**POLITECNICO DI MILANO**
**Master's Degree in Computer Engineering**
**Department of Electronics and Information Technology**

# AN INVESTIGATION OF PIANO TRANSCRIPTION ALGORITHM FOR JAZZ MUSIC

**Supervisor: Prof. Fabio Antonacci, Politecnico di Milano**
**Co-supervisor: Prof. Peter Knees, Technische Universität Wien**
**Co-supervisor: Richard Vogl, Technische Universität Wien**

**Master Thesis of:**
**Giorgio Marzorati, ID 876546**

**Academic Year 2017-2018**

*“ Never say never.*
*Because limits, like fears are often just an illusion.”*

Michael Jordan

# Abstract

The thesis aims to create an annotated musical dataset and to propose an Automatic Music Transcription system specific to jazz music only. Although many available annotated datasets are built from the audio recordings, the proposed one is built from MIDI file format data, providing robust annotation. The automatic polyphonic transcription method uses a Convolutional Neural Network for the prediction of the outcome.
Automatic Music Transcription is an interesting and active research field of Music Information Retrieval. Automatic Music Transcription refers to the analysis of the musical signal to extract a parametric representation of it, e.g. a musical score or MIDI format file. Even for man, the transcription of music is difficult and still remains a hard task requiring a deep knowledge of music and high level of musical training. Providing a parametric representation of audio signals would be important for application to annotated music for automatic research in large and interactive musical systems. Massive support would be given to the musicology fields producing annotation for audio performance without any written representation, and to the education field. The work hereby presented is focused on the jazz genre, due to its variety of styles and improvisation parts, of which usually there is no available transcription, and to which the field of Automatic Music Transcription can be of help. Its variability makes the problem of Automatic Music Transcription even more challenging and also for that reason there is not much work available.
Results of the transcription system highlighted the difficulties of transcribing jazz music, compared to classical music, but still comparable to state-of-art methodologies, producing an f-measure of 0.837 testing the Neural Network on 30 tracks of MAPS dataset and 0.50 from the jazz dataset experiment.

# Acknowledgment

I would first like to thank my thesis advisor Professor Fabio Antonacci of the Computer Engineering Department at Politecnico di Milano, whose expertise, diligence and patience were crucial in writing this thesis. He consistently allowed this dissertation to be my own work, but steered me in the right direction whenever he thought I needed it.

I would also like to thank the experts who were involved throughout the entire developing phase of this research project: my foreign advisor in Vienna, Professor Peter Knees and assistent Richard Vogl of the Informatic faculty of Technische Universität Wien. Without their passionate participation and input, it could not have been successfully conducted.

I must express my very profound gratitude to my parents and to my family for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis.

Finally, I would like to thank all my true friends for showing me constant love and support. This accomplishment would not have been possible without them. Thank you.

Giorgio Marzorati

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 The challenge of Automatic Music Transcription

Automatic Music Transcription is the process that allows the extraction of a parametric representation of a musical signal through its analysis. Researches has been undertaken in this field of study for 35 years starting with Moorer [1], and cover specific areas of monophonic and polyphonic music. Automatic transcription for monophonic streams is widely considered a problem already solved [2]; as a matter of fact, transcriptions with the highest rates of accuracy are actually obtainable for any musical instrument. Polyphony, on the contrary, has manifold intrinsic complexities to be considered that constrain the research to specific cases of study and analysis for a possible automatic transcription. The difficulties to achieve reasonable results thus indicate that improvements and streamlined approaches are necessary to the Automatic Music Transcription of polyphonic music signals, underlining in it the true challenge of this subject.
Automatic Music Transcription is a complex system divided into subtasks: pitch estimation, related to the detection of a given time frame, onset and offset detection of the relative pitch, loudness estimation, and finally instrument recognition and extraction of rhythmic pattern. The core task of AMT applied to polyphonic music, is the estimation of concurrent pitches, also known as multiple-F0 or multi-pitch estimation. Depending on the number of subtasks, the output of an AMT system can be a simple representation, such as MIDI or piano-roll, or a complex one, such as an orchestral score.

An AMT system has to solve a set of MIR problems to produce a complete music notation, central to which is the multi-pitch estimation.
Other features should be included in the representation in order to improve the transcription performance, such as descriptors of rhythm, melody, harmony and instrumentation. The estimation of most of these features has been studied as isolated tasks, as instrument recognition, detection of onset and offset, extraction of rhythmic information (tempo, beat, musical timing), pitch estimation and harmony (key, chords). Usually, separated algorithms are required for each individual result. Considering the complexity of each task this is necessary, but the main drawback is the combination of outputs from different systems or the creation of an algorithm to perform the joint estimation of all required parameters.
Another challenge is represented by the non-availability of data for evaluation and training. This is not due to a shortage of transcriptions and scores, but to the human effort required to digitize and time-align the music notation to the recording. The only exception is represented by the piano solos thanks to the available data from the MAPS database [3] and the Disklavier piano dataset.

