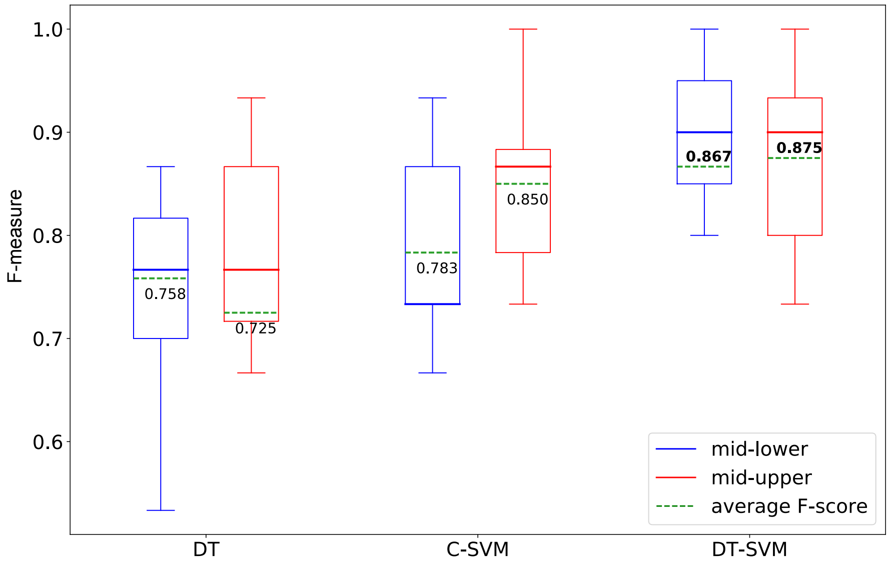


Figure 5.8: Comparison of the F-measure of different classifiers from the eight-fold cross-validation trails.

presents the number of note instances used for cross-validation and testing with respect to the mid-lower and mid-upper region, respectively. For each split in a region, note instances with the same label (N, NLOH, NLH, LOH or LH) contributed the same number of instances. Moreover, key velocities of the instances in each split obtain a ratio of 1:1:1 for *piano*:*mezzo-forte*:*forte*. For instance, the 30 note instances for the mid-lower test set consist of 6 instances with label N, every two of which were played at the same velocity, and likewise 6 × 4 instances with the other four labels (NLOH, NLH, LOH and LH). Because note instances with label N, NLH, LOH and LH are less than the ones with label NLOH, some of them were repeatedly used in the training sets between folds for cross-validation.

5.4.2 Results and Discussions

The performance of each classifier from the cross-validation trails is presented as a box plot in Figure 5.8. An average score of F-measure over the 8 trails can represent the overall performance of a classifier used for multi-class classification tasks (see Section 2.5.1 for the details of F-measure). For pedalling technique detection on notes in both mid-

Table 5-C: Normalised confusion matrices of pedalling technique detection of piano notes in mid-lower region.

		Predicted Label				
		N	NLOH	NLH	LOH	LH
Actual Label	N	**1.00**	0	0	0	0
	NLOH	0	**1.00**	0	0	0
	NLH	0	0	**0.83**	0.17	0
	LOH	0	0	0	**1.00**	0
	LH	0	0	0	0.33	**0.67**

Table 5-D: Normalised confusion matrices of pedalling technique detection of piano notes in mid-upper region.

		Predicted Label				
		N	NLOH	NLH	LOH	LH
Actual Label	N	**0.67**	0	0	0	0.33
	NLOH	0	**1.00**	0	0	0
	NLH	0	0.17	**0.83**	0	0
	LOH	0	0	0	**1.00**	0
	LH	0	0	0	**0.67**	0.33

lower and mid-upper regions, our DT-SVM achieves the highest average F-measure score reported as 0.867 and 0.875 respectively. This yields a performance improvement, outperforming the DT and the C-SVM by 0.109 and 0.084 respectively at the mid-lower region, as well as 0.15 and 0.025 at the mid-upper region.

To take a detailed look at the performance of the trained DT-SVM in the testing phase, Table 5-C and Table 5-D display the confusion matrices corresponding to the test using notes in the mid-lower and the mid-upper region respectively. The confusion matrices are normalised by the number of elements in each class. Labels representing the pedalling techniques in these two tables are identical to the ones listed in Table 5-A. The F-measure from the test result is 0.933 for the mid-lower region and 0.867 for the mid-upper region. Based on the confusion matrices, NLH (*non-legato half*) and LH (*legato half*) pedalling are more easily confused with other pedalling techniques. One possible explanation is that the effects of half pedalling were not thoroughly characterised using the limited note instances in the training set. More frequent interactions between strings and dampers are introduced by half pedalling technique. The interaction is also dependent on key velocity. A more comprehensive modelling on piano tones with half pedalling should take key velocity into consideration. Moreover, we observe ambiguities in the spectrogram between notes with legato pedalling techniques that are different in pedal depth, i.e., LOH (*legato over-half*) versus LH (*legato half*). This is vaguer on the spectrograms of notes in the mid-upper region. The normalised confusion matrix in Table 5-D also shows that notes with LH were falsely classified into LOH. All of these issues indicate that more specialised features should be developed.

To sum up, experiment results present the effectiveness of the designed features in differentiating categories of pedalling techniques. With these features, the trained classifiers (DT, C-SVM and DT-SVM) all achieved averaged F-measure higher than 0.7 in the cross-validation. The proposed DT-SVM obtained the best performance. For DT-SVM, the more the data are misclassified at the upper node of the decision tree, the worse the classification performance becomes. Hence it is important to determine the structure of the DT-SVM to minimise the classification error. We separated notes played normally versus notes played with the pedal at the first node of the decision tree. Such structure for detecting pedalling techniques from notes in the mid-upper region is prone to the misclassification between N and LH at the first node, leading to more errors in detecting the LH pedalling at other nodes. This structure is more effective on the detection from notes in the mid-lower region. There are possibilities to gain better test results using a different structure of DT-SVM.

5.5 Summary

This chapter investigated the effects of pedals on piano tones, which informed our primary study on piano pedalling technique detection from the audio domain. We first observed the evolution of partials and corresponding residuals respectively from isolated notes played with pedalling techniques, which are different in pedal depth and in onset time with respect to note attack time. Features were designed and extracted from the partials and the residuals based on their evolution. A model using decision-tree-based support vector machines was trained to classify the notes into *normal*, *non-legato over-half*, *non-legato half*, *legato over-half* and *legato half* pedalling techniques. Effectiveness of the model was demonstrated using cross-validation. A mean F-measure score of 0.867 and 0.875 was obtained for classification of notes in mid-lower and mid-upper region separately. The results indicate that the proposed features are able to characterise the effects of pedalling techniques on isolated notes in the middle register of the piano.

Although good performance measurements were obtained from the pedalling technique classification task, there are strong assumptions about the ability to extract clean features from isolated notes, which may not apply in more generic cases, such as continuous piano playing. For this reason, more sophisticated approaches should be developed in order to solve pedalling detection in polyphonic piano music. Such approaches are required to deal with feature extraction in the presence of overlapping partials when different notes are sounding. To this end, we start with the detection of the legato-pedal onset in the next chapter, using features from the residual component that correlate with the acoustic characteristics when legato pedalling is played.

Chapter 6

A Sympathetic Resonance Measure for Legato-Pedal Onset Detection

6.1 Introduction

As mentioned in Section 1.3, the sustain pedal is frequently used in expressive piano performances to colour the timbre. Besides sustaining the sounding notes, the sustain pedal also allows strings associated with other keys to vibrate due to coupling via the bridge. This phenomenon is known as *sympathetic resonance* and is defined in the Dictionary of Acoustics as "*resonant or near-resonant response of a mechanical or acoustical system excited by energy from an adjoining system in steady-state vibration*" [19]. Pianists usually embrace the phenomenon to produce seamless legato through a technique called *legato pedalling* [166]. A key element of employing this technique is when to press the pedal, i.e., legato-pedal onset time, which helps to sustain the intended notes and avoid enriching unwanted sonority from the prior notes. In this chapter, we focus on the detection of legato-pedal onset time. As introduced in Chapter 5, features from the residuals can be used to distinguish piano tones played with legato pedalling. This informs us to develop more dedicated features from the residuals for legato-pedal onset detection in the context of polyphonic piano music.

This chapter is organised as follows. In Section 6.2, we design a sympathetic resonance measure based on the residuals obtained by removing the partials from the original sound. The partials are derived using a piano transcription technique. Given that modelling the specific instrument being transcribed can efficiently improve transcription performance [147], a specific piano was used to build the dataset. It is also a reasonable assumption that model parameters of a specific piano are accessible for many performance scenarios. Next, we extract features from the sympathetic resonance measure and consider the legato-pedal onset detection as a binary classification problem, i.e., presence/absence of the onset. Our proposed method is evaluated and discussed in Section 6.3. Finally, we summarise this chapter in Section 6.4. This chapter incorporates material from “Piano Legato-Pedal Onset Detection Based on a Sympathetic Resonance Measure” by Liang, Fazekas and Sandler originally published in *Proceedings of the 26th European Signal Processing Conference (EUSIPCO)* [167]. All code used in this chapter is made publicly available[1].

6.2 Methods

6.2.1 Intuitions and Framework

To illustrate the effect of legato pedalling, especially the resulting sympathetic resonance, idealised spectrograms of two successive chords played with or without the sustain pedal are presented in Figure 6.1. For the pedalled case, the sustain pedal is pressed to prolong the first *Cmaj* chord, while the fingers are still holding down the keys. When the fingers are lifted to reach for the *Fmaj* chord, the *Cmaj* chord is sustained because

[1] `https://github.com/beiciliang/eusipco2018-legatopedal`

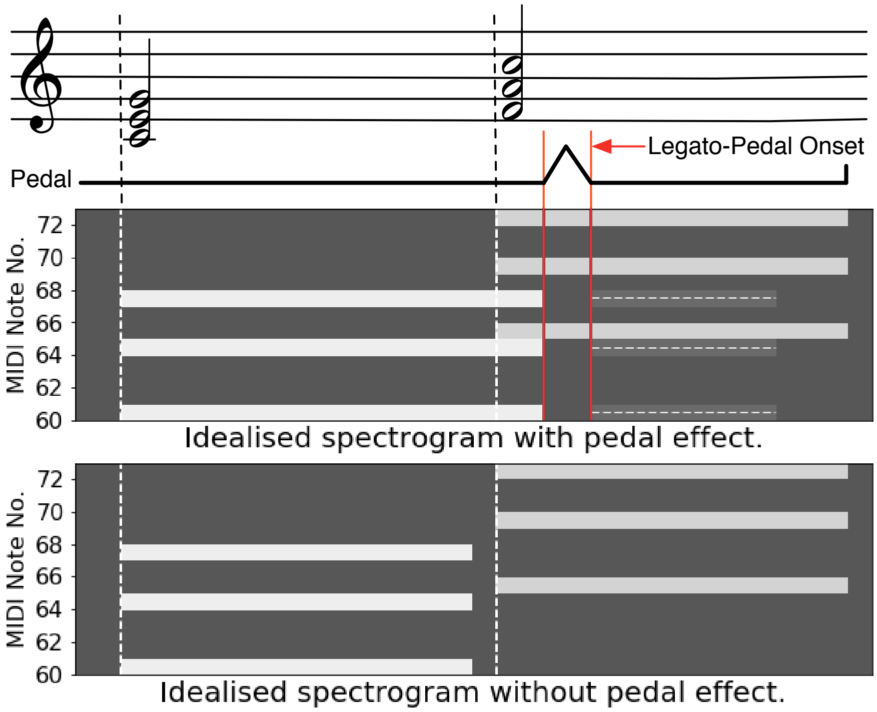

Legato-Pedal Onset
Pedal
MIDI Note No.
72
70
68
66
64
62
60
Idealised spectrogram with pedal effect.
MIDI Note No.
72
70
68
66
64
62
60
Idealised spectrogram without pedal effect.

Figure 6.1: Idealised spectrograms of two successive chords respectively played with and without the sustain pedal.

the pedal prevents the dampers from falling onto the strings. Immediately after the *Fmaj* chord onset, the pedal is released to avoid blurring effect caused by the overlap of the two sonorities. The dampers stop the vibrations in all strings whose keys are not currently pressed. Then, the pedal is pressed again to sustain the *Fmaj* chord, lifting dampers off the strings. This can slightly co-excite the damped strings associated with the previous *Cmaj* chord with the playing *Fmaj* chord (shown as the horizontal dashed line in Figure 6.1). A detailed study of such an indirect excitation on piano tones is introduced in [15]. We proposed to measure the sympathetic resonance using the weak co-excitation of damped notes, which is due to the legato pedalling technique. Features characterising the sympathetic resonance are expected to facilitate the detection of legato-pedal onset.

To this end, a framework describing the detection process is illustrated in Figure 6.2. We first obtain the note events through a state-of-the-art specific piano transcription method proposed in [96]. Isolated notes of the same piano are used to form the templates

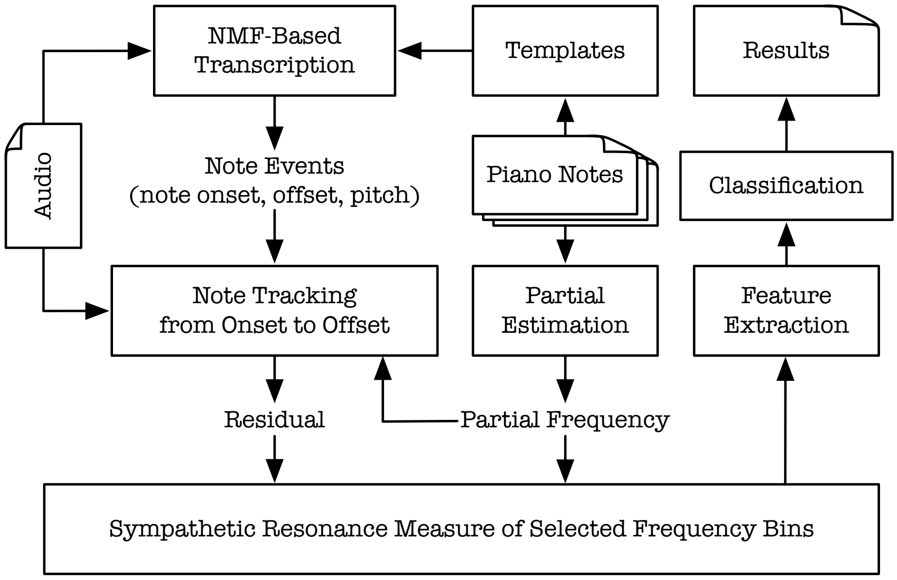


Figure 6.2: Framework of the legato-pedal onset detection method.

for the transcription. Their partial frequencies are estimated using the method proposed in [150], which was also used and introduced in the previous Section 5.3.1.1. Partial components are determined by the partial frequencies of the notes from their onset to offset times based on the transcription results. We then obtain the residuals by subtracting the partial components from the original sound. Features are extracted from residuals using a sympathetic resonance measure, which is the main contribution of this chapter. Finally, the existence of legato-pedal onset is determined via a classification mechanism. Each step is explained in the following sections.

6.2.2 Transcription for Specific Piano

Note-level piano transcription converts audio into a set of note events, each consisting of pitch, onset and offset times. Note dynamics and other expressive techniques are rarely transcribed. A dominant algorithm in automatic music transcription for the last two decades is non-negative matrix factorisation (NMF) [74]. NMF can factorise a spectrogram of a piano recording into 88 spectral bases and corresponding activations. Each spectral base is associated with a piano note. The corresponding activation represents when and how intensely that note is played over time. With NMF, the magnitude spectrogram of the sound to be transcribed can be represented using:

$$V_{mn} = \sum_{\rho} W_{m\rho} H_{\rho n}, \tag{6.1}$$

where $\boldsymbol{V}$ is the reconstructed spectrogram, $\boldsymbol{W}$ is the note template, and $\boldsymbol{H}$ is the note activation. $m \in [1, M]$ is the frequency bin, $n \in [1, N]$ is the time frame, and $\rho \in [1, 88]$ is the pitch index.

For a specific piano, transcription performance can be improved by modelling piano acoustical features. In the case of supervised NMF, this is done by pre-computing and fixing the template using recordings that each contains only a single note from the same piano. Considering the different spectral and temporal characteristics at the attack and decay phases of a piano note, these two phases can be reconstructed individually to form the template. This was proposed in [96], which performed as a state-of-the-art method for specific piano transcription by the time we conducted this study. We employed the methods in [96] and reformulated Equation 6.1 into:

$$V_{mn} = \sum_{\rho} W^{a}_{m\rho} H^{a}_{\rho n} + \sum_{\rho} W^{d}_{m\rho} H^{d}_{\rho n}, \tag{6.2}$$

where $\boldsymbol{W}^a$ and $\boldsymbol{H}^a$ are the template and the activation for the attack phase, respectively, and $\boldsymbol{W}^d$ and $\boldsymbol{H}^d$ are the ones for the decay phase.

According to the piano acoustics, the attack activations can be formulated as a convolution of spike-shaped note activations $\boldsymbol{H}^s$ and a transient pattern $\boldsymbol{P}^t$ decided by the amplitude attack envelope. Accordingly, $\boldsymbol{H}^a$ are formulated as:

$$H^{a}_{\rho n} = \sum_{\tau = n - N_a}^{n + N_a} H^{s}_{\rho \tau} P^{t}(n - \tau), \tag{6.3}$$

where N_a is set to the ratio of the window size and frame hop size for computing the spectrogram[2]. For the decay phase, since piano notes roughly follow an exponential decay, the decay activations can be generated by:

$$H^d_{\rho n} = \sum_{\tau=1}^{n} H^s_{\rho\tau} e^{-(n-\tau)\lambda_\rho}, \tag{6.4}$$

where λ_ρ is the decay rate for pitch ρ. Therefore the complete NMF model is formulated as follows:

$$V_{mn} = \sum_{\rho} W^a_{m\rho} \sum_{\tau=n-N_a}^{n+N_a} H^s_{\rho\tau} P^t(n-\tau) + \sum_{\rho} W^d_{m\rho} \sum_{\tau=1}^{n} H^s_{\rho\tau} e^{-(n-\tau)\lambda_\rho}, \tag{6.5}$$

Given that isolated-note recordings have the same note onset time at 0.5 seconds, activations $\boldsymbol{H}^s$ are fixed in order to update $\boldsymbol{W}^a$, $\boldsymbol{W}^d$, $\boldsymbol{P}^t$ and α in the training stage. In the transcription stage, $\boldsymbol{H}^s$ are updated with the trained templates, transient patterns and decay rates. The updated $\boldsymbol{H}^s$ are then used to detect the onset and offset times for each pitch. All these parameters are estimated by minimising the Kullback-Leibler divergence between the original spectrogram $\boldsymbol{S}$ and the reconstructed spectrogram $\boldsymbol{V}$. A detailed derivation can be found in [168], with our Python implementation available online[3].

Given the estimated $\boldsymbol{H}^s$, the actual transient patterns of notes can be obtained by attack activations, i.e., $\boldsymbol{H}^a$ using Equation 6.3. Note onset can be detected from $\boldsymbol{H}^a$ by peak picking. Only peaks that exceed smoothed attack activations by a threshold are considered as onset candidates. Here, the smoothed attack activations are computed using a moving average filter with a window of 20 bins. The threshold is adapted to each piece using the Equation 6.6:

$$Thre = \delta \max(H^a_{\rho n}), \tag{6.6}$$

[2] We applied the same setup for N_a as in [96] such that the range of the transient pattern is determined by the overlap in the spectrogram.

[3] `https://github.com/beiciliang/modelAttackDecay-for-piano-transcription`

where δ is a hyperparameter which has an impact on the piano transcription results. As seen in the transcription experiments in [168], comparable performance was achieved when the value of δ was around -30dB. Hence δ was set to -30dB for our study. After thresholding, double peaks which are close to each other may be returned. They are merged into one peak if their intervals are smaller than the minimum interval, which are set to 0.1 seconds. A weighted average of the indices of the two peaks determines the index of the merged peak. Here the weight is decided by the amplitude of the two peaks. The amplitude of the merged peak is the sum of that of the two peaks. This process was iteratively applied in [168] to remove multiple peaks.

For the offset detection, a method proposed in [169] was used. The state sequence for each pitch consists of 0 or 1, denoting the state as *off* or *on* respectively. The optimal state sequence for each pitch can be derived by applying dynamic programming on the normalised costs of the two states, which sum to 1 in every frames. The normalised costs are defined as follows:

$$J_\rho(state, n) = \begin{cases} \sum_{m=1}^{M} D_{KL}(X_{mn}, V_{mn} - V_{mn}^{\rho}), & state = 0 \\ \sum_{m=1}^{M} D_{KL}(X_{mn}, V_{mn}), & state = 1 \end{cases} \tag{6.7}$$

$$\tilde{J}_\rho(state, n) = J_\rho(state, n) / \sum_{\tilde{state}} J_\rho(\tilde{state}, n), \tag{6.8}$$

where $\boldsymbol{V} - \boldsymbol{V}^\rho$ is the reconstruction excluding pitch ρ, and D_{KL} is the Kullback-Leibler divergence. More details on the offset detection can be found in [169] and [96].