## 1.2 Scope of the thesis

The thesis has a double aim: to create an annotated musical dataset completely dedicated to jazz music and to propose a specific AMT system specific to polyphonic jazz recordings. The AMT method is based on the estimation of multiple-pitch and onset detection using a Neural Network algorithm.
Annotated music plays a central role in Multi-Pitch estimation and Automatic Music Transcription. A great amount of data is required for the development and evaluation of such algorithms which rely on statistical approaches. However, such data is usually embedded in annotated sound databases, some of which are public, while others, as those used in the MIREX context, are private. Few databases are currently available and usually consist of musical instruments and recordings from which the annotated file is derived with the ensuing problem of inaccurate and erroneous values for pitch, onset and offset time.
The choice of working with jazz genre for the AMT system was mainly justified by the lack of specific works, and hence the need of gathering a dedicated dataset. On the other hand, the choice is also motivated by the genre itself,

as jazz comprises a wide variety of different styles and ways of performing the same musical pieces. Jazz is a genre that is renowed for its syncopated rhythms and chord progression, the improvisational phrasing, the intimate expressiveness and the articulated melodies. It is best defined in the words of one of the greatest trumpeters Wynton Marsalis: "*Jazz is not just, "Well, man, this is what I feel like playing ." It is a very structured thing that comes down from a tradition and requires a lot of thought and study.*".

Usually, opening and closing parts of jazz compositions are characterized by a melody or theme and the specified progression of chords, while the central part is often covered by the solos in a cyclical rhythmic form [4]. Improvisation is a process of elaboration of a melodic line, specified in the lead sheet or score [4]. During this action, the music sheet is a constant reference for the rhythm and the harmony, but the melody is subject to a live reinterpretation. Furthermore, the improvisation is the moment in which the performer may show its virtuosism and its personal style, that is expressed by the use of special musical effects, such as vibrato or slide, or dynamic emphasis. This very peculiar characteristic influenced Jazz music itself; so for the same composition it is possible to find recordings of the same artist quite different among them (like two performances of Blue Train by John Coltrane [5] [6]).

For this reason, one of the advantages of employing AMT to Jazz could be the relevant consequence for musicological studies of specific artists, for deepening their styles and performances.

Furthermore, due to its spontaneous and creative origin, improvisation appears not to follow any harmonic rule or pattern and for this reason could be a good benchmark for transcription [4].

Finally, the choice of focusing the research on the piano only has technical and conceptual reasons. In part, due to the diffused knowledge of the instrument and the large available datasets of synthesized and annotated recordings; in part, because of its double role in jazz music: the lead voice (performing the melodic part and the improvisations), and the accompaniment.

## 1.3 Aims and applications

Automatic Music Transcription converts audio recording into its parametric representation using a specific musical notation. Despite progress in the field AMT research, no end-user applications with accurate and reliable results are available and even the most recent AMT products are clearly inferior compared with human performance [7]. Although the Automatic Music Transcription of polyphonic music cannot be considered a solved problem, it is of great interest due to its numerous applications in the field of music technology.

Music notation is an abstraction of parameters. It is a precise organization of symbols just like words or letters are for languages and the Western music notation is still considered the most important and efficient medium of transmission of the music. Human music transcription is a complicated routine that requires high competence in the musical field and musical training, and it is also a time-consuming process [2]. A its most basic, AMT allows musicians to convert live performances into music sheets and thus easily transcript their performances [8] [9]. For this reason AMT is of special interest for musical styles when no score is available, e.g. folk music, jazz and pop [10] or for musicians that are untrained towards the western notation.

Another field in which AMT would be extremely useful is the analysis of non-written musical pieces, musicological one and musical education [9] [11] [12]. It would allow the investigation of improvised and folk music, simply by retrieving information on the melody. This last application is of particular interest for jazz case study, due to the wide use of improvisation during live performances. A clear example is represented by the two performances of Blue Train by John Coltrane, very different from each other despite the same lead sheet: the first is the live concert in Stockholm in 1961 [5], the second is the mastered piece [6]. It would allow us to focus on the detection of personal fingerprint of an artist.

Musical notation does not only allow the reproduction of a musical piece, but it also allows modification, arrangement and processing of music at a high level of abstraction. Another interesting application of music transcription, that has emerged in the last few years, is structured audio coding [10]. Structured audio representation is a semantic and symbolic description of ultralow-bit-rate transmission and flexible synthesis that uses high-level or algorithmic models, such as MIDI [13].

More recently it has been applied to the automatic search and annotation of new musical information, or to interactive musical systems, such as computers actively participating in live performances, score following or rhythm tracking. In fact, this kind of support of live performances would be very helpful for the musicians, who could freely express themselves without bothering about the musical annotations from where their musical inspiration flow. On the other hand, musicians usually see automatically created compositions and accompaniment as not achieving the same level of the quality. The query by humming MIR task is one of the most recent applications of the AMT to the slofége of a melody, where the output of an AMT can be used as a query for an interactive musical database.

## 1.4 Structure of the thesis

Chapter 2 presents an overview of the main Music Information Retrieval tasks and methodologies applied to the jazz genre. It begins with the description of what MIR and Automatic Music Transcription are. Then it goes on to explain in depth how transcription can be broken down into MIR tasks. The third chapter is dedicated to the general concepts and terminology useful for the understanding of the following chapters. It gives a brief overview of the main application considered in the thesis, as well as musical sound formal description, deep learning, MIDI and Music Information Retrieval. Finally, it describes different approaches in which to deal with the problem of Automatic Music Transcription.
The subsequent sections represent the core of the thesis and explain the method applied, the proposed dataset and finally present results and considerations.
Chapter 4 is focused on the description of the approach used in the thesis, based on machine learning, and presents the complex workflow required for obtaining results, through the various system blocks. And finally it provides an in-depth explanation of the pre-processing and Neural Network stages.
Section 5 is dedicated to the presentation of the dataset creation process, starting from the collection of raw MIDI files, passing through the analytical phase, ending with the synthesis. The chapter also deals with the main design choices made during the building of the dataset and with the main software used.

The final two sections are dedicated to the explanation of the metrics to obtain the results, the evaluation to the presented system and ends with a summary of the main contributions to the thesis, offering a perspective on improving this system and the potential application of the transcription system.

# Chapter 2

# Related works

The following chapter gives an overview of Automatic Music Transcription problem. The first section is focused on the description of the transcription problem with historical references, to give an idea of the evolutive trend. The remaining sections are dedicated to the subtasks of state-of-the-art Automatic Music Transcription presentation. A brief description of the single pitch estimation problem was inserted for the sake of completeness. Despite not being the focus of the thesis, it should give a good explanation of the differences in the multi-pitch estimation. Finally multi-pitch estimation state-of-the-art is presented in depth to evaluate different algorithms and methodologies.

## 2.1 Overview on Automatic Music Transcription

Automatic Music Transcription is thought to be the process to translate an audio signal into one of its possible parametric representations. As explained in section 1.3, there are many applications in this research area, in particular applied to musical technology and to musicology.
Automatic Music Transcription is deeply linked to Music Information Retrieval. The latter is defined as a research field focusing on the extraction of features from a musical signal. These features need to be meaningful to the task of understanding musical content as the auditory human system does. The features extracted can be of different levels depending on the type of information contained.

Actually, Automatic Music Transcription can be decomposed into sub-tasks related to the mid-level features of MIR. Onset and offset detection, beats, downbeats and tempo estimation are some of them.
The main tasks for an AMT system are the pitch estimation and the onset detection. This chapter focuses on these problems and the different approaches used in the state-of-the-art methodologies.

## 2.2 Automatic Music Transcription history

First attempts of AMT systems being applied to polyphonic music date back to the seventies. They showed many limitations concerning polyphony level and a number of voices. Moorer [1] method was based on the autocorrelation of output from a comb-filter, delaying the same input signal to find a pattern in the signal. Blackboard systems came into the AMT scene at the end of the century. Martin [14] [15] proposed a method based on five levels of knowledge ranging from the lowest to highest ones: raw track, partials, notes, melodic intervals, and chords. Blackboard systems are hierarchically structured, and the integrated scheduler determines the order of the action to perform. Nowadays most of the technique can be related to Probability-based techniques, Spectrogram Factorization, Machine Learning, and Signal Processing ones.
A straightforward procedure is offered by the Signal Processing approaches like the Klapuri one [16]. Klapuri proposed in the starting year of 2000 a method where fundamental frequencies, once estimated from the spectrum of the musical signal, are removed from the signal iteratively.
During the same period, more complicated techniques were employed in order to tackle the probabilistic character of the signal. The Goto [17] approach introduced a method for detecting predominant fundamental frequencies taking into consideration all possibilities for $F_0$.
More recent works exploit spectrogram factorization techniques like Non-Negative Matrix Factorization.
As the maths could suggest, the magnitude spectrum of a short-term signal can be decomposed into a sum of basis spectra, representing the pitches. They can be fixed by training on annotated files, or estimated from observed spectra. NMF estimates the parameters of the model. The Vincent [18] method (2010) used harmonicity and spectral smoothness constraint for an NMF-based adaptive spectral decomposition.