Results of the specific piano transcription help to obtain the partial and residual components using the partials estimated from isolated-note recordings of the same piano. The transcription results can also inform which notes are damped, based on which the sympathetic resonance is measured from the residuals.

6.2.3 Sympathetic Resonance Measure Based on Residuals

When a legato pedalling is played, the effect of sympathetic resonance is enhanced by string coupling via the bridge. This transfers energy from string vibrations of a played tone to unstruck strings of the other tones. The core idea of sympathetic resonance measure is to track the excitation of unstruck strings in order to detect the legato-pedal onset. Such excitation can be reflected on the magnitude changes on the damped-note partials as indicated in the spectrogram in Figure 6.1.

To exclude the effects of the sounding notes, a method of partials plus residuals decomposition as mentioned in Section 5.3.1 was applied. Given that isolated recordings of the piano we wish to transcribe are available, partial frequencies of each note can be estimated using the methods introduced in Section 5.3.1.1. Values of the partial frequencies are fairly fixed between the note onset and offset times. For a piece with transcription results, the partial components can be obtained by tracking the amplitude and phase of the transcribed notes' partial frequencies from their detected onsets until offsets. Residuals were thus obtained by subtracting the partial components from the original sound.

At this stage, the residuals mainly consist of background noise, the percussive sound of hammer-string strikes from note attacks, and the effect of sympathetic resonance. To minimise the influence of percussive components, harmonic percussive source separation (HPSS) using a median-filtering technique proposed in [170] was applied to filter out the percussive sound from the residuals. In order to detect the energy changes induced by legato pedalling and exclude the influence of residual components other than sympathetic resonance, only the energy of unstruck strings was measured. Notes associated with unstruck strings were determined by the preceding notes that are beyond the time range between their detected onset and offset times. For instance, damped notes after the second-chord onset in Figure 6.1 are the notes of the preceding chord, i.e., *C4*, *E4* and *G4*. Partials of the damped notes were informed by the partial frequencies estimated

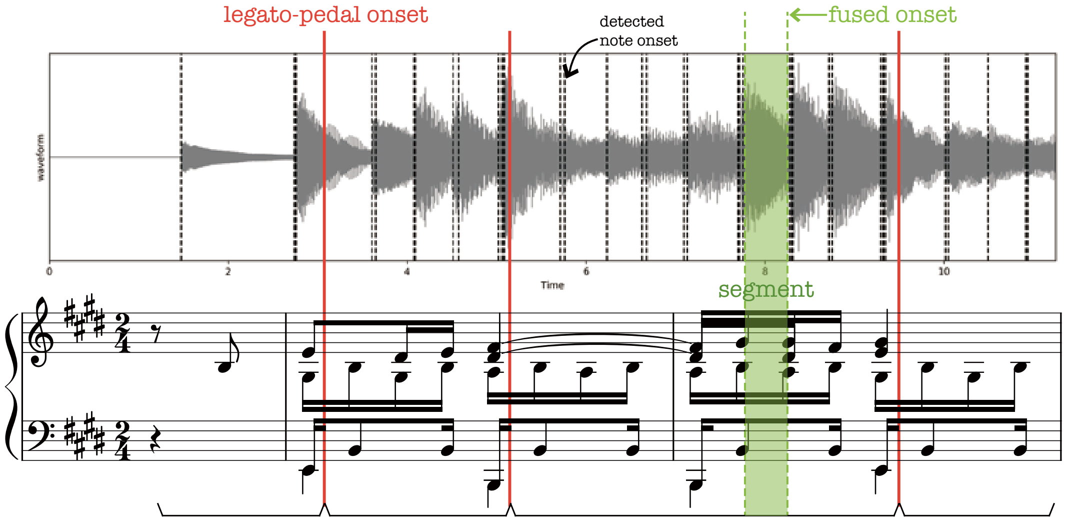


Figure 6.3: Illustration of the difference between detected note onset and fused onset that is used to determine the segments, from which the existence of legato-pedal onset is decided.

from isolated notes (see Section 5.3.1.1 for the details of partials estimation). According to Parseval's theorem, the energy of these selected partials in the frequency domain can be used to represent the energy of unstruck strings in the time domain. This energy is related to the extent of sympathetic resonance and therefore used to indicate the existence of legato-pedal onset. It is measured by the root-mean-square (RMS) energy using Equation 6.9:

$$SRM(r[n]) = RMSE(R[n, SP_n]) = \sqrt{\sum_{m_{sp}} |\frac{|R[n]|[m_{sp}]}{M_{sp}[n]}|^2}, \tag{6.9}$$

where $SRM(r[n])$ denotes the sympathetic resonance measure of $r[n]$, which is the residual components at the n-th time frame, and $R[n]$ is the associated residual spectrum. $RMSE(R[n, SP_n])$ denotes the RMS energy of the selected partials (SP) of the damped notes in $R[n]$. The frequency bins corresponding to the selected partials are designated as m_{sp}, where sp is the element in the list SP_n. The amplitude in each selected frequency bin can be then obtained from $R[n]$ and denoted as $|R[n]|[m_{sp}]$. $M_{sp}[n]$ is the total number of selected partials in the current frame.

Due to the nature of legato pedalling, its onset happens between two note onsets. We can define segments as the frames between two successive note onsets and then detect legato-pedal onset from each segment. To determine the segments, we fused the detected note onsets that are within a fixed temporal tolerance window of an estimated 16th note duration. This is because multiple onsets may be detected for the notes played as a single chord, and it is not possible to change the pedal with every note [22]. Frames between every two successive onsets after fusing were defined as a segment. Figure 6.3 illustrates how a segment is determined using the fused note onsets based on the transcribed note events from the first three bars of Chopin's Op. 10 No.3. The existence of legato-pedal onset can be decided from every segment using the proposed sympathetic resonance measure.

The whole procedure of measuring the sympathetic resonance from every segment is illustrated by Algorithm 1, where $P[i]$, $ON[i]$ and $OFF[i]$ refer to the pitch index, onset and offset time frames of the i-th transcribed note event ($i \in [1, I]$). $PF[P[i]]$ is the frequency bins corresponding to the estimated partials of the $P[i]$. Finally the sympathetic resonance measure for a piece is saved as a vector SRM.

Algorithm 1 Sympathetic resonance measure

```
Require: $\tau$: estimated duration of a 16th note in frames; $m$: frequency bin and $m \in [1, M]$; $n$:
  time frame and $n \in [1, N]$
  procedure MEASURE(P, ON, OFF, PF, R)
      $SRM \leftarrow zeros(N)$
      for all $j \in [1 : I - 1]$ do
          if $ON[j+1] - ON[j] > \tau$ then
              $n \leftarrow ON[j]$
              $SP_n \leftarrow empty\ list$
              for all $i \in [1 : I]$ do
                  if $ON[i] < n < OFF[i]$ then
                      $SP_n \leftarrow append(SP_n, PF[P[i]])$
              $SRM[ON[j] : ON[j+1]] \leftarrow RMSE(R[ON[j], SP_n] : R[ON[j+1], SP_n])$
      return $SRM$
```

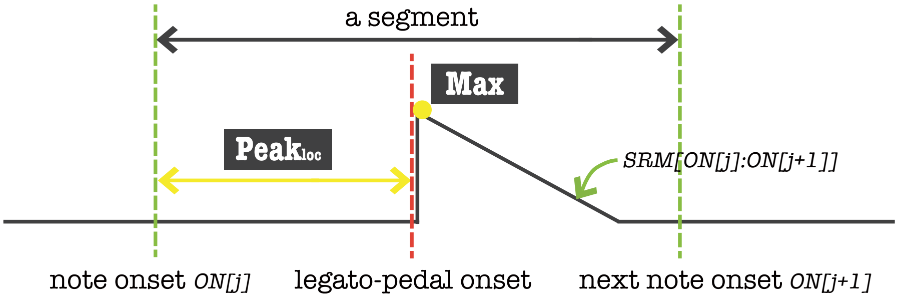


Figure 6.4: Idealised changes of sympathetic resonance measure and the extracted features in a segment with the existence of legato-pedal onset.

6.2.4 Feature Extraction and Binary Classification

In the ideal case as illustrated in Figure 6.4, the value of SRM in a segment is increased at the moment of legato-pedal onset, and then gradually decreased. SRM should stay stable in the segments without legato-pedal onset. We can extract features to characterise such SRM changes in order to determine the existence of legato-pedal onset in a segment. This can turn the legato-pedal onset detection into a binary classification problem, where the decision can be made using a machine learning method. Results from the segment-level detection can be interpreted as which group of notes is played with the legato pedalling technique. We didn't set the detection at every frame because exact times of legato-pedal onsets are not necessary for the pianist to perform a piano piece. It is also observed in most music scores that notation for an intended legato-pedal onset is always aligned with a group of note onsets.

As observed in Figure 6.4, a peak with the maximum value of a segment appears when the SRM is enhanced by the legato-pedal onset. The maximum value was extracted as a feature per segment and recorded on both linear and decibel scale (denoted as Max_{linear} and Max_{dB} respectively). Using the decibel scale can approximate how humans perceive the extent of sympathetic resonance. We assess which scale is more effective in distinguish segment with/without legato-pedal onset in the experiments presented in Section 6.3.3 such that feature representing the maximum SRM can be selected. Moreover, because legato pedalling is used after note onsets, the peak location with respect to the number of frames away from the note onset, i.e., starting point of the segment, was also extracted and denoted as $Peak_{loc}$. Finally, $Peak_{loc}$ and maximum SRM using the selected scale were concatenated together as a 2D feature characterising SRM for each segment.

Since binary classifier selection is not the main focus in this study, logistic regression, as one of the most common methods for binary classification problems, was chosen to model the probability of the existence of legato-pedal onset per segment using the input features. In the training stage, a threshold value from the probability can be obtained to discriminate the presence or absence of legato-pedal onset. Then in the testing stage, the trained logistic regression model can be used as a binary classifier.

6.3 Experiments

In this section the proposed method is evaluated in the legato-pedal onset detection task. We first describe the dataset and detail the experimental setup. A logistic regression model with the selected features are trained into a binary classifier. Presence or absence of legato-pedal onset can be determined. Finally we present and discuss the testing results.

6.3.1 Dataset

Existing public annotated piano datasets were developed for research on multi-pitch estimation. For piano pedalling technique detection, we built our dedicated dataset using Disklavier rendering as introduced in Section 4.3. Recordings of 88 isolated notes played with *mezzo-forte* dynamics on the same Disklavier are available. They were used to train the NMF templates and to estimate the partial frequencies. Recordings of four well-known pieces from different music eras were obtained using the same piano. They were labelled with pedal onset times in seconds according to the input MIDI data.

Using the proposed method, legato-pedal onset times are detected at a segment level.

Table 6-A: The number of labels and total segments in each piece.

Piece	**#Label 0**	**#Label 1**	**#Segments**
Beethoven Op.31 No.2-3	1113	84	1197
Chopin Op.10 No.3	438	108	546
Brahms Op.10 No.1	161	110	271
Ravel Jeux d'eau	710	88	798
Total Number	2422	390	2812

Accordingly, the method was evaluated using another ground truth, which annotates the existence of legato-pedal onset per segment. To prepare this ground truth, each audio signal was segmented as discussed in Section 6.2.3 based on the transcribed note onsets. Each segment was then labelled "1" to denote the presence of legato-pedal onset, otherwise segments were labelled "0". Table 6-A lists the number of segments annotated using "0" or "1" in each piece. The current dataset is limited in the number of music pieces, however there are almost 3000 segments in total to be classified.

6.3.2 Experimental Setup

The input signals were divided into frames using 2048-sample Hanning window (hop size = 512) to compute the spectrogram. Given the transcription results and the estimated partial frequencies, partials can be tracked and then removed in order to obtain the residuals. The sympathetic resonance was measured from the residuals and then segmented. The proposed features were extracted from the calculated sympathetic resonance measure in every segment.

To evaluate the model, data in each piece was separated into two halves, one for training and the other one for testing. This piece-level evaluation was selected because the overall tempo and dynamics in a piece affect the attributes of the trained model. Moreover, models were trained with weighted classes in the Beethoven and Ravel pieces, which exhibit very unbalanced data as seen in Table 6-A. Their ratio of segments with-/without legato-pedal onset is around 7.55% and 12.39% respectively. Given the test results for every piece, we then calculated precision (P_1), recall (R_1) and F-measure (F_1)

Table 6-B: AIC results of logistic models with different features.

Features	**AIC**
Max_{linear}	2555.36
Max_{dB}	2507.39
$Max_{dB} + Peak_{loc}$	**2361.35**

with respect to label "1". In addition, we show the overall performance using micro-averaged F-measure (F_{micro}) due to the imbalance of two labels. Details on calculating P_1, R_1, F_1 and F_{micro} were introduced in Section 2.5.1. We used *Scikit-learn* [139] to construct the model and compute the performance metrics.

6.3.3 Feature Selection

As we introduced in Section 6.2.4, Max_{linear} and Max_{dB} in every segment were extracted as features. To determine which one better represents the maximum value of segments, data from the training set were used. We evaluated logistic regression models with the two features separately, using the Akaike information criterion (AIC) [171]. AIC estimates the relative amount of information lost by a given model. A logistic regression model with a more effective feature should yield a smaller value of AIC.

Table 6-B presents the AIC values corresponding to the logistic regression model with different features. To determine the feature representing the maximum value of SRM in a segment, Max_{dB} was selected because it returns a smaller AIC value of 2507.39 than 2555.36 by Max_{linear}. We also evaluated the logistic regression model with 2-dimension features consisting of Max_{dB} and $Peak_{loc}$. This was chosen as the final feature to train the logistic model because it gives the smallest AIC value of 2361.35.

6.3.4 Results and Discussions

Table 6-C presents the model performances for each piece. The overall results indicate that our method extracts relevant features to represent the effect of sympathetic reso-

Table 6-C: Test results of legato-pedal onset detection in each piece.

Piece	$\boldsymbol{P_1}$	$\boldsymbol{R_1}$	$\boldsymbol{F_1}$	$\boldsymbol{F_{micro}}$
Beethoven Op.31 No.2-3	0.13	0.38	0.20	0.80
Chopin Op.10 No.3	0.69	0.69	**0.69**	**0.88**
Brahms Op.10 No.1	0.56	0.58	0.57	0.62
Ravel Jeux d'eau	0.23	0.87	0.36	0.71

Table 6-D: Percentage of segments with label 1 and mode of segment duration in seconds in each piece.

Piece	**Label 1 %**	**Mode of Segment Duration (Seconds)**
Beethoven Op.31 No.2-3	7.02	0.1
Chopin Op.10 No.3	19.78	0.5
Brahms Op.10 No.1	40.59	0.2
Ravel Jeux d'eau	11.03	0.1

nance and helps to detect legato-pedal onsets from audio. In terms of the performance metric for label "1" (indicating legato-pedal onset exists in a segment), a higher value of R_1 than P_1 is obtained in general. However, given that most segments are labelled as "0", a bias towards "0" can be introduced in the training process. The trained model can result in more false positives and therefore decrease the P_1. This also reflects on the classification performance of music pieces with more imbalance in their data. According to the ground truth data, percentage of the segments with the label "1" with respect to each piece is presented in Table 6-D. With a higher percentage, the trained model can detect legato-pedal onset with higher P_1. Accordingly, detection on the Chopin and Brahms piece achieve better P_1 and F_1 than those of the Beethoven and Ravel piece.

Moreover, we assume the segment duration, i.e., the time interval between two successive note onsets, can have an effect on the proposed detection method. We calculated the mode value of segment duration in the four pieces respectively based on their ground truth data. As presented in Table 6-D, pieces with more segments of longer duration tend to obtain better performance. Given that our features were extracted per segment, in the segment with a short duration, features can be masked by the note transients. They are hence less representative as an indicator of legato-pedal onset which leads to poor classification performance in the Beethoven and Ravel pieces.

Differences in pedal use between composers and musical eras also contribute to explaining our evaluation results. The highly unbalanced training data in the Beethoven piece is due to the fact that legato pedalling was rarely used in Beethoven's time. This results in poor P_1, R_1 and F_1 in the Beethoven piece. On the contrary, it is well acknowledged that legato pedalling is an essential ingredient to create contrast between pedalled and unpedalled notes in Chopin's works. This helps to yield cleaner features and consequently the best performance in the Chopin piece. If the piece itself has cross-rhythms and dense harmonic structure, which Brahms' music is firmly rooted in, the extent of sympathetic resonance may not be significantly changed by legato-pedal onset. In this case, our features are less discriminative. Similarly, other playing techniques that are correlated with the effect of sympathetic resonance may degrade classification performance. This is observed in the Ravel piece, which puts emphasis on timbral nuances, expanding the keyboard and pedalling techniques more than the use of legato pedalling.

6.4 Summary

This chapter presented a method for detecting legato-pedal onsets of a known piano based on a measure of sympathetic resonance. The intuition behind this method is that the energy of unstruck strings can represent the extent of sympathetic resonance which changes with the legato pedalling technique. It is noted that our method is the first to detect pedalling technique in polyphonic piano music.

In the proposed method, residuals were obtained using piano transcription and partial estimation. Sympathetic resonance was measured from the residuals and segmented. In each segment, the maximum value in the decibel scale and peak location were extracted as features. The existence of legato-pedal onset per segment was determined using a logistic regression classifier trained on the features. The overall performance shows that the trained model can be used as an indicator of legato-pedal onset, especially in the music pieces with more instances of legato pedalling techniques and longer time intervals between note onsets.

From a practical perspective, the specific piano transcription technique, which was used as an intermediate step in our method, would affect the residual acquisition and segmentation due to the errors in the transcribed note events. In addition, because of the different nature of music pieces, data from part of a piece were required for training the machine learning models in order to facilitate the detection in the rest part of the piece. How to generalise the detection for any music piece without this training strategy remains a question. To address these issues, deep learning algorithms are proposed in the following Chapter 7 to design more accurate and general methods for detecting piano pedalling techniques, which are not limited to legato pedalling.

Chapter 7

Deep Learning Methods for Sustain-Pedal Detection

7.1 Introduction

Pedalling techniques change very specific acoustic features, which can be observed from their spectral and temporal characteristics on isolated notes as seen in Chapter 5. However, their effects are typically obscured by the variations in pitch, dynamics and other elements in polyphonic music. Automatic detection of pedalling techniques using hand-crafted features becomes a challenging problem as discussed in Chapter 6. Given enough labelled data, deep learning models have shown the ability of learning hierarchical features. If these features are able to represent characteristics corresponding to pedalling techniques, the model can serve as a detector.

In this chapter, we focus on detecting the technique of the sustain pedal from audio recordings using deep learning methods. According to piano acoustics and the observations in the previous chapters, musical features can be different at the start (pedal onset) versus during the pedalled segment. We propose to train the deep learning models for pedal onset and pedalled segment separately to better localise the audio frames played with the sustain pedal. The models are designed through exploiting the knowledge of piano acoustics and physics in Section 7.2. Experiments on distinguishing excerpts with or without the sustain-pedal effect using the trained models are presented in Section 7.3. Here the excerpts are arranged in pairs (*pedal* versus *no-pedal* versions). They were clipped from the paired audio of music pieces in the dataset introduced in Section 4.4.

For the frame-wise detection on music pieces, two strategies with the help of the trained models are presented in Section 7.4. One is to apply decision fusion [172] to the outputs of the two trained models from sliding windows over the music piece in order to decide the portions played with the sustain pedal. The other one is based on transfer learning techniques [113]. This allows the trained model to be adapted to the target task, where the recording instrument and room acoustics are different. Given that our deep learning models are trained using synthesised data, transfer learning is expected to obtain a better feature representation for the data consisting of acoustic piano recordings. Hence better performance on frame-wise detection can be achieved by transfer learning. This is examined through cross-validation using 10 passages of Chopin's music, which were recorded in a real scenario as introduced in Section 4.2.

This chapter incorporates materials from "Piano Sustain-Pedal Detection Using Convolutional Neural Networks" and "Transfer Learning for Piano Sustain-Pedal Detection" by Liang, Fazekas and Sandler originally published in *Proceedings of the 44th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* [173] and *Proceedings of the International Joint Conference on Neural Networks (IJCNN)* [174], respectively. All code used in this chapter is made publicly available[1].

7.2 Training Convolutional Neural Networks

Convolutional Neural Networks (CNNs, see Section 2.3.4.2 for technical background) have been used to boost the performance in MIR tasks, with the ability to efficiently

[1] `https://github.com/beiciliang/sustain-pedal-detection`

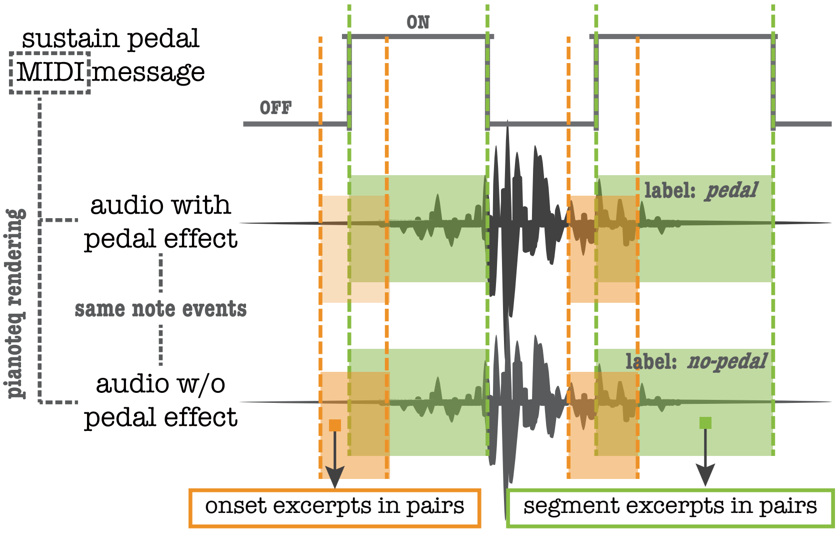


Figure 7.1: Process of generating excerpts in pairs based on Pianoteq rendering.

model temporal features [175] and timbre representations [176]. We chose CNNs to learn time-frequency contexts related to the sustain pedal, using synthesised excerpts in pairs, which correspond to the *pedal* and *no-pedal* versions of audio data with the same note events. Using this method, contexts that are invariant to large pitch and dynamics changes can be learned by the CNN models (collectively denoted by `convnet` hereafter). Preparation of excerpts in pairs is detailed in Section 7.2.1. The input and output representations for training the `convnet` are presented in Section 7.2.2. How to incorporate knowledge of piano acoustics and physics for the `convnet` design is discussed in Section 7.2.3. The models were separately trained for two binary classification tasks, which have the same goal of differentiating the *pedal* case from the *no-pedal* one. One task is focused on pedal onset detection, from which the trained model is denoted as `convnet-onset`. The other task is to determine the pedalled segment. The corresponding trained model is designated as `convnet-segment`.

7.2.1 Preparation of Music Excerpts in Pairs

The preparation process of excerpts in pairs for training and validating `convnet-onset` and `convnet-segment` is illustrated in Figure 7.1. MIDI files and the associated two versions of Pianoteq-rendered pieces were provided by the dataset introduced in Section 4.4. Ground-truth annotations for each piece consist of binary labels (*on* and *off*) indicating the sustain pedal is pressed or released at every frame. They were obtained by thresholding the sustain-pedal MIDI message in the range [0,127] at 64. A pedal onset is determined to have happened during a frame where the pedal state changes from *off* to *on*. A pedalled segment is determined to start at a pedal onset and finish when the state returns to *off*.

According to the distribution of pedalled-segment duration calculated from all the MIDI files, the sustain pedal is commonly pressed for between 0.3 and 2.3 seconds. To prepare fixed-length excerpts for training `convnet-onset`, we clipped 0.5-second excerpts around every pedal onset. Each excerpt starts from 0.2 seconds before a pedal onset time and ends at 0.3 seconds after the pedal onset time. Excerpts for training `convnet-segment` were clipped from pedalled segments which are more than 0.3-second long, and then processed to obtain 2 seconds in length through repeating/trimming the pedalled segments shorter/longer than 2 seconds. The start and end times of these *pedal* excerpts were also used to obtain *no-pedal* excerpts from audio without sustain-pedal effect. Therefore excerpts were obtained in pairs.

As introduced in Section 4.4, the train and validation sets were formed by paired excerpts clipped from pieces of the competition year 2002, 2004, 2006, 2008 and 2009. To compare `convnet-onset` and `convnet-segment` of different architectures in a more efficient way, we created a smaller train/validation set to reduce the training time. This was done by randomly taking a thousand samples from the excerpts of each composer. Since there are less than a thousand excerpts for some of the composers, 67540 excerpts were formed for `convnet-onset`, and 62424 for `convnet-segment`. The excerpts were sampled in pairs such that the ratio of *pedal* and *no-pedal* excerpts was kept as 1:1. They were then split into 80%/20% as the train/validation set.

Table 7-A contains aggregate statistics of the original dataset presented in Section 4.4 and the paired excerpts we prepared. We aim to train `convnet-onset` and `convnet-`

Table 7-A: Statistics of the Pianoteq-rendering dataset.

Split	**Excerpts** (small version for efficient training)		**Pieces**	**Composers**
	`convnet-onset`	`convnet-segment`		
Train	54004	49908	1392	84
Valid	13536	12516		
Test	not applicable		175	28
Sum	67540	62424	1567	90

`segment` to distinguish excerpts with pedal onset and pedalled segment respectively from their associated *no-pedal* excerpts. The trained two models can be used as detectors in short-time analysis using overlapping windows to obtain local information in the test pieces with various lengths.

7.2.2 Input and Output Representations

Training CNNs to solve MIR problems is computationally intensive, therefore optimisation is necessary. Optimisation can be done by selecting an input representation which can provide audio data in an effective form. Meanwhile, training a network starting from the raw audio signal instead of its two-dimensional representations requires a larger dataset. Accordingly, 2D representations of the audio are preferred as an effective and efficient input data.

Given a large training data set consisting of 2D representations of audio excerpts in pairs, `convnet` models are expected to learn the nuances in sound played with/without the sustain pedal, while invariant to other musical elements such as pitch and loudness. Considering that the use of the sustain pedal can have effects on every piano string, this could lead to changes that affect the entire spectrum, i.e., take place at a global level. Therefore representations that reveal finer details such as short-time Fourier transform (STFT), may be redundant and inefficient for training. Mel-spectrogram is a 2D representation that approximates human auditory perception through compressing STFT in the frequency axis. This computationally efficient input has been shown to be more successful than STFT in MIR tasks such as music tagging [85]. For our case, we present the

2D representation input in pairs
conv2d & max-pooling (c, (m_c, n_c)) & (2, 2)
batch normalization - ReLU activation
conv2d & max-pooling (c, (3, 3)) & (2, 2)
batch normalization - ReLU activation
conv2d & max-pooling (c, (3, 3)) & (2, 2)
batch normalization - ReLU activation
conv2d & max-pooling (c, (3, 3)) & (4, 4)
batch normalization - ReLU activation
global average pooling
fully-connected layer
softmax output

Figure 7.2: Details of the `convnet` architecture for binary classification tasks.

effect of input representations on the performance of binary classification tasks in Section 7.3. This shows Mel-spectrogram is an adequate input representation and obtains better performance than using STFT for our tasks.

The target outputs we used to train the model are one-hot vectors which encode the two labels, i.e., *pedal* and *no-pedal*. To classify a new excerpt, the trained model can output a likelihood score for each label. The label with a higher value will be assigned to the excerpt as its classification result.

7.2.3 Models

Inspired by *Vggnet* [177] which has been found to be effective in music classification [86], our `convnet` model uses a similar architecture with fewer trainable parameters to learn the differences in time-frequency patterns in *pedal* versus *no-pedal* cases. The model consists of a series of convolutional and max-pooling layers, which are followed by one fully-connected layer with two softmax outputs. The architecture we proposed to start with and related hyperparameters are summarised in Figure 7.2, where (c, (m_c, n_c)) corresponds to *(channel, (kernel lengths in frequency, time))* specifying the convolutional layers. Pooling layer is specified by *(pooling length in frequency, time)*.

It was noted in [176] that designing filter shapes within the first layer can be motivated by domain knowledge in order to efficiently learn musically relevant time-frequency contexts with spectrogram-based CNNs. To decide (m_c, n_c) of the first layer with the best representational power, we selected their values motivated by piano acoustics and physics which can substantially change the sustain-pedal effect. Performance of `convnet` with different filter shapes within the first layer were evaluated using the validation set as discussed in Section 7.3. Apart from the common small-square filter shape, the shapes we experimented with are either wider rectangles in the time domain to model short time-scale patterns, or in the frequency domain to fit the spectral context. To be specific, according to the time-frequency transformation used in this chapter, models with the following (m_c, n_c) were trained:

- As a baseline: (3, 3) (denoted the model by `convnet-baseline`).
- For modelling larger frequency contexts: (6, 3), (15, 3), (36, 3) for STFT input, which are roughly equivalent to (9, 3), (20, 3), (45, 3) for Mel-spectrogram input (collectively denoted by `convnet-frequency`). These values of kernel length in frequency were motivated by the piano acoustics and physical structure, which fundamentally decide how the sustain-pedal effect sounds at notes of different registers. A frequency range from 0 to 283 Hz can be covered by 6 frequency bins of STFT. This corresponds to 9 Mel bands in our case when Mel-spectrogram was used as the input representation. 283 Hz approximately corresponds to the frequency of note *C4*, which is a split point between bass and treble for piano. Accordingly, (15, 3) and (36, 3), i.e., (20, 3) and (45, 3) for the Mel scale, can be separately mapped to note *D5* and *G6*. The stress bar near the strings of *D5* separates the piano frame into different regions. The strings associated with notes higher than *G6* are always free to vibrate because there are no more dampers above these strings.

- For modelling larger time contexts: (3, 10), (3, 20), (3, 30), covering 100, 200 and 300 ms respectively (collectively denoted by `convnet-time`).

The number of channels (c) was set to 21 for all the convolutional layers as a starting point. According to the best performance measurements of `convnet-frequency` and `convnet-time` respectively, we selected the corresponding (m_c, n_c) along with (3, 3) to create another model with multiple filter shapes (designated as `convnet-multi`). Outputs of the first convolutional layer were then concatenated together along the channel dimension. The rest of the `convnet-multi` architecture remained the same as the other `convnet` models.

In all the convolutional layers, batch normalisation was used to accelerate convergence. The output was then passed through a Rectified Linear Unit (ReLU) [112], followed by a max-pooling layer to prevent the network from over-fitting and to be invariant to small shifts in time-frequency. To further minimise over-fitting, global average pooling was used before the final fully-connected layer. The final layer used softmax activation in order to map the output to the range [0,1], which can be interpreted as a likelihood score of the presence of the sustain pedal in the input. We trained `convnet` with the Adam optimiser [115] to minimise binary cross entropy.

There are possibilities that simpler model architecture, i.e., with fewer channels or convolutional layers, would be sufficient for our binary classification tasks using reduced parameters. We explored the effect of the number of channels and layers in Section 7.3. With the above configurations, models dedicated to the detection of pedal onset and pedalled segment were trained using their associated excerpts in order to decide the best configuration for `convnet-onset` and `convnet-segment`, respectively.

7.3 Binary Classification Experiments

In this section, we examined whether the `convnet` models with various input representations and architectures can perform differently on discriminating *pedal* versus *no-pedal* excerpts. We first describe the experimental setup. The performance of binary classification for pedal onset and pedalled segment detection are separately presented. Finally we conduct visual analysis on convolutional layers of the trained `convnet` models to discuss what the models have learned.

7.3.1 Experimental Setup

For the input, STFT was performed using 1024-point FFT with a hop size of 441 samples (10 ms). The corresponding Mel-spectrogram was obtained using 128 Mel bands. These time-frequency transformations were done in real-time on the GPU using *Kapre* [178], which can simplify audio preprocessing and save storage. *Keras* [179] and *Tensorflow* [180] frameworks were used for the implementation.

As presented in Section 7.2.1, excerpts in pairs were split into 80%/20% to form the training/validation set, which is 54004/13536 for `convnet-onset` and 49908/12516 for `convnet-segment`. Models were trained until the accuracy no longer improved for 10 epochs. Batch size was set to 128 examples. To examine which `convnet` model can best discriminate *pedal* versus *no-pedal* excerpts, we compared AUC-ROC (or simply AUC, representing Area Under Curve - Receiver Operating Characteristic) scores of the models using the validation set.

7.3.2 Pedal Onset Detection

To decide the hyperparameters of `convnet-onset` for differentiating the excerpts with or without the existence of pedal onset, we trained `convnet` models with different configurations proposed in Section 7.2.3. Table 7-B and Table 7-C present the model perfor-

Table 7-B: Performance of different models using spectrogram input for pedal onset detection.

Model	**(m, n)**	**Accuracy**	**AUC**
`convnet-baseline`	**(3, 3)**	0.8976	0.9678
`convnet-frequency`	(6, 3)	0.8863	0.9670
	(15, 3)	0.9141	0.9735
	(36, 3)	0.9209	0.9774
`convnet-time`	(3, 10)	0.9248	0.9790
	(3, 20)	0.9307	0.9817
	(3, 30)	0.9167	0.9749
`convnet-multi`	-	0.9304	0.9801
Average	-	0.9152	0.9752

Table 7-C: Performance of different models using Mel-spectrogram input for pedal onset detection.

Model	**(m, n)**	**Accuracy**	**AUC**
`convnet-baseline`	**(3, 3)**	0.9198	0.9766
`convnet-frequency`	(9, 3)	0.9093	0.9702
	(20, 3)	0.9236	0.9758
	(45, 3)	0.9169	0.9740
`convnet-time`	**(3, 10)**	**0.9346**	**0.9827**
	(3, 20)	0.9261	0.9799
	(3, 30)	0.9198	0.9748
`convnet-multi`	-	0.9170	0.9755
Average	-	0.9209	0.9762

mance using spectrogram and Mel-spectrogram as the input representation, respectively. According to the AUC scores obtained by different (m_c, n_c) in `convnet-frequency` and in `convnet-time`, we selected the corresponding (m_c, n_c) with the highest score along with (3, 3) to create `convnet-multi`. To be specific, the first convolutional layer of `convnet-multi` consisted of (7, (36, 3)), (7, (3, 20)) and (7, (3, 3)) when spectrogram input was used. These values were (7, (20, 3)), (7, (3, 10)) and (7, (3, 3)) when Mel-spectrogram input was used. The above selected (m_c, n_c) were highlighted in the two tables.

According to the average of accuracy and AUC, Mel-spectrogram input can result in a slightly higher value. Mel-spectrogram is therefore an adequate input representation. The highest AUC of 0.9827 was also obtained using Mel-spectrogram input in the

Table 7-D: Performance of different models based on `convnet-time` ($m_c = 3$, $n_c = 10$) using Mel-spectrogram input for pedal onset detection.

`convnet-time`	c	l_c	Accuracy	AUC
Models with Reduced Parameters	3	2	0.7702	0.8500
	12	2	0.8398	0.9128
	21	2	0.8325	0.9095
	3	3	0.8036	0.8846
	12	3	0.8405	0.9184
	21	3	0.8957	0.9630
	3	4	0.8548	0.9321
	12	4	0.9171	0.9752
Original Model	21	4	0.9346	0.9827
Models with More Layers	21	5	0.9258	0.9808
	21	6	0.9301	0.9811
Models with More Channels	30	4	0.9357	0.9837
	39	**4**	**0.9383**	**0.9848**

Note: l_c denotes the number of convolutional layers.

`convnet-time` with $m_c = 3$ and $n_c = 10$ as shown in Table 7-C. Besides, better performance was obtained by `convnet-time` than `convnet-frequency` in general. We can infer that pedal onset can bring up more temporal dependencies than timbral features.

To examine the effects of the number of channels and layers, we experimented on the `convnet-time` model where $m_c = 3$, $n_c = 10$ and input representation was Mel-spectrogram. This was selected because the model's AUC is numerically better than the other models. From the results presented in Table 7-D, model performance decreased with fewer channels or layers. When the number of channels and layers were set to a larger value than the original model, the performance was more dependent on the effect of the number of channels. Models with more layers could cause overfitting problem such that the corresponding AUC based on the validation set obtained a slightly lower value. However, models with more channels obtained a higher value of AUC.

The best performing model among all the above models we experimented with is the `convnet-time` model where $m_c = 3$, $n_c = 10$, $c = 39$, $l_c = 4$ and input representation was Mel-spectrogram. The corresponding accuracy and AUC were highlighted in red in Table 7-D. This model was selected as the final `convnet-onset`, which was used as one

Table 7-E: Performance of different models using spectrogram input for pedalled segment detection.

Model	$(\boldsymbol{m_c}, \boldsymbol{n_c})$	**Accuracy**	**AUC**
`convnet-baseline`	**(3, 3)**	0.9545	0.9925
`convnet-frequency`	(6, 3)	0.9589	0.9953
	(15, 3)	0.9714	0.9969
	(36, 3)	0.9697	0.9945
`convnet-time`	(3, 10)	0.9807	0.9978
	(3, 20)	0.9845	0.9980
	(3, 30)	0.9753	0.9964
`convnet-multi`	-	0.9808	0.9980
Average	-	0.9720	0.9962

Table 7-F: Performance of different models using Mel-spectrogram input for pedalled segment detection.

Model	$(\boldsymbol{m_c}, \boldsymbol{n_c})$	**Accuracy**	**AUC**
`convnet-baseline`	**(3, 3)**	0.9755	0.9963
`convnet-frequency`	(9, 3)	0.9630	0.9905
	(20, 3)	0.9751	0.9956
	(45, 3)	0.9747	0.9968
`convnet-time`	**(3, 10)**	0.9815	0.9973
	(3, 20)	0.9787	0.9972
	(3, 30)	0.9816	0.9971
`convnet-multi`	-	**0.9837**	**0.9983**
Average	-	0.9762	0.9961

of the “pre-trained” models for frame-wise detection in Section 7.4.

7.3.3 Pedalled Segment Detection

Similar to how we compared different models for pedal onset detection, we trained `convnet` models with different configurations proposed in Section 7.2.3 in order to decide the hyperparameters of `convnet-segment` for distinguishing the pedalled segment. The performance of `convnet-baseline`, `convnet-frequency`, `convnet-time` and `convnet-multi` when the input representation was spectrogram versus Mel-spectrogram were separately presented in Table 7-E and Table 7-F. It is noted that the first convolutional layer of `convnet-multi` consisted of multiple filter shapes. In the case of spectrogram input, it consisted of (7, (15, 3)), (7, (3, 20)) and (7, (3, 3)). When Mel-spectrogram

Table 7-G: Performance of different models based on `convnet-multi` using Mel-spectrogram input for pedalled segment detection.

`convnet-multi`	c	l_c	**Accuracy**	**AUC**
Models with Reduced Parameters	3	2	0.8781	0.9486
	12	2	0.9389	0.9804
	21	2	0.9552	0.9890
	3	3	0.9436	0.9849
	12	3	0.9708	0.9948
	21	3	0.9741	0.9960
	3	4	0.9513	0.9870
	12	4	0.9762	0.9964
Original Model	21	4	0.9837	0.9983

Note: l_c denotes the number of convolutional layers.

Table 7-H: Performance of fewer-layer models based on `convnet-multi` using Mel-spectrogram input for pedalled segment detection.

`convnet-multi`	c	l_c	**Accuracy**	**AUC**
Models with More Channels and Fewer Layers	30	2	0.9522	0.9893
	39	2	0.9419	0.9929
Original Model	**21**	**4**	**0.9837**	**0.9983**

Note: l_c denotes the number of convolutional layers.

input was used, it consisted of (7, (45, 3)), (7, (3, 10)) and (7, (3, 3)). These values of (m_c, n_c) were decided because their corresponding model achieved the highest value of AUC within `convnet-frequency` and within `convnet-time` as highlighted in the two tables.

According to the AUC scores presented in Table 7-E and Table 7-F, there is not much difference between using spectrogram or Mel-spectrogram as the input to the models. AUC scores are all higher than 0.99. This informed us again that Mel-spectrogram is an adequate input representation for our detection tasks. The highest one is 0.9983 obtained by the `convnet-multi` model where $m_c = 3$, $n_c = 10$ and input representation was Mel-spectrogram. This model was selected to be trained with fewer channels and convolutional layers to examine if the same level of performance can be obtained with reduced trainable parameters.