The probabilistic variant of NMF is PLCA studied by Smaragdis [19] (2006) and Poliner [20] (2010).
Finally, machine learning techniques seem to be the most promising and generalizable methods, capable of achieving better performances in terms of reliability. The Support Vector Machine, a supervised machine learning algorithm, was used by Poliner and Ellis [21] (2006). They trained the SVM on spectral features to have a frame-based classification of note instances. HMM was used to introduce temporal constraint during the elaboration of detected note events. Sigtia et al. [22] (2014) exploited the RNN capability of capturing temporal structure in data and MLM to generate a prior probability for the occurrence in the sequence to improve the transcription.

## 2.3 Single-pitch estimation

Single pitch estimation refers to the detection of the pitch in monophonic tracks, where monophonic means that no more than one voice and pitch at a time can be present in each time frame. This massive initial hypothesis simplifies the task and the problem of single pitch estimation both for the speech and for the musical signals is taken as solved.
Chevigné's dissertation [23] takes an overview on various single-pitch methods dividing them into framework using spectral components, temporal ones, and spectro-temporal ones.
Spectral methods use spectral components relying on the analysis of the frequency element within each note. Since musical sounds are considered quasi-harmonic signals, it can be stated that partials will be at integer multiples of the fundamental frequency. The energy spectrum should have a maximum indicating the fundamental frequency. Different spectral methods exploits different algorithm for the detection of pitch. Autocorrelation, cross-correlation and maximum likelihood functions are the most used depending on the representation employed. Spectral techniques are affected by harmonic errors placed at integer multiples of the fundamental one.
Temporal methods make use of autocorrelation function for the estimation of the fundamental frequency from raw audio signals. Due to the periodicity of audio signals, peaks in the function indicate the targets. Among them, fundamental frequency of the waveform would be the first peak, the other represents sub-harmonic errors due to higher harmonic components.
Spectro-temporal methods merge the two techniques to avoid errors derived

from their approaches. The signal is segmented in short frequency range as in Hewitt's work [24]. The proposed model exploits the human auditory system making use of a log-spaced filter-bank in the first stage of the framework. An autocorrelation function will detect the pitch for each channel of the filter-bank. A summary autocorrelation to merge all the information from all the frequency bands is required for overall results.

## 2.4 Multi-pitch estimation

Polyphonic music is usually characterized by multiple voices or instruments and multiple concurrent notes in the same time-frame. This led to the impossibility of making any assumption about the spectral content of a time frame. A couple of papers highlight a good division of multi-pitch estimation methods, as for single-pitch estimation, in temporal, spectral and spectro-temporal methods. The paper [23] suggests that most of the methods are based on the spectral features analysis. However, within the same set of representations, it also focuses on many differences concerning the used technique. Furthermore, the paper from Benetos [25] explains in depth all the state-of-the-art methods separated by the employed techniques.
The pitch extraction method can be a way of classifying different multi-pitch estimation approaches. As a matter of fact, joint algorithms are computationally heavier than iterative algorithms. Unlike the iterative ones, joint methods do not introduce errors at each iteration. In fact, iterative methods extract pitch at each iteration usually subtracting the estimated pitch till no more fundamental frequencies can be detected. These kinds of methods tend to accumulate errors but are really light computationally speaking. On the other hand, joint algorithms try to extract a set of pitches from the single time-frame. In table 2.1, some multi-pitch estimation methods are divided according to which kind of time-frequency representation is used.

| Time-Frequency Representation | Citation |
|---|---|
| Short-Time Fourier Transform | Klapuri [16] Yeh [26] Davy [27] Duan [28] Smaragdis [19] Poliner [21] |
| Constant-Q Transform | Chien [29] |
| Constant-Q Bispectral Analysis | Argenti [27] |
| Multirate Filterbank | Goto [12] |
| Resonator Time-Frequency Image | Zhou [27] |
| Specmurt | Saito [30] |
| Wavelet Transform | Kameoka [31] |
| Adaptive Oscillator Networks | Marlot [10] |

*Table 2.1: Multiple-F0 estimation approaches organized according to time-frequency representation employed*

One of the most used representations is the Short-Time Fourier Transform, because of the easy fruition of a fast and robust algorithm and because of the deep technical knowledge of that method. However, the Short-Time Fourier Transform has the main problem of using a linear frequency scale. To overcome this issue other representations can be used like Q-transform, which employs a logarithmic frequency scale using constant ratio between harmonic components of a sound, or other kind of filter-banks. Finally, the specmurt is a representation of the signal based on the inverse Fourier Transform of a spectrum calculated on log-frequency.