Table 7-G presents the performance of different models based on the selected `convnet-`

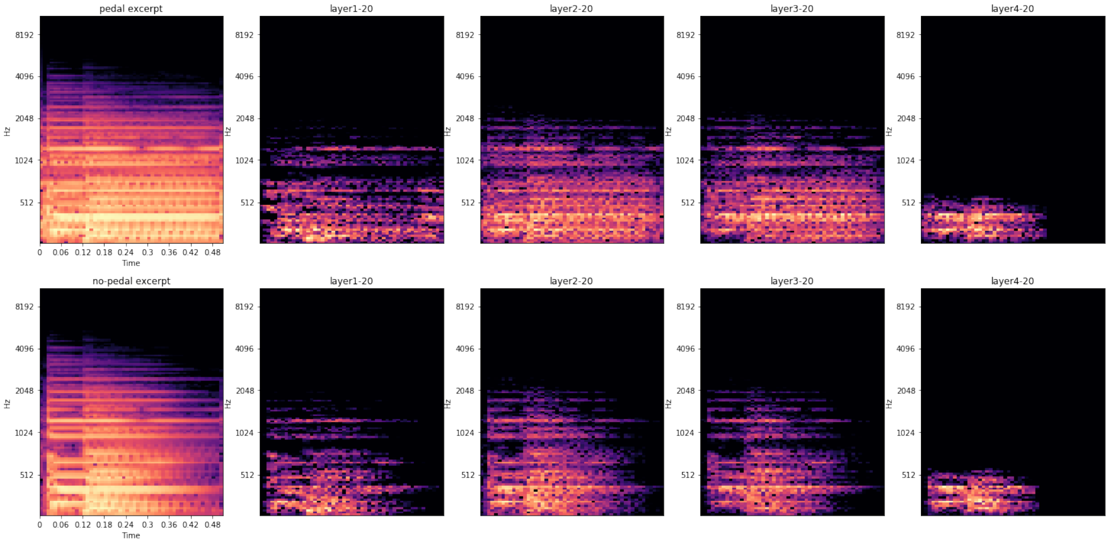


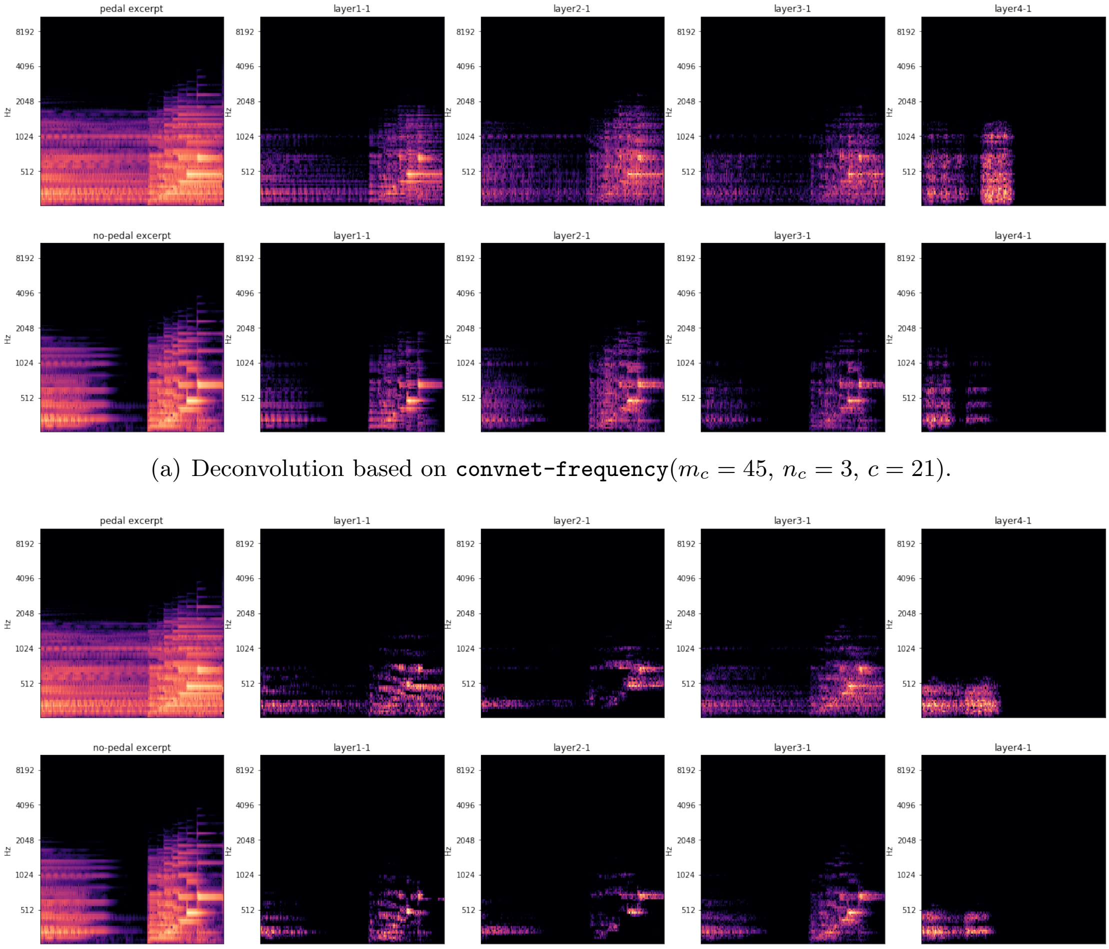
Figure 7.3: Visual analysis of music excerpts in pairs by deconvolving the layers in `convnet-onset`. The deconvolved Mel-spectrogram corresponding to the 20th kernel in layer l_c is designated by *layer*l_c*-20*.

`multi` using Mel-spectrogram input. We noticed that the model with 2 layers and 21 channels can already obtain an AUC score of almost 0.99. The performance seems to be more dependent on the effect of the number of channels. Therefore we also trained the model with 2 layers but more channels and presented the associated performance in Table 7-H. The AUC scores increased with the number of channels and achieved a value of more than 0.99 with 39 channels.

The best accuracy and AUC scores in this experiment were achieved by the selected `convnet-multi`, i.e., 0.9837 and 0.9983 as highlighted in red in Table 7-H. This model was selected as the final `convnet-segment`, which is used as a "pre-trained" model for frame-wise detection in Section 7.4.

7.3.4 Discussions

To understand our `convnet` models, one effective way is to visualise what the models have learned by deconvolution. This enables us to observe which parts of the input 2D representation are focused on by each kernel. Visualisation of CNN was first introduced



(a) Deconvolution based on `convnet-frequency`($m_c = 45$, $n_c = 3$, $c = 21$).

(b) Deconvolution based on `convnet-time`($m_c = 3$, $n_c = 10$, $c = 21$).

Figure 7.4: Visual analysis of music excerpts in pairs by deconvolving 4 layers in `convnet` models for pedalled segment. The deconvolved Mel-spectrogram corresponding to the first kernel in layer l_c is designated by *layer*l_c*-1*.

in the field of computer vision [181] to facilitate an intuitive explanation of how the shapes that kernels represent have evolved. For instance, kernels can capture simple lines in the first layer, certain shapes in the intermediate layers and finally the outlines of the target objects. In our case, we conducted a visual analysis of the deconvolved Mel-spectrogram of music excerpts in pairs, which have the same note event, but differently labelled (*pedal* versus *no-pedal*).

Given the trained `convnet-onset` with 39 kernels in each layer, it can determine the existence of pedal onset in an excerpt of 0.5 seconds at high accuracy and AUC as presented in Section 7.3.2. Visualisation results of each convolutional layer using the 20th kernel as an example are shown in Figure 7.3, where the ground-truth pedal onset takes place at the 0.2 seconds. From the first to the fourth layer, we can observe kernels tend to focus on the frequency bands with larger magnitude within the frames around the pedal onset time. We could infer that the transient brought by the pedal onset has more effects on these frequency bands. This can be captured by `convnet-onset` for the detection.

For `convnet` models dedicated to capturing the acoustic characteristics when the sustain pedal stays pressed, the models with 4 layers can all obtain an AUC score higher than 0.99 as shown in Table 7-E and Table 7-F. We assume that pressing the sustain pedal could result in acoustic characteristics that significantly change the patterns in both frequency and time. Thereby the `convnet-frequency` and `convnet-time` can be both comparably favourable. Their associated first kernel in each convolutional layer that responds to what part in the Mel-spectrogram is presented in Figure 7.4. From layer 1 to 3, the two models both focus on the time-frequency contexts centred around the fundamental frequency and their partials. More contexts in the higher frequency bands can be learned by the `convnet-frequency`. In the fourth layer, only the first half of Mel-spectrograms are emphasised. We could infer the sustain-pedal effect is more significant on the notes which the pedal just started to play with. Meanwhile, main differences between *pedal* and *no-pedal* excerpts lie in the lower frequency bands indicated by the `convnet-time`. Considering a slightly lower accuracy score was obtained by `convnet-frequency`, dependencies within the higher frequency range could be a redundant knowledge to learn.

Through inspecting the detection results and the learned filters, we can extend our understanding of the `convnet` models in music. In the following section, the trained models are applied to music pieces in order to point our which frames were played with the sustain pedal.

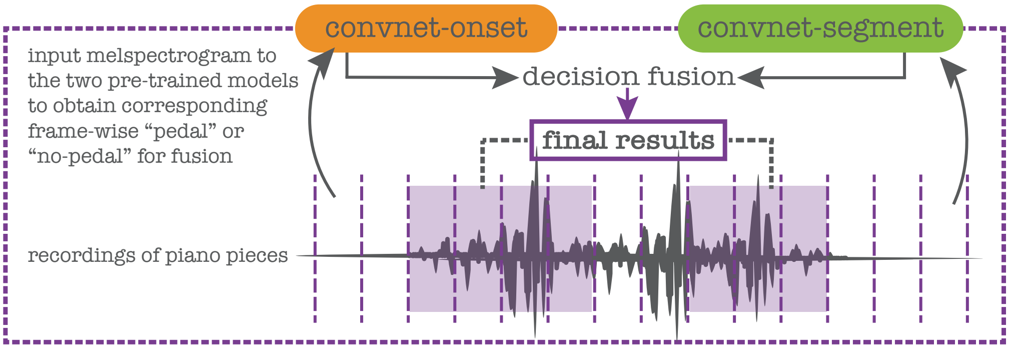

convnet-onset
convnet-segment
input melspectrogram to the two pre-trained models to obtain corresponding frame-wise "pedal" or "no-pedal" for fusion
decision fusion
final results
recordings of piano pieces

Figure 7.5: Framework of the proposed decision fusion method.

7.4 Frame-Wise Detection

In this section, we proposed two methods of frame-wise detection to identify how the sustain pedal was used in a recording of piano piece. Based on the performance measurement of the investigated models, we can decide `convnet-onset` and `convnet-segment`, which obtained the highest AUC score presented in Section 7.3.2 and Section 7.3.3, respectively. The two models can be used as detectors to jointly decide the *pedal* frames in a piano piece using heuristics. This detection strategy is known as "decision fusion", which is presented in the following Section 7.4.1. Given that our `convnet` models were trained from the synthesised data, adapting the learned knowledge encoded in the `convnet` to the detection task where the recording instruments and room acoustics are different is essential to guarantee the performance. We can approach this using transfer learning in Section 7.4.2.

7.4.1 Decision Fusion

7.4.1.1 Method

The concept of fusion has been adopted in MIR tasks such as note onset detection [182]. Fusion can take place at different stages during the detection process. It can either combine different features to better represent the signal or combine the results from multiple detectors. The latter strategy is known as "decision fusion". In our case, we opted for decision fusion to combine the pedal onset detected by `convnet-onset` and the pedalled segment detected by `convnet-segment` in order to refine the *pedal* portions in a piano recording. This framework is illustrated in Figure 7.5, where the final results express a partitioning of the input audio data into intervals played with the sustain pedal.

We first used `convnet-onset` and `convnet-segment` as detectors in short-time analysis using overlapping windows to obtain the corresponding local information in a piece. Since we trained the `convnet` models using softmax activation at the last layer, the local information is a likelihood score of the presence of the pedal onset or pedalled segment at the current frame. A binary decision can be made by thresholding the score at a value, which is equal to 0.5 by default. The threshold value can also be informed by the average value of the scores from excerpts correctly classified as *pedal* in the validation set of the binary classification experiments in the previous Section 7.3. This is expected to make a binary decision at higher precision. We examine the effects of the threshold value in the following experiment.

Detection was then reinforced by decreasing the rate of false positives through fusion, which is described by Algorithm 2. The underlying hypothesis is that the inferences made by `convnet-segment` can be assured by `convnet-onset`, because the starting point of an interval played with the sustain pedal should have a pedal onset detected. Let ST_{seg} be the lists of pedal-segment starting times in second produced by the `convnet-segment` detector, ET_{seg} be the associated ending times, and T_{ons} be the list of pedal-onset times in second produced by the `convnet-onset` detector. ST_{pedal} and ET_{pedal} are the lists of final detection results that respectively imply the onset and offset times of the sustain pedal.

According to the ground-truth annotation, we can evaluate the performance of the decision fusion method by two criteria: classification evaluation metrics and boundary detection metrics. More details on evaluation metrics for structural segmentation were

Algorithm 2 Decision fusion

```
Require: $\tau$: the tolerance time window
  procedure FUSION($ST_{seg}$, $ET_{seg}$, $T_{ons}$)
      for all $j \in \{0, ..., len(ST_{seg}) - 1\}$ do
          for all $i \in \{0, ..., len(T_{ons}) - 1\}$ do
              if $abs(T_{ons}[i] - ST_{seg}[j]) < \tau$ then
                  $ST_{pedal} \leftarrow append(T_{ons}[i])$
                  $ET_{pedal} \leftarrow append(ET_{seg}[j])$
      return $ST_{pedal}$, $ET_{pedal}$
```

introduced in Section 2.5.

7.4.1.2 Experiment

As introduced in Section 4.4, the test set includes 173 pieces, which were rendered by Pianoteq using the same settings for generating the excerpts in the train/validation set. We applied sliding windows to a test piece in order to get outputs of the two trained models separately at every frame. The window for `convnet-onset` covers a duration of 0.5 seconds, with a hop size equivalent to 0.01 seconds. For `convnet-segment`, the window corresponds to 0.3 seconds with a hop size of 0.1 seconds. Then the 0.3-second samples were tiled to 2 seconds such that the input size was coherent with the one in the training phase.

Binary decisions (*pedal*/*no-pedal*) were made by thresholding the outputs of `convnet-onset` and `convnet-segment` at $Thre_{onset}$ and $Thre_{segment}$, respectively. It is tempting to assume that the decision threshold should always be 0.5, but thresholds are problem-dependent. In our case, mistakenly labelling a pedal-off frame as pedal-on is undesirable, given the fact that overusing the sustain pedal can mix up all the notes and blur the sonorities in a performance. On the contrary, failing to identify a frame as pedal-on is unpleasant, but the intended effect is still possible to obtain. We set $Thre_{onset}$ and $Thre_{segment}$ to 0.98, which is the average value of the softmax output from excerpts accurately classified as *pedal* in the validation stage in Section 7.3.

Following the decision fusion policy in Algorithm 2, we first located portions that had more than three frames continuously considered as *pedal* by `convnet-segment`. If `convnet-onset` also returned *pedal* within 0.1 second around the beginning of a portion, the sustain pedal was detected as *on* in the frames within this portion. The rest of the frames were assigned to *off*. We finally obtained frame-wise *on/off* results for a piece. We used classification evaluation metrics that include pairwise precision, recall and F-measure scores with respect to label *on* to evaluate our method (designated as P_1, R_1 and F_1). Considering the imbalanced occurrence counts of the two labels, micro-averaged F-measure (F_{micro}) was selected to represent the overall performance.

Moreover, frame-wise results can be processed into intervals. Each interval indicates the start and end time of a pedal event. This can be evaluated by the boundary detection metrics [183], including boundary detection precision, recall and F-measure scores (designated as P_b, R_b and F_b). An estimated boundary is considered correct if it falls within a window around a reference boundary. The window was decided by the estimated global tempo in each piece. Its average value from the 173 test pieces corresponds to 0.48 seconds.

The proposed decision-fusion-based detection method was evaluated on every piece in the test set using the classification evaluation metrics and boundary detection metrics. We compared the results when $Thre_{onset}$ and $Thre_{segment}$ were set to 0.5 versus 0.98. Another experiment without fusion was conducted by using the `convnet-segment` output only to obtain frame-wise *on/off*. This can inform us that the fusion strategy could facilitate our sustain-pedal detection task to what extent.

7.4.1.3 Result and Discussion

We obtained the evaluation measures for every piece in the test set using the method based on fusion and `convnet-segment`, respectively. The average scores over the 173 pieces are presented in Table 7-I. If we consider structure annotation metrics, directly applying the pre-trained `convnet-segment` by thresholding its output at 0.98 can lead

Table 7-I: Average performance of the proposed decision-fusion-based detection method versus the method using `convnet-segment` only.

Metrics	**Fusion** ($Thre_{onset}$, $Thre_{segment}$)		`convnet-segment` $Thre_{segment}$	
	(0.5, 0.5)	(0.98, 0.98)	0.5	0.98
P_1	0.7373	**0.8572**	0.7350	0.7813
R_1	0.9485	0.6655	**0.9748**	0.9267
F_1	0.8092	0.7422	0.8150	**0.8328**
F_{micro}	0.7928	0.7361	0.8021	**0.8197**
P_b	0.7738	0.7988	0.7664	**0.8018**
R_b	0.3557	**0.5237**	0.3624	0.4869
F_b	0.4301	**0.6164**	0.4300	0.5701

to a frame-wise binary decision with the highest average scores of F_1 and F_{micro}. The decision fusion method can achieve a comparable F_{micro} when the default value of the decision threshold was used. When we set the threshold value to 0.98, the continuity of each individual pedal event can be enforced. The highest P_1 but a significantly decreased R_1 were obtained. This fusion strategy doesn't result in favourable pairwise measures, however, produces superior performance when boundary detection metrics are used for evaluation. This suggests decision fusion can serve as a post-filtering method, which is useful to reduce fragmentation caused by false positives. Its pairwise performance is possible to be increased if a fine-tuned threshold value is used.

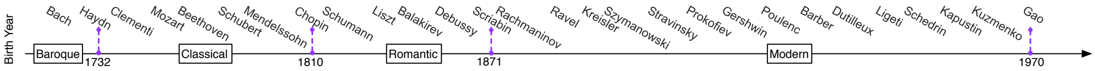


Figure 7.6: Composers arrangement in a chronological order based on their birth year.

To take a detailed look at how the two methods perform on the pieces by different composers, we presented F_{micro}, F_1 and F_b as box plot annotated with their associated median value. The composers are arranged in a chronological order based on their birth year as shown in Figure 7.6. According to the average performance in Table 7-I, we selected the box plots corresponding to the methods with a threshold value of 0.98 to discuss. Figure 7.7 and Figure 7.8 present the performance using the decision-fusion-based and `convnet-segment`-based method, respectively. The percentage of the *on* frames according to the ground truth and the number of pieces associated with each composer are also shown.

In general, both methods work best for the pieces around the Romantic era (from Chopin to Scriabin), when modern pedalling techniques appear to have been established and widely used by pianists. The `convnet-segment`-based method inclines to detect a frame as *on*. Accordingly, it obtained higher pairwise F1 than the decision-fusion-based method, especially on the pieces with a larger percentage of frames played with the sustain pedal. However, it also leads to increased false positive rate, which has more negative effects on the pieces that rely less on the sustain pedal in performance. For these pieces such as the ones by Bach and Mozart, the overall pairwise performance measured by micro-F1 obtains a higher score by the decision-fusion-based method. This method also commonly obtains a higher F1 score in pedal boundary detection, i.e., F_b, except on a few pieces by composers in the Modern era.

To sum up, our proposed fusion method can effectively reduce the false positive rate. This benefit the sustain-pedal detection on the pieces in the Baroque era, when pedalling techniques are rarely used. For the pieces commonly played with the sustain pedal, such as the ones in the post-Classical and Romantic era, the fusion strategy should be adjusted in order to obtain a comparable pairwise performance by the `convnet-segment`-based method.

7.4.2 Transfer Learning

7.4.2.1 Method

To apply a `convnet` trained from the synthesised data into the context of real recordings, we can use transfer learning as illustrated in Figure 7.9. Transfer learning exploits the knowledge gained during training on a source task and applies this to a target task [119] (see Section 2.3.4.3 for technical background). This is crucial for our case, where the target-task data is obtained from recordings of a different piano, therefore it is difficult to

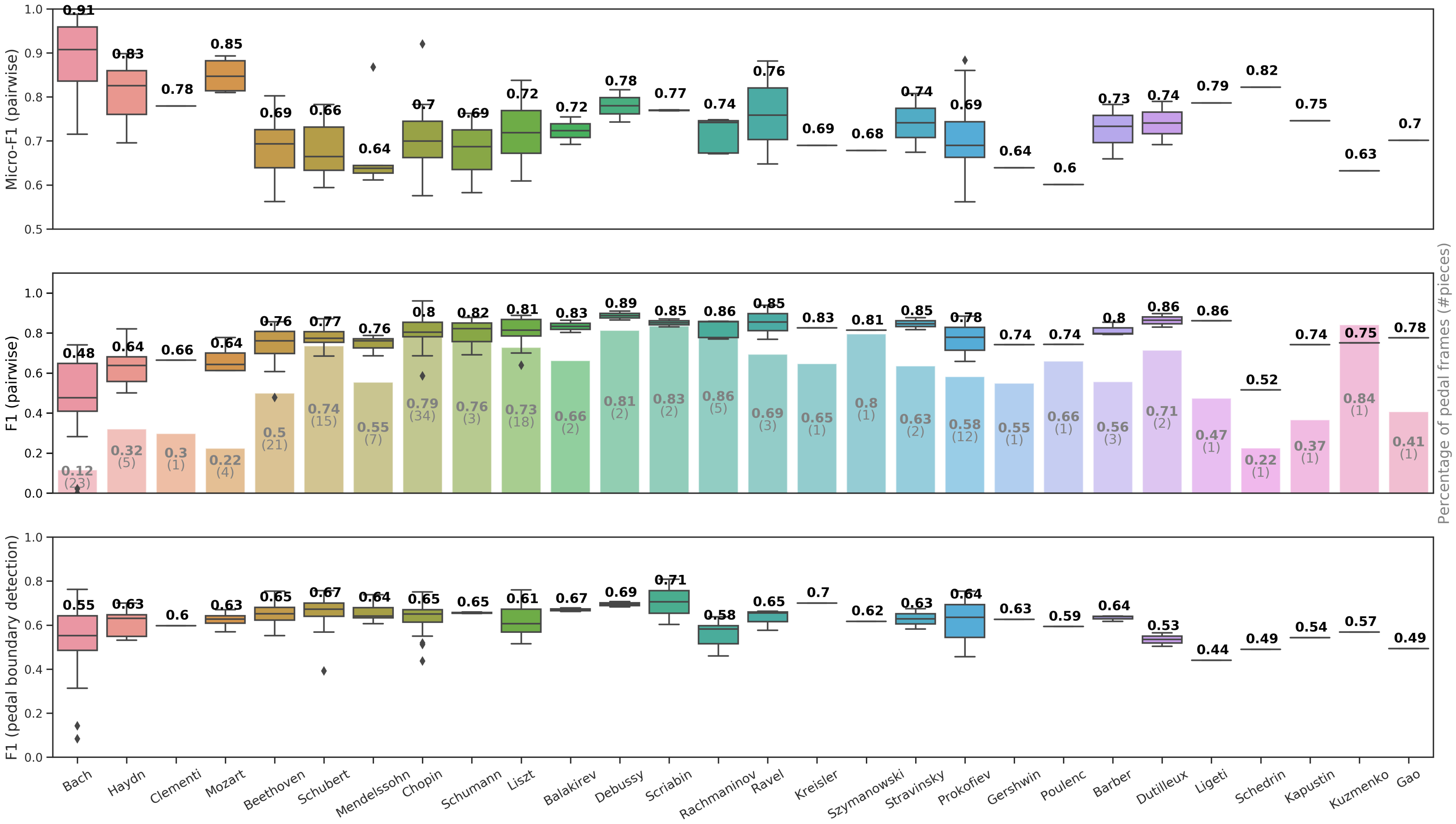


Figure 7.7: Box plot of evaluation measures using the decision-fusion-based method ($Thre_{onset} = 0.98$, $Thre_{segment} = 0.98$) and bar plot of *pedal*-frame proportion for the pieces by different composers.

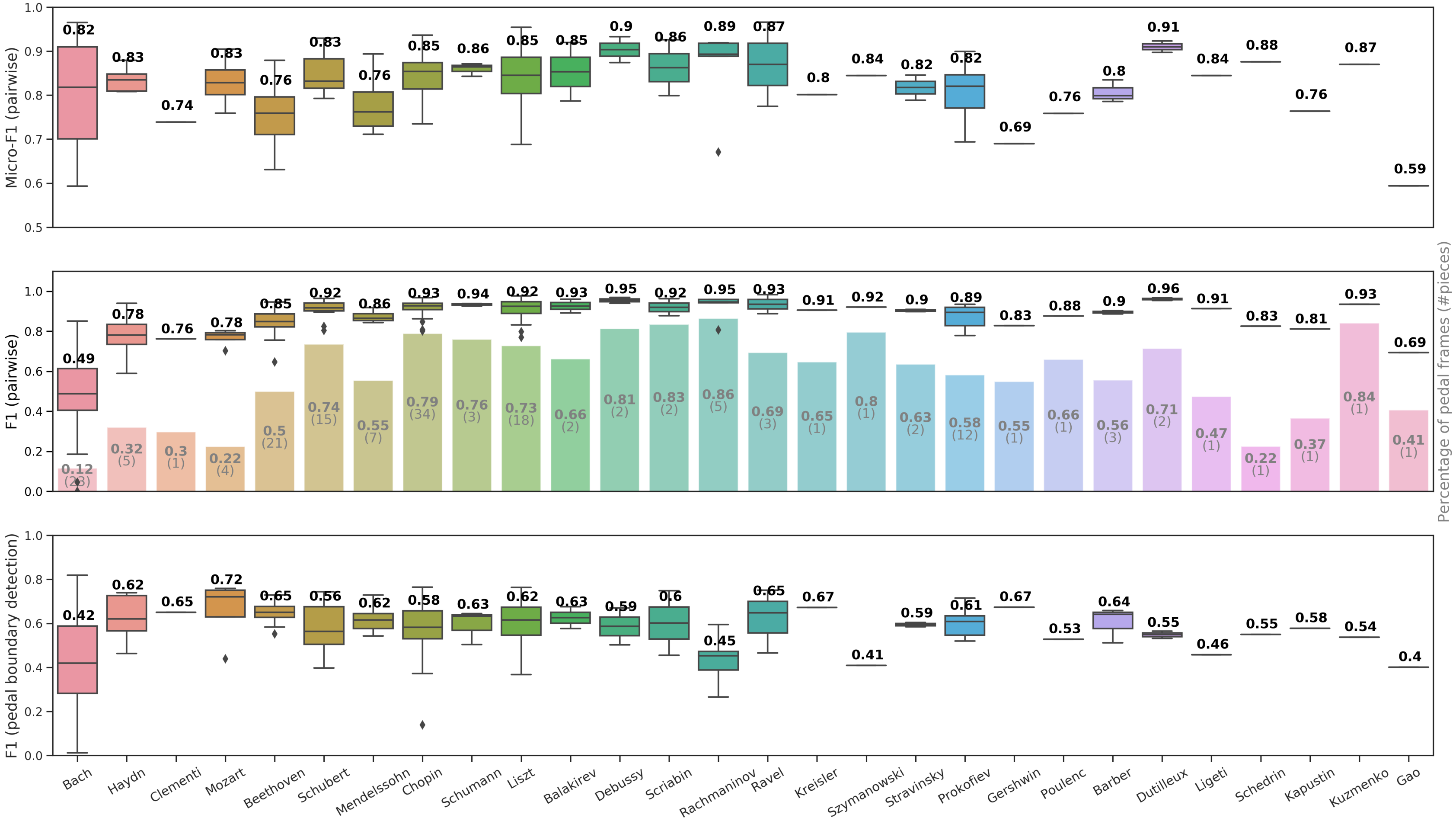


Figure 7.8: Box plot of evaluation measures using the `convnet-segment`-based method ($Thre_{segment} = 0.98$) and bar plot of *pedal-frame* proportion for the pieces by different composers.

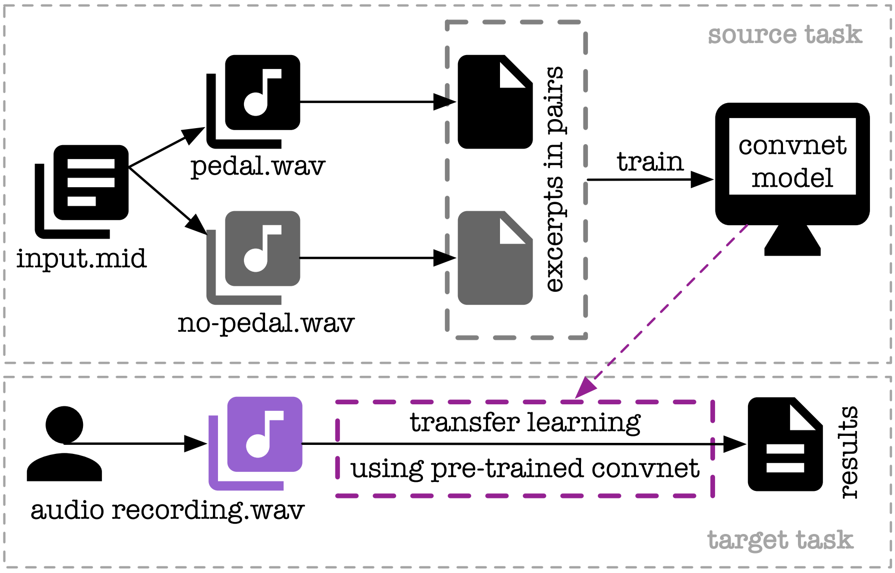


Figure 7.9: Framework of the proposed transfer learning method.

learn a "good" representation due to mechanical and acoustical deviations. The source task is to train a `convnet` model which can distinguish synthesised music excerpts with or without the sustain-pedal effect. This has been completed in Section 7.3. Then in the target task, we can use the learnt representations from the trained `convnet` as features, which are extracted from every frame of a real piano recording. These features help to train a dedicated classifier adapted to the actual acoustics of the piano and the performance venue used in the recording. With the new feature representation, the proposed transfer learning method is expected to better identify frames played with the sustain pedal.

Given that the target-task data consists of ten well-known passages of Chopin's piano music as introduced in Section 4.2, the `convnet-segment` was selected as the pre-trained model. This is because the `convnet-segment`-based method obtains a superior performance in the sustain-pedal detection task on the synthesised pieces by the Romantic-era composers in the previous Section 7.4.1. The hierarchical features from the pre-trained `convnet-segment` represent acoustic characteristics when the sustain pedal of a virtual piano is played in a certain recording environment. We can use the following two methods of feature representation transfer to facilitate the target task, i.e., sustain-pedal

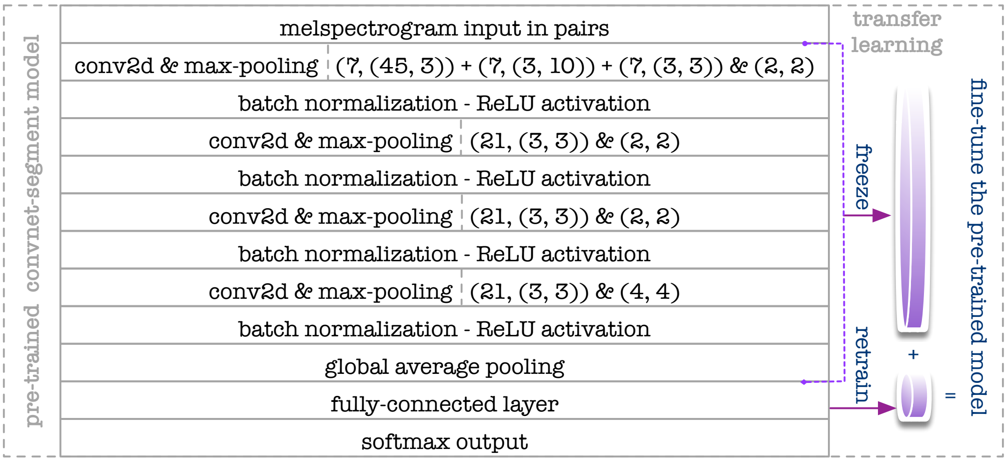

pre-trained convnet-segment model
melspectrogram input in pairs
conv2d & max-pooling (7, (45, 3)) + (7, (3, 10)) + (7, (3, 3)) & (2, 2)
batch normalization - ReLU activation
conv2d & max-pooling (21, (3, 3)) & (2, 2)
batch normalization - ReLU activation
conv2d & max-pooling (21, (3, 3)) & (2, 2)
batch normalization - ReLU activation
conv2d & max-pooling (21, (3, 3)) & (4, 4)
batch normalization - ReLU activation
global average pooling
fully-connected layer
softmax output
transfer learning
freeze
retrain
+
=
fine-tune the pre-trained model

Figure 7.10: A schematic of transfer learning by fine-tuning the last fully-connected layer.

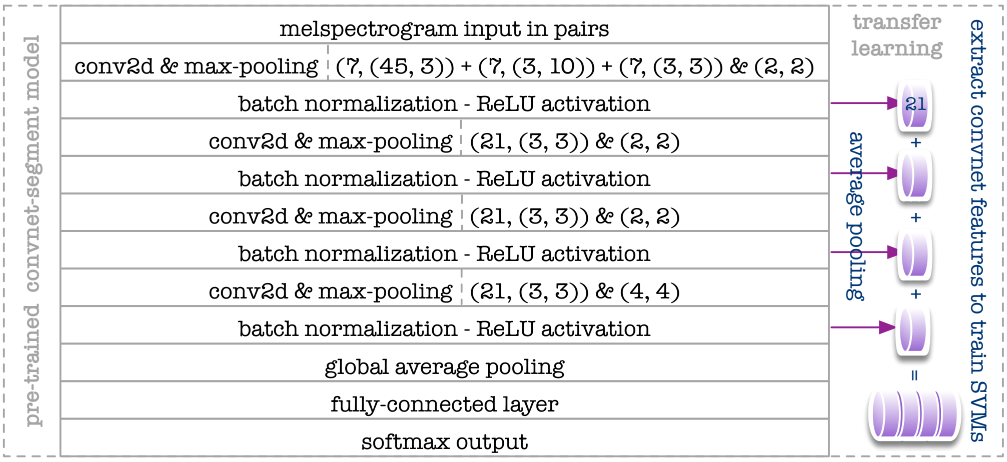

pre-trained convnet-segment model
melspectrogram input in pairs
conv2d & max-pooling (7, (45, 3)) + (7, (3, 10)) + (7, (3, 3)) & (2, 2)
batch normalization - ReLU activation
conv2d & max-pooling (21, (3, 3)) & (2, 2)
batch normalization - ReLU activation
conv2d & max-pooling (21, (3, 3)) & (2, 2)
batch normalization - ReLU activation
conv2d & max-pooling (21, (3, 3)) & (4, 4)
batch normalization - ReLU activation
global average pooling
fully-connected layer
softmax output
transfer learning
21
+
+
+
=
average pooling
extract convnet features to train SVMs

Figure 7.11: A schematic of feature extraction procedures for transfer learning with SVMs.

detection of a specific piano in a real scenario:

1. The `convnet-segment` model can be fine-tuned by retraining the last fully-connected layer only, which is commonly considered a basic transfer learning technique. This was used as a baseline method as illustrated in Figure 7.10.

2. The activations of each intermediate layers of `convnet-segment` can be subsampled using average pooling and then concatenated into features as illustrated in Figure 7.11. Here average pooling can summarise the global statistics and reduce the size of feature maps to a vector of length associated with the number of chan-

nels. In the end, a 21 × 4 dimensional feature vector was generated since there are 4 convolutional layers in the `convnet-segment`. SVMs were then trained with these features into a detector using a supervised learning method. We opted for SVMs first because the features extracted from the carefully-trained model in the source task should be representative and separable. Second, the SVM algorithm was originally devised for classification problems, involving finding the maximum margin hyperplane that separates two classes of data [142].

In the target task, outputs of the above methods were obtained from short-time sliding windows over the Mel-spectrogram of a passage. The output of each audio frame corresponds to an estimated pedal state (*on* or *off*). This can be evaluated according to the frame-wise ground truth. As shown in the following sections, the proposed transfer learning method with SVMs overall outperformed using the pre-trained `convnet-segment` with a fine-tuned last layer.

7.4.2.2 Experiment

Similar to the experimental settings in Section 7.4.1, Mel-spectrograms with 128 Mel bands were extracted from excerpts to serve as input to the network, The processing was done in real-time on the GPU using *Kapre* [178], which can simplify audio preprocessing and saves storage. Time-frequency transformation was performed using 1024-point FFT with a hop size of 441 samples (10 ms). *Keras* [179] and *Tensorflow* [180] frameworks were used for the implementation.

The source task was identical to the CNN training in Section 7.2. The `convnet-segment` trained in Section 7.3.3 was chosen as the pre-trained model. In the target task, a sliding window was applied to the acoustic piano recordings in order to extract features of the pre-trained model at every frame. The window covers a duration of 0.3 seconds with a hop size equivalent to 0.1 seconds. The 0.3-second samples were then tiled to 2 seconds and transformed into Mel-spectrogram such that the input size was coherent with the one in the source task. For our proposed transfer learning method with SVM, the extracted features were used to train the SVM constructed by *Scikit-learn* [139].

The experiment was done by conducting *leave-one-group-out* cross-validation, where samples were grouped in terms of music passages. The performance of the proposed transfer learning method was validated in each music passage where the frame-wise features need to be classified by the SVM into pedal *on* or *off*, while the rest of the passages constitute the training set. The SVM parameters were optimised using grid-search based on the validation results. The radial kernel was used, and the parameters were selected from the ranges below:

- γ: [$1/2^3$, $1/2^5$, $1/2^7$, $1/2^9$, $1/2^{11}$, $1/2^{13}$, 1/*feature vector dimension*]
- C: [0.1, 2.0, 8.0, 32.0]

We compared the proposed transfer learning method with the detection using a fine-tuned `convnet-segment` model, which can serve as a baseline classifier. Within each cross-validation fold, the fully-connected layer was updated until the accuracy stopped increasing for 10 epochs. Then we obtained the fine-tuned `convnet-segment` outputs from short-time sliding windows over the Mel-spectrogram of the validation passage.

Given the frame-wise *on*/*off* results for every music passage, we calculated precision (P_1), recall (R_1) and F-measure (F_1) with respect to the label *on*. We also compared the overall performance of the two methods along with directly using the pre-trained `convnet-segment`. Considering the imbalanced occurrence counts of the two labels, the micro-averaged F-measure (F_{micro}) was selected to evaluate the overall performance, because it calculates metrics globally by counting the total true positives, false positives and false negatives with respect to both labels of *on* and *off*.

Table 7-J: Performance of the two transfer learning methods in the target task.

Passages	**Retrain Last Layer Only**			**Transfer Learning with SVM**		
	P_1	R_1	F_1	P_1	R_1	F_1
Op.10 No.3	0.7615	0.9965	0.8633	0.8457	0.9941	**0.9139**
Op.23 No.1	0.6670	0.8573	0.7503	0.8643	0.9349	**0.8982**
Op.28 No.4	0.7569	0.9698	0.8502	0.8148	0.9859	**0.8922**
Op.28 No.6	0.7357	0.9607	0.8332	0.8178	0.9569	**0.8819**
Op.28 No.7	0.8217	0.8866	0.8529	0.8971	0.8385	**0.8668**
Op.28 No.15	0.6659	0.9329	0.7771	0.8412	0.9624	**0.8977**
Op.28 No.20	0.7405	0.9949	0.8490	0.7849	0.9974	**0.8785**
Op.66	0.7720	0.9439	0.8494	0.9425	0.9439	**0.9432**
Op.69 No.2	0.7622	0.9272	0.8366	0.9649	0.7902	**0.8688**
B.49	0.7091	0.9172	0.7998	0.8175	0.9919	**0.8963**
Average	0.7392	0.9387	0.8262	0.8591	0.9396	**0.8938**

7.4.2.3 Result and Discussion

Table 7-J presents the performance measurement of the two transfer learning methods respectively for every validation passage in the cross-validation fold. In general, our proposed transfer learning method with SVM obtains better performance. The associated average value of P_1, R_1 and F_1 are 11.99%, 0.9% and 6.76% higher than using the transfer learning method with the fine-tuned `convnet-segment`. We also compared the two methods with directly using the pre-trained `convnet-segment` model. Their F_1 and the overall performance (F_{micro}) are presented passage by passage in Figure 7.12. We can observe that the transfer learning method with SVM presents the best overall performance with more than 10% higher than the F_{micro} obtained by the other two methods. Moreover, methods based on fine-tuning versus directly applying the `convnet-segment` yield comparable performance. This implies our pre-trained model successfully captures sustain-pedal-related acoustic characteristics that are shared across virtual and real pianos. However, relying on the output from the pre-trained model only can be inadequate when the sustain-pedal detection aims at real piano recordings.