| Technique | Citation |
|---|---|
| Signal Processing | Argenti [32], Klapuri [9], Yeh [26], Saito [33], Zhou [31] |
| Maximum Likelihood | Goto [12], Kameoka [3], Duan [28] |
| Bayesian | Davy [27] |
| Support Vector Machine | Poliner [21], Chien [29] |
| Neural Network | Böck [34], Marlot [10] |
| Spectrogram Factorization | Smaragdis [19] |

*Table 2.2: Multiple-F0 techniques organized according to the employed technique*

Table 2.2 was built following the division concerning the used technique for the multi-pitch extraction. In particular, it shows the majority of methods exploits signal processing techniques. In this case, extracted audio features are used in the multi-pitch estimation without the help of any learning algorithm. Methods based on the spectrogram factorization, such as Non-Negative Factorization Matrix, are more recent. Those kind of algorithms try to analyze the input space in order to decompose the signal representation in time-frequency. Other methods rely on probability concerning signal field, exploiting Bayes formulation of a problem and using Monte Carlo Markov Chain in order to reduce computational costs and Hidden Markov Model as a post-processing system for note tracking.
Finally, learning algorithms are now increasingly used and seem to be the more promising methods. Supervised learning procedures such as Support Vector Machine for the multiple-pitch classification can also be found. For what concerns the unsupervised learning algorithm Gaussian Mixture Model and any Artificial Neural Network can easily be found in the literature.
The following sections will go through each multi-pitch estimation technique highlighting the merits and defects of each one.

### 2.4.1 Signal processing techniques

Signal processing techniques are probably the most widespread for the detection of pitches within a single time-frame. The input signal is processed for the extraction of the representation which can be in the temporal or in the frequency domain. The detection of pitches is computed using a pitch salience function, also called pitch strength function, or a set of possible pitches.
Klapuri [16] exploited the smoothness of a waveform computing Magnitude Power Spectrum and filtering it with a moving average filter for the noise suppression. The Klapuri method is based on spectral subtraction with the inference of polyphony. A pitch salience function applied within a specific band, estimating the pitch from the spectrum. It also calculates the spectrum to subtract it from the input signal in order to exclude the detected pitch from the input signal. Yeh [26] developed a joint multi-pitch estimation algorithm basing the detection of the fundamental frequencies on a set of candidate pitches. The used spectro-temporal representation is the Short-Time Fourier Transform. A pre-processing stage is employed to estimate the

noise level present in the signal in an adaptive fashion. The pitch candidate score function takes into account different parameters like harmonicity features, mean bandwidth of the signal, spectral centroid, and synchrony. With reference to iterative methods, some were developed like the Zhou [27]. Zhou used a filter-bank composed of complex resonators that should approximate pitch representation. The energy spectrum is used as a representation of the audio signal. Rules for the iterative elimination of candidates pitches are based on the number of harmonic components detected in each pitch and a measure of spectral irregularity. Another mid-level representation used is the specmurt. It consists of the inverse Fourier Transform of the power spectrum computed in a logarithmic fashion. Saito [35] proposed a method based on it where the input spectrum can be seen as the convolution of harmonic structures and pitch indicator functions. On the other hand, the deconvolution of the spectrum by the harmonic pattern, results in the estimation of the pitch indicator function. This last stage is achieved through the specmurt analysis, detecting iteratively notes. Representation exploiting log-frequency scale are also used in order to improve the methods, such as the Q-transform representation. Argenti [29] proposed a method using both Q-Transform and b-spectral analysis of the input signal.
Signal processing techniques are computationally lighter than other techniques and were the first to be applied due to their simplicity. However, to reach performances of more complicated techniques, ah-hoc hypothesis needed to be done to improve the raw systems. For this reason they still remain less prone to a generalization to but different type of data.

### 2.4.2 Statistical techniques

Statistical methods rely on basic principles of statistics to analyze dependencies of the signal from itself. Usually, it is a frame analysis where given a frame v and the possible fundamental frequencies combination, C the problem of estimating multiple pitches can be formulated as a Maximum a Posteriori problem. The MP formula: $\hat{C} = argmax_{C \in \mathbf{C}} P(C|v)$ indicates with $\hat{C}$ the estimated set of pitches and with $P$ is the probability to estimate the pitch set $C$.

If, instead, we do not have any prior information about the mixture of the pitches, the problem can be seen as a Maximum likelihood estimation problem exploiting the Bayes rule:

$$\hat{C} = argmax_{C \in \mathbf{C}} \frac{P(v|C)P(C)}{P(v)} = argmax_{C \in \mathbf{C}} P(v|C).$$

The model proposed by Davy and Godsill [32] makes use of Bayesian harmonic models. This technique models the spectrum of the signal as a sum of Gabor atoms. The parameters for the unknown model of Gabor atoms are detected using a Markov Chain Monte Carlo.
The method proposed by Kameoka [31] takes as input a wavelet spectrogram and the partials are represented by Gaussian placed in a frequency bin along the logarithmic distributed axis. A Gaussian-mixture model tries to identify partials taking into account the synchrony of partials. Pitches are extrapolated using the Expectation-Maximization algorithm.
Statistical multiple-pitch estimation methods for the modeling region with and without peaks use the Maximization likelihood approach. The one proposed by Duan [33] is based on a likelihood function, composed of two complementary regions. One where there is the probability of detecting a certain peak in the spectrum given a pitch, and the other where there is the probability of no detection of peaks. The stage dedicated to the pitches estimation makes use of a greedy algorithm.

### 2.4.3 Spectrogram factorization techniques

The main spectrogram factorization models are the Non-Negative Matrix Factorization and the Probabilistic Component Analysis. They aim to cluster automatically columns of the input data.
NMF tries to find a low dimensional structure for patterns present in a higher dimensional space. The input matrix V is decomposed in W atoms basis matrix and H atom activity matrix. The distance between the input matrix and the decomposed one is usually measured with the Kullback-Leibler distance or the Euclidean distance. A post-processing phase is used to link atoms to pitch classes and to sharpen the onset and offset detection for the note events.
PLCA introduced by Smaragdis [19] is the probabilistic extension of NMF using a Kullback-Leibler cost function. The input representation, the spec-

trogram, is modeled as the histogram of independent random variables distributed accordingly to the probability function. The latter can be expressed by the product of the spectral basis matrix and the matrix of the active components.
It represents a more convenient way of incorporating prior knowledge on different levels inducing a major control on the decomposition. $P(\omega|z)$ is the spectral template at z component, and $P_t(z)$ represent the activation of $z^{th}$ component. PLCA is expressed as $P_t(\omega) = \sum_z P(\omega|z)P_t(z)$. $P_t(\omega)$ is the estimation of parameters performed through the Expectation-Maximization algorithms. Due to the temporal constraint on both NMF and PLCA algorithm, they can not be applied to non-stationary signals. For that reason, alternatives to these methods were developed, such as the Non-Negative Hidden Markov Model. In the Non-Negative Markov Model, each hidden state is linked to a set of spectral components in order to be used by them. The input spectrogram is decomposed as a series of spectral templates per component and state. Thus, the temporal constraint can be introduced in the framework of an NMF, modelling a non-stationary event.

### 2.4.4 Machine learning techniques

Despite the previous year's machine learning algorithms not being given too many chances, the number of methods applying them is increasing. Good results and the potential they are showing in the latest research seem to be continually growing.
Chien Jeng [3] proposed a frame-based method applying a supervised machine learning algorithm to signal processing data. The method exploits the Q-Transform time-frequency representation, trying to solve octave errors, and it makes use of the classification procedure called Support Vector Machines. Each pitch is characterized by a single dedicated class.
A really interesting paper was the one about the comparison between different Neural Networks written by Marlot [10]. He uses Neural Networks of different natures working on the same kind of input data to understand which one could be the best performer. The outcome of his study founded that the Time-Delay Neural Network got the best performance parameters. Musical strong time correlation is exploited by the neurons of the Time-Delay Neural Network, outperforming the other types of Neural Network.
An algorithm using Short-Time Fourier Transform for time-frequency anal-

ysis was the one proposed by Poliner and Ellis [21]. It exploits a frame-based method focused just on the piano note classification. The classification method was used jointly to a Support Vector Machine, and the multi-pitch task is also supported by a Hidden Markov Model. The latter helps the improvement of the classification system during a post-processing stage.
Other interesting types of unsupervised learning were used in the work of Böck and Schedl [34]. This work makes use of Recurrent Neural Networks focusing on the polyphonic piano transcription. The proposed Neural Networks is made of bidirectional Long-Short Time memory atoms that are of consistent help in the note classification and onset detection task. The input of the Neural Network is represented by the output of a semitone spaced filter-bank analyzed with a long and short window.