To gain a straightforward insight into the pros and cons of the transfer learning method with SVM, we visualised the detection results in the passage of *Op. 66*, which

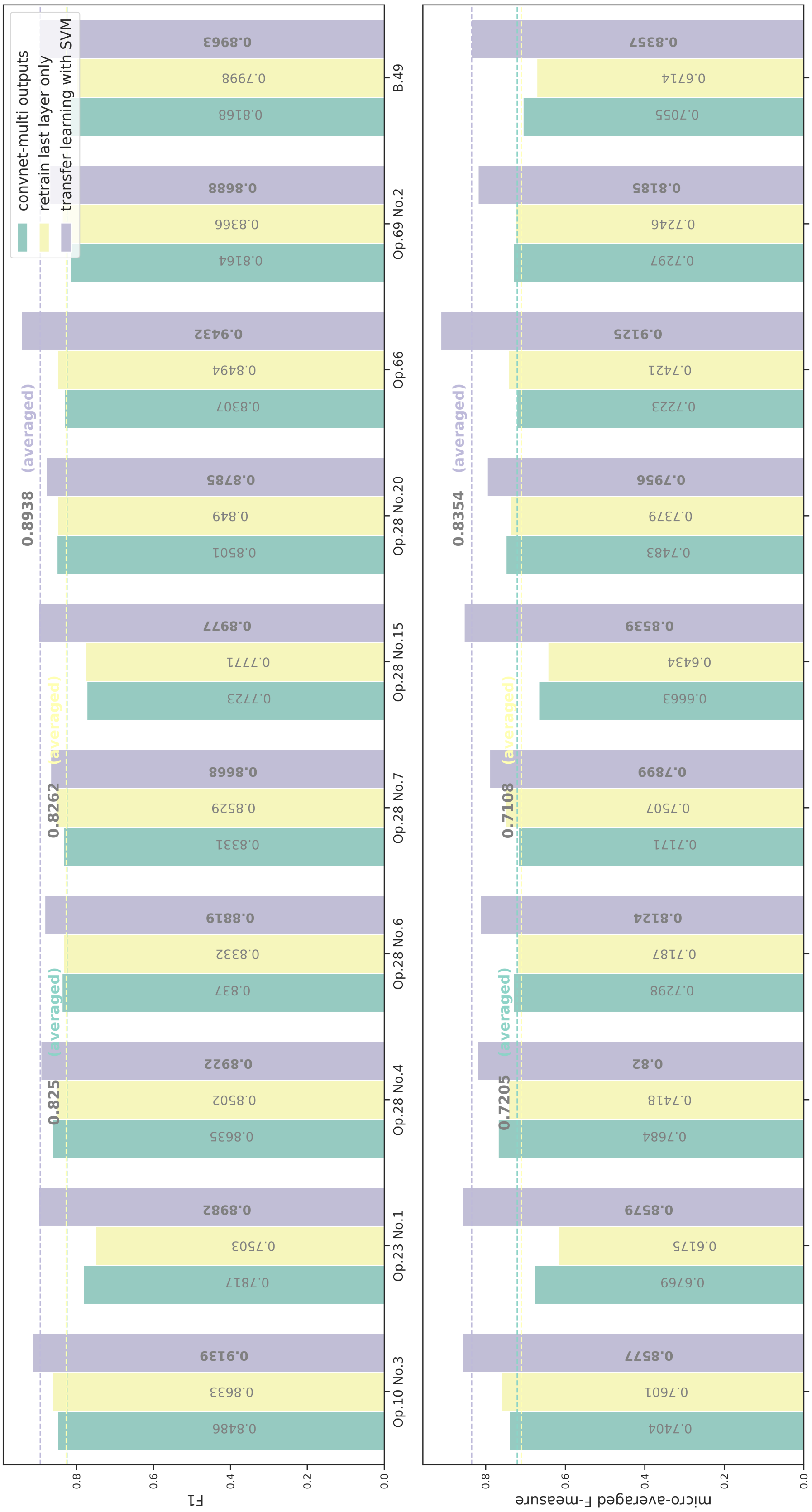


Figure 7.12: F_1 and overall performance (F_{micro}) of the three methods in the target task.

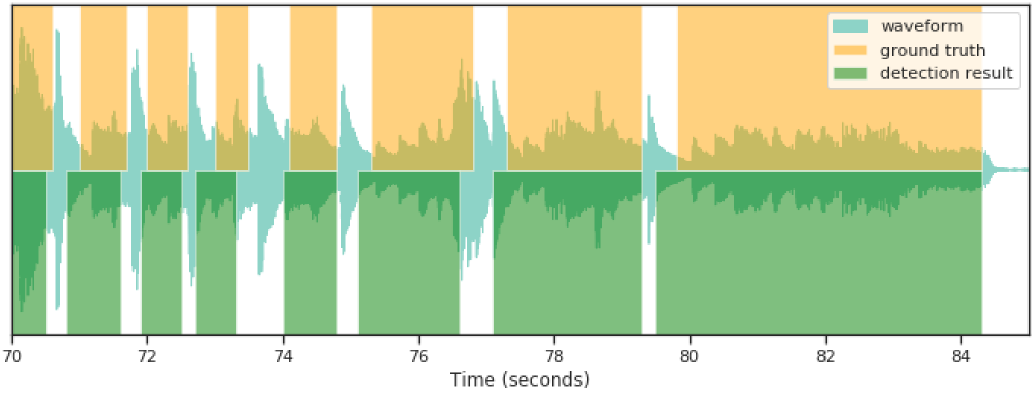


Figure 7.13: Visualisation of the ground truth (top row) and the detection result (bottom row) in *Op. 66*. Audio frames that are annotated/detected as pedal *on* are highlighted in orange/green.

obtained the highest score of F_1 and F_{micro}. Figure 7.13 presents the last 15 seconds of the passage as an example, where the portions highlighted in orange/green correspond to the audio frames annotated/detected as pedal *on*. Most of the frames were correctly identified. Yet, there were false positives and false negatives around the true sustain-pedal onset and offset times. One possible solution is to use the fusion method as proposed in Section 7.4.1 in order to localise the pedal boundary with better precision. However, as also pointed out in Section 7.4.1, the fusion strategy should be carefully-designed to guarantee the recall. How to better deal with the transients introduced by the pedal changes remains a question here. Moreover, more audio data including pieces by other composers and using various recording conditions should be tested to verify the robustness of the proposed transfer learning methods. This also constitutes our future works.

7.5 Summary

In this chapter, deep learning methods were applied to help detecting sustain-pedal techniques in polyphonic piano music. We first took advantage of CNNs to learn the time-frequency contexts corresponding to acoustic characteristics of the sustain pedal, instead of larger variations introduced by other musical elements. The CNN can be designed through exploiting the knowledge of piano acoustics and physics in Section 7.2, and trained as binary classifiers using excerpts in pairs (*pedal* versus *no-pedal*) in Section 7.3. The resulting models of `convnet-onset` and `convnet-segment` can capture the nuances of two phases of pressing the sustain pedal, i.e., at the start versus in the process of a pedal event.

To answer the question: "Can a computer point out pedalling techniques when a piano recording from a virtuoso performance is given?", we used the pre-trained models to facilitate the frame-wise detection in Section 7.4. We first proposed a decision-fusion-based method, which is useful for indicating onset and offset times of the sustain pedal. However, this method was implemented on a synthesised dataset. The reduced acoustic complexity may lead to generalisation issues on detecting the sustain pedal from real piano recordings. We therefore proposed to adapt the pre-trained models to the real-world scenarios using transfer learning. Features with more representation power dedicated to the sustain-pedal effect of an acoustic piano can be extracted from the intermediate layers of `convnet-segment`. SVMs trained with these features can identify frame-wise pedal *on/off* state in each test piece at higher precision. Thus better performance was obtained compared to fine-tuning or directly applying the pre-trained `convnet-segment` model.

Chapter 8

Conclusions

8.1 Achievements

This thesis presented an in-depth analysis of the acquisition of pedalling gestures and techniques in piano performances. We emphasised the gestures on the sustain pedal, which is commonly used by pianist for seamless legato and the enrichment of sound. The gestures are categorised into pedalling techniques in terms of their onset time and depth of the sustain pedal. These pedalling techniques along with note dynamics and timing constitute the main control parameters for expressive piano performances. It has been shown that professional pianists adapt their performance controls to different room acoustics and pianos [184, 185]. Automatic retrieval of control parameters can therefore reveal the secrets of virtuoso performances. It is noted that pedalling techniques lead to rather subtle nuances in sound and are considered challenging to be detected from audio alone. This thesis is the first study that achieves sustain-pedal detection from polyphonic piano audio recordings. Such indirect acquisition methods are evaluated by the datasets detailed in Chapter 4. In our datasets, pedalling techniques can be annotated by direct acquisition systems.

Apart from using specific reproducing pianos like Disklavier or virtual pianos like

Pianoteq to form our dataset, a dedicated direct acquisition system was designed for dataset construction using any acoustic piano in Chapter 3. The system used near-field optical reflectance sensing, enabling the pedalling gesture to be captured in a non-intrusive way. Pedalling techniques can be directly detected using the captured gesture data. Detection results feature the temporal locations of pedalling events and the employed technique within each event. Among the common supervised learning classifiers, SVM was trained with the best classification performance. It can categorise the continuous gesture data in each pedalling event into four pedalling techniques related to pedal depth: *quarter pedal*, *half pedal*, *three-quarter pedal* and *full pedal*. They are the four levels widely used to detail the part-pedalling technique. Our visualisation application can present the detection results together with the corresponding audio recording in a score following system. Here, the audio recording was simultaneously recorded with the gesture data into the embedded platform of the system. This dedicated system can therefore be used to form another dataset consisting of acoustic piano recordings to examine the performance of deep learning models originally trained on Pianoteq-generated data in Chapter 7.

To develop indirect acquisition methods, Chapter 5 presented a study using isolated piano tones as a starting point. Each tone can be decomposed into sinusoidal and corresponding residual components. Especially the method modelling the sinusoidal components was informed by piano acoustics, which took inharmonicity into consideration. Different pedalling techniques alter the decay patterns of these two components, respectively. We modelled their decay patterns and used the resulting coefficients to form a feature vector as a representation of a tone played normally or with a pedalling technique. Decision-tree-based SVM was trained with the extracted features to classify the tones into *normal*, *non-legato over-half*, *non-legato half*, *legato over-half* and *legato half* pedalling techniques. The cross-validation experiments obtained good performance, indicating the effectiveness of the proposed features and classifier. It can be observed from the results that notes played with half pedalling obtain a faster decay rate of the first partial than the ones with full pedalling. The moment when the sustain pedal is pressed leads to a transient which can be seen in the magnitude evolution of residuals. Such transient happens more often when half pedalling is used. Although most features are extracted from isolated tones and hence too clean to be applied in a more generic case, the most significant peak in the residual decay when the legato pedalling technique was used inspired our study on legato-pedal onset detection from polyphonic music.

Accordingly, a method for detecting legato-pedal onset was proposed based on a measure of sympathetic resonance from the residuals in Chapter 6. This is because sympathetic resonance is suddenly enhanced at the moment when the legato pedalling technique is used. The extent of sympathetic resonance is associated with the energy of unstruck strings. To decide which strings were unstruck, the sounding strings were first identified using NMF-based piano transcription, in which the templates were obtained using the 88 tones of the same piano. These templates also returned a more accurate partial estimation for each tone. By combining the results from piano transcription and partial estimation, residuals were obtained by removing the partials of the transcribed notes from their detected onset until offset times. Residuals were divided into segments according to the detected note onset times. In a segment, notes associated with unstruck strings were determined by the preceding notes that are beyond the time range between their detected onset and offset times. Sympathetic resonance can be measured by the root-mean-square energy of these notes. The existence of legato-pedal onset per segment was determined using a logistic regression classifier trained on two features based on the sympathetic resonance measure. One feature is the maximum energy of the current segment; the other one represents the distance between the maximum-energy location and the segment starting point. Segment-wise results indicate whether a note onset was followed by a legato-pedal onset or not.

Yet, the proposed method using the sympathetic resonance measure was only aimed at the detection of legato-pedal onset, which is essential but served as one specific technique on the sustain pedal. Moreover, this method relied on piano transcription, which is still considered a challenging and open problem in the literature [74]. The hand-crafted features tended to be less representative to determine the presence of legato-pedal onset in the segment with a shorter duration. These issues set barriers to generalise this method to a new piece recorded using different recording conditions or pianos.

In Chapter 7, to facilitate a more accurate estimation on onset times of any pedalling techniques on the sustain pedal, we took advantages of CNN. This aims to model the temporal and spectral contexts by setting up the CNN's first layer with different filter shapes. To train the CNN with a focus on the nuances produced by pedal onset instead of larger variations introduced by other musical elements, the inputs were time-frequency representations of music excerpts in pairs. The only difference in such paired excerpts was the presence or absence of pedal onset. Likewise, another CNN was trained to distinguish an excerpt played with the sustain pedal that was kept pressed from the associated *no-pedal* version of the excerpt. These two CNN models (`convnet-onset` and `convnet-segment`) were trained separately due to the different acoustic characteristics at the start (pedal onset) versus during the pedalled segment. We experimented with different input representations, hyper-parameters and structures of CNN models based on the knowledge of piano acoustics and physics. The two CNNs with the highest AUC score in the corresponding binary classification task were selected for the *pedal* and *no-pedal* detection on a test piece. Sliding windows were applied to the test piece in order to get decision outputs from the two trained models separately at every frame. The frame-wise decision outputs were fused to locate segments played with the sustain pedal with better performance on boundary detection.

The dataset for training and testing the above CNN-based methods consists of audio data generated from MIDI data using Pianoteq. Such dedicated dataset reduced acoustic complexity which could lead to generalisation issues on commercial recordings. To apply the sustain-pedal detection on real acoustic recordings, the pre-trained CNN model can be employed as a feature extractor to obtain a better representation of the sustain-pedal effect in the recordings. In this way, knowledge learned from the synthesised recordings were transferred. This approach was evaluated on ten passages of Chopin's music, which were recorded using an acoustic grand piano under normal playing condition and synchronously annotated with sustain-pedal movement measured by our direct acquisition system. Better performance was obtained compared to fine-tuning or directly applying the pre-trained CNN models. This shows that the investigated transfer learning method can adapt the detection to acoustic piano recordings, which are sometimes not able to provide enough data to train deep learning models.

From the evaluation experiments, the proposed deep learning methods are effective in learning hierarchical features that represent acoustic characteristics corresponding to pedalling techniques on the sustain pedal. Better performance was obtained for the detection on music pieces around the Romantic era when modern pedalling techniques appeared to have been established and widely used. This gives an affirmative answer to our research question: "Can a computer point out when the sustain pedal is pressed or released if a piano recording from a virtuoso performance is given?" Moreover, to better understand the "black box" of deep neural networks, we conducted visual analysis on the convolutional layers using deconvolution. It was observed from the deconvolved Mel-spectrogram that the first few layers had similar focuses and the last layer narrowed down the differences between *pedal* versus *no-pedal* excerpts to lower frequency bands. More efficient training can be achieved by decreasing the number of convolutional layers. This was reflected in the comparable performance of CNN models with fewer layers but more channels.

To sum up, we recall the research questions presented in Chapter 1 and briefly give the corresponding answers as follows:

1. *What are the pros and cons of the existing methods for measuring instrumental gestures and detecting the corresponding playing techniques in piano performances?*

   Pros and cons of the existing methods for measuring instrumental gestures and detecting the corresponding playing techniques are surveyed in Chapter 2. The pros are that different representations of music content are used. The detection task can be approached from three perspectives: direct acquisition by measurement devices, sensors and embedded systems; indirect acquisition from the audio domain using signal processing, machine learning and deep learning methods; and multimodal modelling strategy. The cons are that research on piano pedalling is underdeveloped despite its importance in expressive piano performances. This thesis is the first to achieve a bottom-up study for piano pedalling detection using indirect, direct and multimodal methods.

2. *How to design a non-intrusive measurement system that could accurately record how the piano sound is modulated by pedalling?*

   A non-intrusive measurement system was designed in Chapter 3. How the piano sound is modulated by pedalling can be recorded directly. The system can also categorise the pedalling gestures into pedalling techniques. This enabled the system to provide a dataset with ground truth automatically annotated, which constituted one of the datasets detailed in Chapter 4.

3. *What features can represent different pedalling techniques in order to facilitate audio-based detection?*

   By exploring the effects of different pedalling techniques on isolated piano tones in Chapter 5, decay patterns of the first partial and the residuals were separately modelled to characterise the effects. The transient that appears in the residual components inspired us to design features representing the sympathetic resonance, which was used to detect the presence of legato-pedal onset in Chapter 6.

4. *Can automatic detection of piano pedalling be improved by considering acoustics and physics of the piano?*

   Yes, the acoustics and physics of the piano were used not only in the design of sympathetic resonance measure, but also the configurations for deep learning models in Chapter 7. The latter one efficiently facilitated the training process in order to effectively detect the audio frames played with or without the sustain pedal.

5. *How to incorporate all the knowledge to generalise the pedalling technique detection so it performs well on any pianos?*

   Given a large dataset, deep learning methods were investigated for incorporating acoustical characteristics when the sustain pedal is used. Since the deep learning models were trained on synthesised data, a transfer learning strategy was proposed to adapt the models into real-world scenarios. This was examined by the dataset consisting of acoustic piano recordings annotated by our non-intrusive measurement system.

8.2 Future Perspectives

8.2.1 Methods for Other Pedalling Techniques

Methods presented in this thesis aimed to detect instrumental gestures and techniques of the sustain pedal, which is more essential and commonly used than the other two pedals, i.e., the sostenuto pedal and the una corda pedal. The sustain pedal is occasionally used in combination with the other two pedals. This is embraced by pianists and composers to deliver new sound effects. With the help of our dedicated measurement system, movements on either pedal are easy to capture directly. In terms of indirect acquisition from audio signals, a future direction is to develop new audio features to better characterise the acoustical properties when the other pedals are used. Given enough training data, audio-based detection can be also approached using deep learning methods.

For the study of the sustain pedal itself, there are pedalling techniques that need to be further investigated for indirect acquisition. This is especially the case for part-pedalling techniques. Regardless of the fact that continuous pedalling gesture data can be obtained, the collection of "ground truth" labels that categorise the gesture into part-pedalling techniques remains a particularly challenging problem. This is due to the nature of part-pedalling techniques as introduced in Section 1.3. There are no absolute pedal positions to define levels of part-pedalling techniques. Pianists have different understandings of the levels, which are dependent on pianos, the acoustics of performance venues and so on. In future work, one can start with one pianist using a specific piano in a studio scenario. If a full spectrum of piano pedalling techniques can be detected, it is possible to achieve a full transcription of piano music with the help of the state-of-the-art note event detection.

8.2.2 Adaptation to Other Pianos

Our datasets were developed using a limited number of pianos with similar recording configurations. Data from one specific grand piano was used to train the proposed models, which may fail to be implemented on the data from another piano. For instance, upright piano obtains different acoustics and physics from the grand one.

A transfer learning strategy was investigated in Section 7.4.2, which was suited to adapt the pre-trained models in detecting the sustain pedal from polyphonic music recorded in another different acoustic and recording conditions. We aim to generalise and examine the detection methods across a wider range of pianos and recording configurations in the future.

8.2.3 Modelling Other Instrumental Gestures and Techniques

Pedalling gestures and techniques were analysed as a case study in this thesis. The acoustics and physics of pianos performed a key role in developing both the direct and indirect acquisition systems. Meanwhile, the direct measurement system can facilitate indirect detection methods by providing automatic-annotated datasets. This motivates us to apply our approach to modelling the instrumental gestures and techniques of other instruments as a future direction. We expect it is feasible on instruments with similar acoustical features, such as struck-string or plucked-string instruments.

Followed by our bottom-up approach, the bottom-level sensor signals representing instrumental gestures can track how they modulate the audio signals. Once a model is trained for a specific instrument using feature engineering or deep learning methods, the high-level playing techniques can be estimated from audio signals without the need for the sensors any more. Successful automatic detection of playing techniques across different instruments can not only reveal the secrets of expressive performance, but also benefit music pedagogy and musicology studies. Development of practical applications will constitute our future works as well.

Appendix A

Datasets Details

A.1 Example of the Notated Score

In Chapter 3, we notated the music scores of ten well-known passages of Chopin's piano music with pedalling techniques. For instance, annotations for the passage from Op. 20 No. 4 are shown in Figure B.1. Different colours of annotations represent pedalling techniques varied in sustain-pedal depth, which is also indicated by the distance between the annotation and stave. Pedalling annotations for the music scores of all the ten passages and their corresponding audio and gesture data can be downloaded from: `http://doi.org/10.5281/zenodo.3237929`.

A.2 MIDI Specifications for Disklavier Rendering

In Section 4.3, specifications with different conditions of pedalling techniques and piano touch were encoded in MIDI files, which were used to render audio using a Yamaha Disklavier grand piano. MIDI specifications for rendering isolated notes played with pedalling techniques on the sustain pedal include:

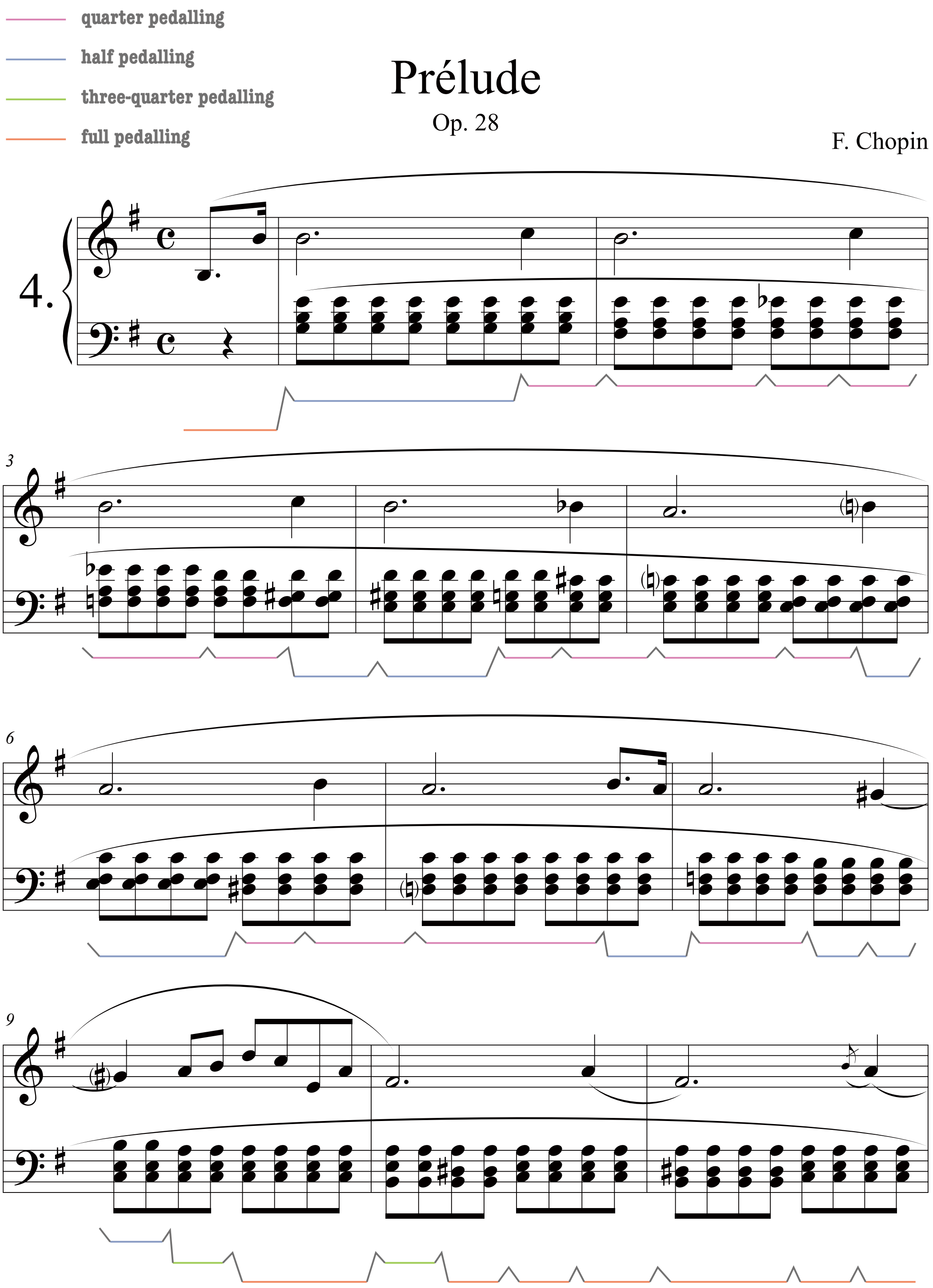


Figure B.1: Pedalling annotations for passage of Chopin's Op. 20 No. 4.