## 2.5 Trend and future directions

Automatic Music Transcription is considered by many to be the Holy Grail in the field of music signal analysis. Its core problem is the detection of multiple concurrent pitches, a time-line of which was given in section 2.2.
As can be observed, before 2006 common approaches were signal processing, statistical and spectrogram factorization. Furthermore, within the MIREX context [36], best performing algorithm was the one proposed by Yeh in 2010 [37], reaching an accuracy measure of 0.69. Despite significant progress in AMT research, those types of systems are affected by a lack of flexibility to deal with different target data.
The work proposed by Benetos et al. [7] takes an overview of multi-pitch estimation techniques which are state-of-the-art. Benetos et al. analyzed the results and the trend of proposed systems, pointing out how performances seemed to converge towards an unsatisfactory level. Furthermore, they tried to propose techniques to 'break the glass ceiling' of reached performances, such as the insertion of specific information into the transcription system.
With the development of Machine Learning techniques, new perspectives seem to open up. Last years MIREX results highlighted how approaches employing Neural Networks are achieving better performances. Indeed, the Böck [34] ranking in MIREX can be a clear signal of how promising are Neural Networks methods. The Böck approach clearly outperforms the system of Poliner and Ellis [21], highlighting its good generalization capability. Furthermore, the system also performs better than the one of Boogaart and Lienhart

[38], which was trained with a single MIDI instrument. This is remarkable, since Böck system is not trained specifically for a single instrument. These observations demonstrate better perspective for Machine Learning methods compared to others.
Furthermore, Neural Network framework is widely applied in many research areas from video to audio analysis. Within the audio field, it is applied also to transcription of different instruments, such as the piano, as in the Böck work, or drums, as in the Vogl et al. work [39]. The latter certify the flexibility of the framework.
Furthermore, Marlot [40] guided an interesting study on different types of networks. He highlighted how networks accounting for time correlation, such as Recurrent Neural Networks, due to high correlation of audio signals, can retrieve better performances.

# Chapter 3

# Background and terminology

Musical Information Retrieval is the general field under which Automatic Music Transcription can be categorized. Musical Information Retrieval is the science that tries to extrapolate meaningful musical information from a music signal. Thinking of Automatic Music Transcription, its aim is to retrieve a parametric representation of the audio signal.
Although AMT is a Music Information Retrieval task, it also goes through a different field of the music technology and it requires different knowledge taken from different disciplines: Acoustic, Music theory, Digital Signal Processing, as well as Computer Engineering.
The current chapter is dedicated to principal important technologies and background notions employed during the development of the method. Musical characterization, MIR sub-tasks, MIDI, Neural Networks and Madmom library are explained in depth.

## 3.1 Musical sounds

AMT sub-tasks need a unique representation to describe precisely a musical sound. It can be characterized by four base attributes: pitch, loudness, duration, timbre [25]. If duration can easily be described as the duration of a signal in time till the imperceptibility of it, the same cannot be said for the other three attributes. In this section, we will focus on these attributes giving a rough description of the signal basis theory.

### 3.1.1 Pitch

Musical sounds are a sub-set of the acoustical signals and can be approximated as harmonic, or better nearly-harmonic signals.
In the frequency domain, harmonic sounds are signals described by a set of frequency components. The lowest harmonic component is called fundamental frequency $F_0$, the other sound components, called harmonics, play the role of enhancing the signal. Harmonics, in harmonical signals, are placed at integer multiple of the fundamental frequency, following the equation $kF_0$, where k is greater than one and belongs to an integer number set. Regarding near-harmonic signals, the harmonics are not at precise multiple integers, but they differ from a value depending on the nature of the instrument.

$$f_n = nF_0\sqrt{(1+n^2B)/(1+B)}$$

The formula represents the distribution of fundamental frequencies, where $B = 0.0004$ is the inharmonicity coefficient for a pinned stiff string in a piano.
In the case of piano, extensively employed in this work, the sound can be characterized as quasi-harmonic and pitched. For this reason, it can be analyzed over the physical viewpoint thanks to the pitch attribute. The pitch represents the perceived component of a sound wave, expressed in a frequency scale. I.e. the fundamental frequency that refers to the physical term, measured in Hertz and defined for periodic signals. As reported by Hartmann: "a sound has a certain pitch if it can be reliably matched by adjusting the frequency of a sine wave of arbitrary amplitude." [41].
To be more specific the pitch of a sound can be thought of as a subjective impression of the fundamental frequency of a sound, allowing us to identify a specific note on a musical scale.

### 3.1.2 Loudness

Loudness is the subjective perception of the sound intensity and is related to Sound Pressure Level, frequency content, and duration of the signal. The sensitivity of human auditory system changes as a function of the frequency and not only as a function of the SPL as shown in the plot 3.1. The figure 3.1 shows the diagram of Fletcher and Munson and two main thresholds can be detected. The hearing threshold indicates the minimum sound level

perceivable to the human ear. The pain threshold represents the maximum sound level that a human ear can perceive without feeling pain. The other lines, called isophonic curve, show the SPL required for each frequency to be perceived at the same loudness. We can also observe from the diagram that the ear was thought to perform at its best in the speech range between 1 kHz and 4 kHz. However, it has a minimum perceivable pitch of 20 30 Hz while a maximum of 15-20 kHz.
In the Fletcher and Munson diagram the Loudness Level is also indicated, expressed in Phon, for each isophonic curve at 1 kHz.