- pedal timing: anticipatory, rhythmic, legato;
- pedal depth: 127 (full pedal), 96 (three-quarter pedal), 64 (half pedal);

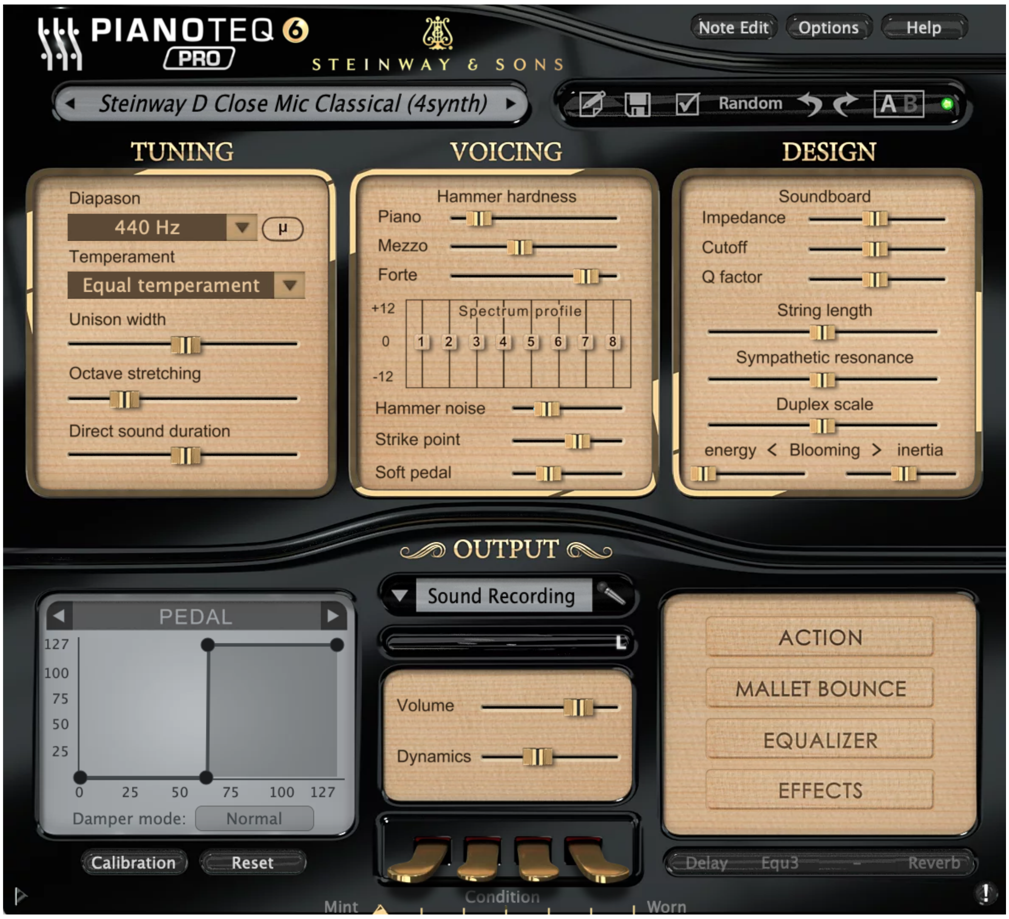
- note velocity: 96 (forte), 84 (mezzo-forte), 49 (piano);
- pitch: MIDI note value in [28, 32, 36, 40, 44, 48, 52, 56, 60, 64, 68, 72, 76, 80, 84, 88, 92, 96, 100, 104].

The above specifications on note velocity and pitch were used to render normally played isolated notes. To estimate partial frequencies and train NMF template in Chapter 6, we generated all the 88 notes played at the mezzo-forte velocity without any pedalling technique.

Apart from isolated notes, repeated notes (same note played repeatedly with an accelerated speed) and trills (rapid alternation between two adjacent notes) at three note velocities played without or with three depths of the anticipatory pedal were generated. Similar specifications were used to render chords and arpeggios (a group of notes from a chord played one after the other in an ascending/descending order). However, specifications on the pitch were different:

- for chords: major, minor, diminished and augmented triad chord with the MIDI value of root note in [36, 48, 60, 72, 84, 96];
- for arpeggios: notes in C major chord going up two octaves, including four arpeggios that correspond to MIDI note value in [[36, 40, 43, 48, 52, 55, 60], [48, 52, 55, 60, 64, 67, 72], [60, 64, 67, 72, 76, 79, 84], [72, 76, 79, 84, 88, 91, 96]].

The same Disklavier piano was also used to playback MIDI files of four well-known pieces. The generated audio data were used in Chapter 6. All the MIDI files and their associated audio recordings presented in this section can be downloaded from: `http://doi.org/10.5281/zenodo.3242149`.



Steinway D Close Mic Classical (4synth)

Diapason
440 Hz

Temperament
Equal temperament

Sound Recording

Damper mode: Normal

Figure B.2: Pianoteq settings for generating *pedal*-version audio.

A.3 Guide to Generating Large Dataset by Pianoteq

For the reproducibility of the study on deep learning in Chapter 7, here we provide a guide to generating large dataset using the MIDI-file playback function in Pianoteq. Given a MIDI file as input, Pianoteq can export the corresponding *pedal* and *no-pedal* versions of the audio data, which forms the dataset presented in Section 4.4.

Pianoteq is a commercial software and offers physically modelled virtual instruments approved by Steinway & Sons. Its GUI allows to playback a single MIDI file with selected piano model and recording setup. For our dataset, we used the "Steinway D Close Mic Classical" preset provided in Pianoteq. To generate *pedal*-version audio, we calibrated "Pedal" in the output setting as shown in Figure B.2 such that the sustain pedal is pressed when its MIDI value exceeds 63. For the *no-pedal* version, we simply let Pianoteq ignore the pedal messages in the MIDI file. To automatically generate both versions based on all the MIDI files within a directory, a bash script is provided to drive Pianoteq in a macOS system as follows:

```bash
#!/bin/bash
for dir in "$@"; do
  for file in "$dir"/*.[Mm][Ii][Dd]; do
    wavfile="${file%.mid}"
    wavfile="${wavfile%.MID}"
    wavfilep="${wavfile}-p.wav"
    wavfilenp="${wavfile}-np.wav"
    "/Applications/Pianoteq 6/Pianoteq 6.app/Contents/MacOS/Pianoteq 6" --
        headless --preset "/Presets/Steinway D Close Mic Classical (4
        synth)" --midi "$file" --wav "$wavfilep"
    "/Applications/Pianoteq 6/Pianoteq 6.app/Contents/MacOS/Pianoteq 6" --
        headless --preset "/Presets/Steinway D Close Mic Classical (4
        synth-np)" --midi "$file" --wav "$wavfilenp"
  done
done
```

where `Steinway D Close Mic Classical (4synth)` and `Steinway D Close Mic Classical (4synth-np)` are the calibrated presets for generating *pedal* and *no-pedal* versions of audio, respectively. Presets were saved as `fxp` files. To run the above script, we can use command line `bash script.sh DIRECTORY`, where `DIRECTORY` should be the path of a folder which includes MIDI files for rendering.

Bibliography

[1] R. I. Godøy and M. Leman, *Musical Gestures: Sound, Movement, and Meaning.* Routledge, 2010.

[2] H.-M. Lehtonen, H. Penttinen, J. Rauhala, and V. Välimäki, “Analysis and Modeling of Piano Sustain-Pedal Effects,” *The Journal of the Acoustical Society of America*, vol. 122, no. 3, pp. 1787–1797, 2007.

[3] H.-M. Lehtonen, A. Askenfelt, and V. Välimäki, “Analysis of the Part-Pedaling Effect in the Piano,” *The Journal of the Acoustical Society of America*, vol. 126, no. 2, pp. EL49–EL54, 2009.

[4] W. Goebl, S. Dixon, G. De Poli, A. Friberg, R. Bresin, and G. Widmer, “Sense in Expressive Music Performance: Data Acquisition, Computational Studies, and Models,” *Sound to Sense - Sense to Sound: A State of the Art in Sound and Music Computing*, pp. 195–242, 2008.

[5] C. Drake, “Into the Fundamentals of Musical Gesture,” in *A. Berthoz, ed. Le Cerveau et le mouvement. Science et Vie, numéro spécial*, 1998, pp. 114–121.

[6] A. R. Jensenius and M. M. Wanderley, “Musical Gestures: Concepts and Methods in Research,” in *Musical Gestures.* Routledge, 2010, pp. 24–47.

[7] F. Delalande, “La gestique de Gould: éléments pour une sémiologie du geste musical,” *Glenn Gould Pluriel*, pp. 85–111, 1988.

[8] S. Gibet, "Codage, représentation et traitement du geste instrumental. Application à la synthèse de sons musicaux par simulation de mécanismes instrumentaux," Ph.D. dissertation, Institut national polytechnique de Grenoble, 1987.

[9] C. Cadoz, "Instrumental Gesture and Musical Composition," in *Proceedings of the International Computer Music Conference (ICMC)*, 1988, pp. 1–12.

[10] M. M. Wanderley and P. Depalle, "Gestural Control of Sound Synthesis," *Proceedings of the IEEE*, vol. 92, no. 4, pp. 632–644, 2004.

[11] C. Cadoz and M. M. Wanderley, "Gesture-Music," 2000.

[12] M. M. Wanderley, "Non-obvious Performer Gestures in Instrumental Music," in *Proceedings of the International Gesture Workshop.* Springer, 1999, pp. 37–48.

[13] H. Fletcher, E. D. Blackham, and R. Stratton, "Quality of Piano Tones," *The Journal of the Acoustical Society of America*, vol. 34, no. 6, pp. 749–761, 1962.

[14] E. D. Blackham, "The Physics of the Piano," *Scientific American*, vol. 213, no. 6, pp. 88–99, 1965.

[15] G. Weinreich, "Coupled Piano Strings," *The Journal of the Acoustical Society of America*, vol. 62, no. 6, pp. 1474–1484, 1977.

[16] A. M. Keil, "The Dawn of Modern Piano Pedaling: Early Twentieth-Century Piano Pedaling Literature and Techniques," Ph.D. dissertation, Bowling Green State University, 2015.

[17] D. L. Root, *Grove Music Online.* Oxford University Press, 2007.

[18] N. H. Fletcher and T. Rossing, *The Physics of Musical Instruments.* Springer Science & Business Media, 2012.

[19] C. Morfey, *Dictionary of Acoustics.* Academic Press, 2001.

[20] D. Rowland, *A History of Pianoforte Pedalling.* Cambridge University Press, 2004.

[21] S. P. Rosenblum, “Pedaling the Piano: A Brief Survey from the Eighteenth Century to Present,” *Performance Practice Review*, vol. 6, no. 2, p. 8, 1993.

[22] K. U. Schnabel, *Modern Technique of the Pedal: A Piano Pedal Study.* Mills Music, 1954.

[23] E. Chew and A. R. J. François, “MuSA. RT and the Pedal: the Role of the Sustain Pedal in Clarifying Tonal Structure,” in *Proceedings of the 10th International Conference on Music Perception and Cognition*, 2008.

[24] T. Machover, *Hyperinstruments: A Progress Report, 1987-1991.* MIT Media Laboratory, 1992.

[25] A. Hunt and M. M. Wanderley, “Mapping Performer Parameters to Synthesis Engines,” *Organised Sound*, vol. 7, no. 2, pp. 97–108, 2002.

[26] A. Kapur, P. Davidson, P. R. Cook, P. F. Driessen, and W. A. Schloss, “Digitizing North Indian Performance.” in *Proceedings of the International Computer Music Conference (ICMC).* Citeseer, 2004.

[27] D. Overholt, “The Overtone Violin,” in *Proceedings of the 5th International Conference on New Interfaces for Musical Expression (NIME).* National University of Singapore, 2005, pp. 34–37.

[28] A. McPherson, “The Magnetic Resonator Piano: Electronic Augmentation of an Acoustic Grand Piano,” *Journal of New Music Research*, vol. 39, no. 3, pp. 189–202, 2010.

[29] R. Timmers, H. Honing, and Others, “On Music Performance, Theories, Measurement and Diversity,” *Cognitive Processing (International Quarterly of Cognitive Sciences)*, vol. 1, no. 2, pp. 1–19, 2002.

[30] W. Goebl and G. Widmer, “On the Use of Computational Methods for Expressive Music Performance,” *Modern Methods for Musicology: Prospects, Proposals, and Realities*, pp. 93–113, 2009.

[31] W. Goebl, S. Dixon, and E. Schubert, "Quantitative Methods: Motion Analysis, Audio Analysis, and Continuous Response Techniques," *Expressiveness in Music Performance: Empirical Approaches Across Styles and Cultures*, pp. 221–239, 2014.

[32] A. Tindale, A. Kapur, and G. Tzanetakis, "Training Surrogate Sensors in Musical Gesture Acquisition Systems," *IEEE Transactions on Multimedia*, vol. 13, no. 1, pp. 50–59, 2011.

[33] W. Goebl, "Movement and Touch in Piano Performance," *Handbook of Human Motion*, pp. 1–18, 2017.

[34] J. MacRitchie, "The Art and Science behind Piano Touch: A Review Connecting Multi-Disciplinary Literature," *Musicae Scientiae*, vol. 19, no. 2, pp. 171–190, 2015.

[35] W. Goebl and C. Palmer, "Tactile Feedback and Timing Accuracy in Piano Performance," *Experimental Brain Research*, vol. 186, no. 3, pp. 471–479, 2008.

[36] ——, "Finger Motion in Piano Performance: Touch and Tempo," in *Proceedings of the International Symposium on Performance Science.* European Association of Conservatoires (AEC) Utrecht, The Netherlands, 2009, pp. 65–70.

[37] S. Furuya, T. Goda, H. Katayose, H. Miwa, and N. Nagata, "Distinct Inter-joint Coordination During Fast Alternate Keystrokes in Pianists with Superior Skill," *Frontiers in Human Neuroscience*, vol. 5, p. 50, 2011.

[38] S. Dalla Bella and C. Palmer, "Rate Effects on Timing, Key Velocity, and Finger Kinematics in Piano Performance," *PloS one*, vol. 6, no. 6, p. e20518, 2011.

[39] W. Goebl and C. Palmer, "Temporal Control and Hand Movement Efficiency in Skilled Music Performance," *PloS one*, vol. 8, no. 1, p. e50901, 2013.

[40] V. F. Ferrario, C. Macrì, E. Biffi, P. Pollice, and C. Sforza, "Three-dimensional Analysis of Hand and Finger Movements During Piano Playing," *Medical Problems of Performing Artists*, vol. 22, no. 1, pp. 18–24, 2007.

[41] N. Sakai, M. C. Liu, F.-C. Su, A. T. Bishop, and K.-N. An, "Hand Span and Digital Motion on the Keyboard: Concerns of Overuse Syndrome in Musicians," *The Journal of Hand Surgery*, vol. 31, no. 5, pp. 830–835, 2006.

[42] A. Hadjakos, "Pianist Motion Capture with the Kinect Depth Camera," in *Proceedings of the Sound and Music Computing Conference.* Citeseer, 2012, pp. 303–310.

[43] M. Li, P. Savvidou, B. Willis, and M. Skubic, "Using the Kinect to Detect Potentially Harmful Hand Postures in Pianists," in *Proceedings of the 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society.* IEEE, 2014, pp. 762–765.

[44] D. Johnson, I. Dufour, D. Damian, and G. Tzanetakis, "Detecting Pianist Hand Posture Mistakes for Virtual Piano Tutoring," in *Proceedings of the International Computer Music Conference (ICMC)*, 2016, pp. 167–170.

[45] M. Bernays and C. Traube, "Investigating Pianists' Individuality in the Performance of Five Timbral Nuances through Patterns of Articulation, Touch, Dynamics, and Pedaling," *Frontiers in Psychology*, vol. 5, no. 157, 2014.

[46] M. Müller, V. Konz, W. Bogler, and V. Arifi-Müller, "Saarland Music Data (SMD)," in *Late-Breaking and Demo Session of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, Miami, USA, 2011.

[47] V. Emiya, N. Bertin, B. David, and R. Badeau, "MAPS-A Piano Database for Multipitch Estimation and Automatic Transcription of Music," Télécom ParisTech, Tech. Rep., 2010.

[48] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, "Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset," *arXiv preprint arXiv:1810.12247*, 2018.

[49] W. Goebl and R. Bresin, “Measurement and Reproduction Accuracy of Computer-controlled Grand Pianos,” *The Journal of the Acoustical Society of America*, vol. 114, no. 4, pp. 2273–2283, 2003.

[50] C. Medeiros and M. Wanderley, “A Comprehensive Review of Sensors and Instrumentation Methods in Devices for Musical Expression,” *Sensors*, vol. 14, no. 8, pp. 13 556–13 591, 2014.

[51] A. Binet and J. Courtier, “Recherches Graphiques Sur La Musique,” *L’Année Psychologique*, vol. 2, no. 1, pp. 201–222, 1895.

[52] O. Ortmann, *The Physical Basis of Piano Touch and Tone.* Kegan Paul, Trenc, Trubner & Co., London, 1925.

[53] ——, *The Physiological Mechanics of Piano Technique.* Kegan Paul, Trench, Trubner; J. Curwen; EP Dutton, London, 1929.

[54] A. Hadjakos, E. Aitenbichler, and M. Mühlhäuser, “Syssomo: A Pedagogical Tool for Analyzing Movement Variants Between Different Pianists,” in *Proceedings of the 5th International Conference on Enactive Interfaces*, 2008.

[55] T. Großhauser, B. Tessendorf, G. Tröster, H. Hildebrandt, and V. Candia, “Sensor Setup for Force and Finger Position and Tilt Measurements for Pianists,” in *Proceedings of the Sound and Music Computing Conference.* Aalborg University, 2012, pp. 264–270.

[56] A. McPherson, “TouchKeys: Capacitive Multi-Touch Sensing on a Physical Keyboard,” in *Proceedings of the 12th International Conference on New Interfaces for Musical Expression (NIME)*, 2012.

[57] ——, “Portable Measurement and Mapping of Continuous Piano Gesture,” in *Proceedings of the 13th International Conference on New Interfaces for Musical Expression (NIME)*, 2013, pp. 152–157.

[58] J. MacRitchie and N. J. Bailey, “Efficient Tracking of Pianists’ Finger Movements,” *Journal of New Music Research*, vol. 42, no. 1, pp. 79–95, 2013.

[59] J. MacRitchie and A. P. McPherson, “Integrating Optical Finger Motion Tracking with Surface Touch Events,” *Frontiers in psychology*, vol. 6, p. 702, 2015.

[60] C. Zhang, L. Shen, D. Wang, F. Tian, and H. Wang, “CoolMag: A Tangible Interaction Tool to Customize Instruments for Children in Music Education,” in *Proceedings of the 13th International Conference on Ubiquitous Computing.* ACM, 2011, pp. 581–582.

[61] J. Harriman, “Start’em Young: Digital Music Instrument for Education,” in *Proceedings of the 15th International Conference on New Interfaces for Musical Expression (NIME)*, 2015, pp. 70–73.

[62] H. Leeuw, “The Electrumpet, a Hybrid Electro-Acoustic Instrument.” in *Proceedings of the 9th International Conference on New Interfaces for Musical Expression (NIME)*, 2009, pp. 193–198.

[63] D. Schlessinger and J. O. Smith, “The Kalichord: A Physically Modeled Electro-Acoustic Plucked String Instrument.” in *Proceedings of the 9th International Conference on New Interfaces for Musical Expression (NIME).* Citeseer, 2009, pp. 98–101.

[64] E. Berdahl, S. Salazar, and M. Borins, “Embedded Networking and Hardware-Accelerated Graphics with Satellite CCRMA.” in *Proceedings of the 13th International Conference on New Interfaces for Musical Expression (NIME)*, 2013, pp. 325–330.

[65] G. Moro, A. Bin, R. H. Jack, C. Heinrichs, A. P. McPherson, and Others, “Making High-performance Embedded Instruments with Bela and Pure Data,” in *Proceedings of the International Conference on Live Interfaces.* University of Sussex, 2016.

[66] A. McPherson, "Bela: An Embedded Platform for Low-latency Feedback Control of Sound," *The Journal of the Acoustical Society of America*, vol. 141, no. 5, p. 3618, 2017.

[67] A. Luciani, M. Evrard, D. Couroussé, N. Castagné, C. Cadoz, and J.-L. Florens, "A Basic Gesture and Motion Format for Virtual Reality Multisensory Applications," *arXiv preprint arXiv:1005.4564*, 2010.

[68] A. R. Jensenius, T. Kvifte, and R. I. Godøy, "Towards a Gesture Description Interchange Format," in *Proceedings of the 6th International Conference on New Interfaces for Musical Expression (NIME)*. IRCAMCentre Pompidou, 2006, pp. 176–179.

[69] M. Wright, A. Freed, X. Rodet, D. Virolle, and R. Woehrmann, "New Applications of the Sound Description Interchange Format," in *Proceedings of the International Computer Music Conference (ICMC)*, 1998.

[70] A. Jensenius, N. Castagné, A. Camurri, E. Maestre, J. Malloch, and D. Mc Gilvray, "A Summary of Formats for Streaming and Storing Music-related Movement and Gesture Data," in *Proceedings of the 4th International Conference on Enactive Interfaces*, 2007, pp. 125–128.

[71] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler, "A Tutorial on Onset Detection in Music Signals," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 1035–1047, 2005.

[72] A. De Cheveigné and H. Kawahara, "YIN, a Fundamental Frequency Estimator for Speech and Music," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[73] J. Salamon and E. Gómez, "Melody Extraction from Polyphonic Music Signals Using Pitch Contour Characteristics," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 6, pp. 1759–1770, 2012.

[74] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic Music Transcription: An Overview," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2019.

[75] J. Nam, K. Choi, J. Lee, S.-Y. Chou, and Y.-H. Yang, "Deep Learning for Audio-Based Music Classification and Tagging: Teaching Computers to Distinguish Rock from Bach," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 41–51, 2019.

[76] V. Lostanlen, J. Andén, and M. Lagrange, "Extended Playing Techniques: The next Milestone in Musical Instrument Recognition," in *Proceedings of the 5th International Workshop on Digital Libraries for Musicology (DLfM)*, 2018.

[77] C.-W. Wu, C. Dittmar, C. Southall, R. Vogl, G. Widmer, J. Hockman, M. Muller, and A. Lerch, "A Review of Automatic Drum Transcription," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 26, no. 9, pp. 1457–1483, 2018.

[78] L. Yang, K. Z. Rajab, and E. Chew, "The Filter Diagonalisation Method for Music Signal Analysis: Frame-wise Vibrato Detection and Estimation," *Journal of Mathematics and Music*, pp. 1–19, 2017.

[79] L. Su, L.-F. Yu, and Y.-H. Yang, "Sparse Cepstral, Phase Codes for Guitar Playing Technique Classification." in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, 2014, pp. 9–14.

[80] Y.-P. Chen, L. Su, and Y.-H. Yang, "Electric Guitar Playing Technique Detection in Real-World Recording Based on F0 Sequence Pattern Recognition." in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015, pp. 708–714.

[81] P.-C. Li, L. Su, Y.-h. Yang, A. W. Y. Su, and Others, "Analysis of Expressive Musical Terms in Violin Using Score-Informed and Expression-Based Audio Features." in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015, pp. 809–815.

[82] A. Perez-Carrillo and M. M. Wanderley, “Indirect Acquisition of Violin Instrumental Controls from Audio Signal with Hidden Markov Models,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 5, pp. 932–940, 2015.

[83] D. Gabor, “Theory of Communication. Part 1: The Analysis of Information,” *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, vol. 93, no. 26, pp. 429–441, 1946.

[84] R. A. Rasch, “Synchronization in Performed Ensemble Music,” *Acta Acustica united with Acustica*, vol. 43, no. 2, pp. 121–131, 1979.

[85] K. Choi, G. Fazekas, and M. B. Sandler, “Automatic Tagging Using Deep Convolutional Neural Networks,” in *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, 2016, pp. 805–811.

[86] K. Choi, G. Fazekas, M. Sandler, and K. Cho, “Convolutional Recurrent Neural Networks for Music Classification,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 2392–2396.

[87] J. C. Brown, “Calculation of a Constant Q Spectral Transform,” *The Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, 1991.

[88] R. D. Patterson, I. Nimmo-Smith, J. Holdsworth, and P. Rice, “An Efficient Auditory Filterbank Based on the Gammatone Function,” in *IOC Speech Group on Auditory Modelling at RSRE*, vol. 2, no. 7, 1987.

[89] X. Serra and Others, “Musical Sound Modeling with Sinusoids plus Noise,” *Musical Signal Processing*, pp. 91–122, 1997.

[90] D. D. Lee and H. S. Seung, “Learning the Parts of Objects by Non-negative Matrix Factorization,” *Nature*, vol. 401, no. 6755, p. 788, 1999.

[91] P. Smaragdis and J. C. Brown, “Non-negative Matrix Factorization for Polyphonic

Music Transcription," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics.* IEEE, 2003, pp. 177–180.

[92] E. Vincent, N. Bertin, and R. Badeau, "Adaptive Harmonic Spectral Decomposition for Multiple Pitch Estimation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 528–537, 2010.

[93] E. Benetos and S. Dixon, "Multiple-instrument Polyphonic Music Transcription Using a Temporally Constrained Shift-invariant Model," *The Journal of the Acoustical Society of America*, vol. 133, no. 3, pp. 1727–1741, 2013.

[94] B. Fuentes, R. Badeau, and G. Richard, "Harmonic Adaptive Latent Component Analysis of Audio and Application to Music Transcription," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 9, pp. 1854–1866, 2013.

[95] S. Ewert and M. Sandler, "Piano Transcription in the Studio Using an Extensible Alternating Directions Framework," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 1983–1997, 2016.

[96] T. Cheng, M. Mauch, E. Benetos, and S. Dixon, "An Attack/Decay Model for Piano Transcription," in *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, 2016.

[97] C.-W. Wu and A. Lerch, "On Drum Playing Technique Detection in Polyphonic Mixtures," in *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, 2016.

[98] G. Peeters, "A Large Set of Audio Features for Sound Description (Similarity and Classification) in the CUIDADO Project," *Rapport Technique, CUIDADO I.S.T.*, 2004.

[99] E. Fix and J. L. Hodges Jr, "Discriminatory Analysis-nonparametric Discrimination: Consistency Properties," University of California, Berkeley, Tech. Rep., 1951.

[100] D. D. Lewis, "Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval," in *Proceedings of the European Conference on Machine Learning.* Springer, 1998, pp. 4–15.

[101] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised Machine Learning: A Review of Classification Techniques," *Emerging Artificial Intelligence Applications in Computer Engineering*, vol. 160, pp. 3–24, 2007.

[102] S. Ruder, "An Overview of Gradient Descent Optimization Algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[103] C. Cortes and V. Vapnik, "Support-vector Networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[104] L. Breiman, *Classification and Regression Trees.* Routledge, 2017.

[105] L. E. Baum and T. Petrie, "Statistical Inference for Probabilistic Functions of Finite State Markov Chains," *The Annals of Mathematical Statistics*, vol. 37, no. 6, pp. 1554–1563, 1966.

[106] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[107] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[108] L. R. Rabiner, "A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[109] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[110] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[111] Y. Bengio, P. Simard, P. Frasconi, and Others, “Learning Long-term Dependencies with Gradient Descent Is Difficult,” *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[112] V. Nair and G. E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines,” in *Proceedings of the 27th international conference on machine learning (ICML)*, 2010, pp. 807–814.

[113] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning.* MIT Press Cambridge, 2016, vol. 1.

[114] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation Applied to Handwritten Zip Code Recognition,” *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[115] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

[116] Y. LeCun and Others, “Generalization and Network Design Strategies,” in *Connectionism in Perspective.* Citeseer, 1989, vol. 19.

[117] Y.-T. Zhou and R. Chellappa, “Computation of Optical Flow Using a Neural Network,” in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 1998, 1988, pp. 71–78.

[118] L. Torrey and J. Shavlik, “Transfer Learning,” in *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques.* IGI Global, 2010, pp. 242–264.

[119] S. J. Pan and Q. Yang, “A Survey on Transfer Learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[120] W. Rawat and Z. Wang, “Deep Convolutional Neural Networks for Image Clas-

sification: A Comprehensive Review," *Neural Computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

[121] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Transfer Learning for Music Classification and Regression Tasks," in *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017, pp. 141–149.

[122] C. Liem, M. Müller, D. Eck, G. Tzanetakis, and A. Hanjalic, "The Need for Music Information Retrieval with User-centered and Multimodal Strategies," in *Proceedings of the 1st International Acm Workshop on Music Information Retrieval with User-centered and Multimodal Strategies.* ACM, 2011, pp. 1–6.

[123] S. Essid and G. Richard, "Fusion of Multimodal Information in Music Content Analysis," in *Dagstuhl Follow-Ups*, vol. 3. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2012.

[124] Z. Duan, S. Essid, C. C. S. Liem, G. Richard, and G. Sharma, "Audiovisual Analysis of Music Performances: Overview of an Emerging Field," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 63–73, 2019.

[125] K. Dinesh, B. Li, X. Liu, Z. Duan, and G. Sharma, "Visually Informed Multipitch Analysis of String Ensembles," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2017, pp. 3021–3025.

[126] B. Li, K. Dinesh, G. Sharma, and Z. Duan, "Video-Based Vibrato Detection and Analysis for Polyphonic String Music." in *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017, pp. 123–130.

[127] S. Parekh, S. Essid, A. Ozerov, N. Q. K. Duong, P. Pérez, and G. Richard, "Guiding Audio Source Separation by Video Object Information," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA).* IEEE, 2017, pp. 61–65.

[128] M. Mueller, A. Arzt, S. Balke, M. Dorfer, and G. Widmer, "Cross-modal Music Retrieval and Applications: An Overview of Key Methodologies," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 52–62, 2019.

[129] E. Schoonderwaldt and M. Demoucron, "Extraction of Bowing Parameters from Violin Performance Combining Motion Capture and Sensors," *The Journal of the Acoustical Society of America*, vol. 126, no. 5, pp. 2695–2708, 2009.

[130] L. S. Pardue, C. Harte, and A. P. McPherson, "A Low-Cost Real-Time Tracking System for Violin," *Journal of New Music Research*, vol. 44, no. 4, pp. 305–323, 2015.

[131] E. Maestre, J. Bonada, M. Blaauw, A. Perez, and E. Guaus, "Acquisition of Violin Instrumental Gestures Using a Commercial EMF Tracking Device," in *Proceedings of the International Computer Music Conference (ICMC)*, vol. 1, 2007, pp. 386–393.

[132] A. Perez, J. Bonada, E. Maestre, E. Guaus, and M. Blaauw, "Combining Performance Actions with Spectral Models for Violin Sound Transformation," in *Proceedings of 19th International Congress on Acoustics*, 2007.

[133] M. Levy and M. Sandler, "Structural Segmentation of Musical Audio by Constrained Clustering," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 318–326, 2008.

[134] J. A. Hanley and B. J. McNeil, "A Method of Comparing the Areas under Receiver Operating Characteristic Curves Derived from the Same Cases," *Radiology*, vol. 148, no. 3, pp. 839–843, 1983.

[135] B. Liang, G. Fazekas, A. McPherson, and M. Sandler, "Piano Pedaller: A Measurement System for Classification and Visualisation of Piano Pedalling Techniques," in *Proceedings of the 17th International Conference on New Interfaces for Musical Expression (NIME)*, 2017, pp. 325–329.

[136] B. Liang, G. Fazekas, and M. Sandler, “Recognition of piano pedalling techniques using gesture data,” in *Proceedings of the 12th International Audio Mostly Conference on Augmented and Participatory Sound and Music Experiences.* ACM, 2017.

[137] ——, “Measurement, Recognition, and Visualization of Piano Pedaling Gestures and Techniques,” *Journal of the Audio Engineering Society*, vol. 66, no. 6, pp. 448–456, jun 2018.

[138] A. McPherson and V. Zappi, “An Environment for Submillisecond-Latency Audio and Sensor Processing on BeagleBone Black,” in *Proceedings of the 138th Audio Engineering Society Convention.* Audio Engineering Society, 2015.

[139] 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*, vol. 12, pp. 2825–2830, 2011.

[140] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, and P. B. Kramer, *Numerical Recipes: The Art of Scientific Computing.* AIP, 1987.

[141] K. Pandia, S. Ravindran, R. Cole, G. Kovacs, and L. Giovangrandi, “Motion Artifact Cancellation to Obtain Heart Sounds from a Single Chest-Worn Accelerometer,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2010, pp. 590–593.

[142] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification.* Wiley-Interscience, 2000.

[143] Y. Altun, I. Tsochantaridis, and T. Hofmann, “Hidden Markov Support Vector Machines,” in *Proceedings of the 20th International Conference on Machine Learning (ICML)*, vol. 3, 2003, pp. 3–10.

[144] S. Wang, S. Ewert, and S. Dixon, "Compensating for Asynchronies between Musical Voices in Score-Performance Alignment," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2015, pp. 589–593.

[145] D. J. Berndt and J. Clifford, "Using Dynamic Time Warping to Find Patterns in Time Series," in *Proceedings of the KDD workshop*, vol. 10, no. 16. Seattle, WA, 1994, pp. 359–370.

[146] V. Emiya, R. Badeau, and B. David, "Multipitch Estimation of Piano Sounds Using a New Probabilistic Spectral Smoothness Principle," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2010.

[147] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic Music Transcription: Challenges and Future Directions," *Journal of Intelligent Information Systems*, vol. 41, no. 3, pp. 407–434, 2013.

[148] R. Badeau, N. Bertin, B. David, A. Schutz, and D. Slock, "Piano "Forte Pedal" Analysis and Detection," in *Proceedings of the 124th Audio Engineering Society Convention.* Audio Engineering Society, 2008.

[149] B. Liang, G. Fazekas, and M. Sandler, "Detection of Piano Pedalling Techniques on the Sustain Pedal," in *Proceedings of the 143rd Audio Engineering Society Convention.* Audio Engineering Society, 2017.

[150] F. Rigaud, B. David, and L. Daudet, "A Parametric Model and Estimation Techniques for the Inharmonicity and Tuning of the Piano," *The Journal of the Acoustical Society of America*, vol. 133, no. 5, pp. 3107–3118, 2013.

[151] A. Galembo and A. Askenfelt, "Signal Representation and Estimation of Spectral Parameters by Inharmonic Comb Filters with Application to the Piano," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 2, pp. 197–203, 1999.

[152] J. Rauhala and V. Välimäki, “F0 Estimation of Inharmonic Piano Tones Using Partial Frequencies Deviation Method,” in *Proceedings of the International Computer Music Conference (ICMC)*, 2007, pp. 453–456.

[153] T. Cheng, S. Dixon, and M. Mauch, “Modelling the Decay of Piano Sounds,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 594–598.

[154] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *Proceedings of the Advances in Neural Information Processing Systems*, 2001, pp. 556–562.

[155] F. Rigaud, B. David, and L. Daudet, “A Parametric Model of Piano Tuning,” in *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx)*, 2011, pp. 393–399.

[156] W. M. Szeto and K. H. Wong, “Sinusoidal Modeling for Piano Tones,” in *Proceedings of the IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC)*. IEEE, 2013, pp. 1–6.

[157] T. Virtanen and A. Klapuri, “Separation of Harmonic Sounds Using Multipitch Analysis and Iterative Parameter Estimation,” in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 2001, pp. 83–86.

[158] ——, “Separation of Harmonic Sounds Using Linear Models for the Overtone Series,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2. IEEE, 2002, pp. 1757–1760.

[159] J. H. Friedman, “Multivariate Adaptive Regression Splines,” *The Annals of Statistics*, pp. 1–67, 1991.

[160] S. Wright, “Correlation and Causation,” *Journal of Agricultural Research*, vol. 20, no. 7, pp. 557–585, 1921.

[161] M. Newville, A. Nelson, T. Stensitzki, A. Ingargiola, D. Allan, Y. Ram, C. Deil, G. Pasquevich, T. Spillane, P. A. Brodtkorb, and Others, “LMFIT: Non-linear Least-square Minimization and Curve-fitting for Python,” *Astrophysics Source Code Library*, 2016.

[162] W.-Y. Loh, “Classification and Regression Trees,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 14–23, 2011.

[163] C. J. C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition,” *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[164] F. Takahashi and S. Abe, “Decision-tree-based Multiclass Support Vector Machines,” in *Proceedings of the 9th International Conference on Neural Information Processing (ICONIP)*, vol. 3. IEEE, 2002, pp. 1418–1422.

[165] C.-C. Chang and C.-J. Lin, “LIBSVM: A Library for Support Vector Machines,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[166] J. Banowetz, *The Pianist’s Guide to Pedaling.* Georgetown University Press, 1992.

[167] B. Liang, G. Fazekas, and M. Sandler, “Piano legato-pedal onset detection based on a sympathetic resonance measure,” in *Proceedings of the 26th European Signal Processing Conference (EUSIPCO).* IEEE, 2018, pp. 2484–2488.

[168] T. Cheng, “Exploiting Piano Acoustics in Automatic Transcription,” Ph.D. dissertation, Queen Mary University of London, 2016.

[169] S. Ewert, M. D. Plumbley, and M. Sandler, “A Dynamic Programming Variant of Non-negative Matrix Deconvolution for the Transcription of Struck String Instruments,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2015, pp. 569–573.

[170] D. Fitzgerald, "Harmonic/Percussive Separation Using Median Filtering," in *Proceedings of the 13th International Conference on Digital Audio Effects (DAFx)*, 2010.

[171] H. Akaike, "A New Look at the Statistical Model Identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.

[172] D. L. Hall and J. Llinas, "An Introduction to Multisensor Data Fusion," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 6–23, 1997.

[173] B. Liang, G. Fazekas, and M. Sandler, "Piano sustain-pedal detection using convolutional neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2019, pp. 241–245.

[174] ——, "Transfer learning for piano sustain-pedal detection," in *Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN).* IEEE, 2019.

[175] J. Pons and X. Serra, "Designing Efficient Architectures for Modeling Temporal Features with Convolutional Neural Networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2017, pp. 2472–2476.

[176] J. Pons, O. Slizovskaia, E. Gómez Gutiérrez, and X. Serra, "Timbre Analysis of Music Audio Signals with Convolutional Neural Networks," in *Proceedings of the 25th European Signal Processing Conference (EUSIPCO)*, 2017.

[177] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-scale Image Recognition," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2014.

[178] K. Choi, D. Joo, and J. Kim, "Kapre: On-GPU Audio Preprocessing Layers for a Quick Implementation of Deep Neural Network Models with Keras," in *Proceedings of the Machine Learning for Music Discovery Workshop at 34th International Conference on Machine Learning.* ICML, 2017.

[179] F. Chollet and Others, "Keras," https://keras.io, 2015.

[180] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2016, pp. 265–283.

[181] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," in *Proceedings of the European Conference on Computer Vision.* Springer, 2014, pp. 818–833.

[182] M. Tian, G. Fazekas, D. A. A. Black, and M. B. Sandler, "Design And Evaluation of Onset Detectors using Different Fusion Policies," in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, 2014, pp. 631–636.

[183] D. Turnbull, G. R. G. Lanckriet, E. Pampalk, and M. Goto, "A Supervised Approach for Detecting Boundaries in Music Using Difference Features and Boosting." in *Proceedings of the 8th International Society for Music Information Retrieval Conference (ISMIR)*, 2007, pp. 51–54.

[184] A. Galembo, "Perception of Musical Instrument by Performer and Listener (with Application to the Piano)," in *Proceedings of the International Workshop on Human Supervision and Control in Engineering and Music*, 2001, pp. 257–266.

[185] S. Bolzinger, O. Warusfel, and E. Kahle, "A Study of the Influence of Room Acoustics on Piano Performance," *Le Journal de Physique IV*, vol. 4, no. C5, pp. 617–620, 1994.