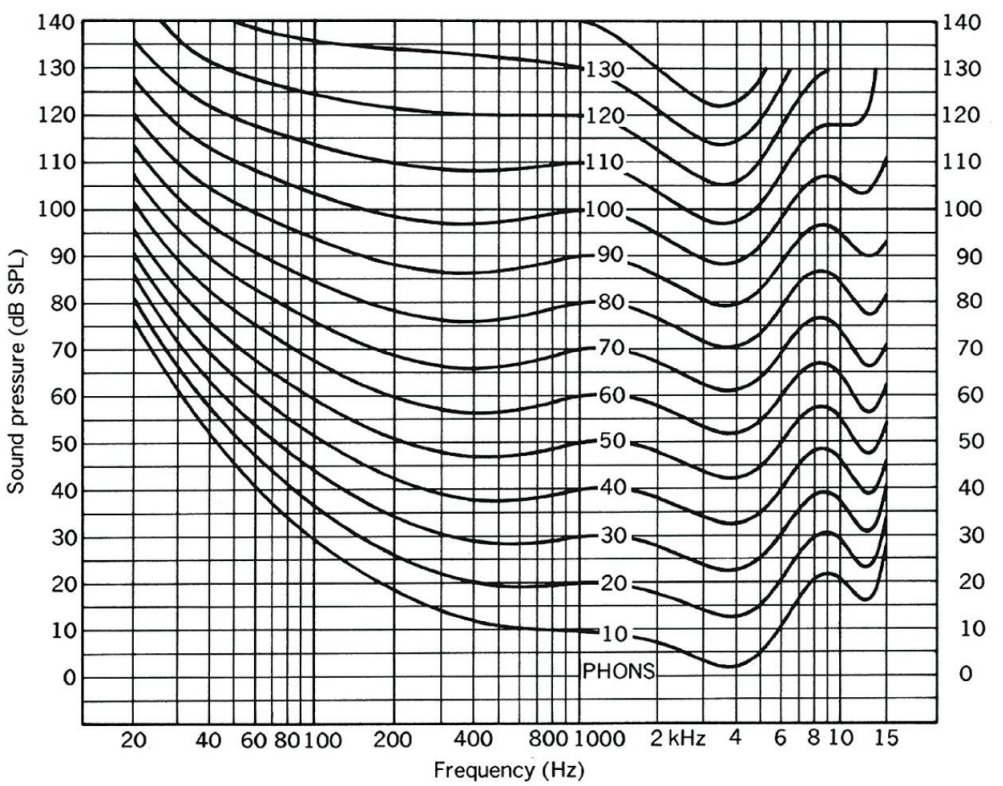


*Figure 3.1: Fletcher Munson diagram*

### 3.1.3 Timbre

In a situation where two sounds have identical pitch, loudness, and duration, they could not be distinguished but thanks to the timbre character they can be. Timbre is a general character of a sound, usually attributed to the sound of an instrument. It denotes a digital fingerprint of all the sounds of an instrument.
From the Acoustical Society of America, the Acoustical Terminology defines the timbre as "that attribute of auditory sensation which enables a listener to judge that two nonidentical sounds, similarly represented and having the same loudness and pitch, are dissimilar. The timbre depends primarily upon the frequency spectrum, although it also depends upon the sound pressure and the temporal characteristics of sound" (Acoustical Society of America Standards Secretariat 1994).
The timbre is used to define the color or the quality of the sound. It is closely influenced both by the time evolution (attack, decay, sustain, release time) and by the spectral components in a sound.

### 3.1.4 Rhythm

The temporal relation between events is described by the rhythm. The perception of it is linked to two different factors: the grouping, which is more formal measure, whereas the meter, is a more perceptive one. Indeed, grouping refers to hierarchical division of a musical signal in the rhythmic structures of variable length.
A group can be extended from a set of notes to a musical phrase to a musical part. On the other hand, meter refers to regular alteration between a strong beat and a weak beat heard by the listener. Pulses or beats do not have an explicit assignment in the music but are induced by the observation of a rhythmic pattern underlying the musical structure.
The main measure to define the rhythm of a song is the tempo. Tempo defines the rate of the most prominent among the pulses and it is expressed in Beat Per Minute. Indicating with Tactus the beat, i.e. the measured tempo reference for each individual event, the Tempo can be expressed as the time rate of the Tactus. The shortest time interval between events in the track is called Tatum and constitutes the base structure of it. Finally, bars refer to harmonic changes and rhythm pattern changes. The number of beats in every measure is called time signature.

## 3.2 Music information retrieval

Music Information Retrieval (MIR) is "a multidisciplinary research endeavor that strives to develop innovative content-based searching schemes, novel interfaces, and evolving networked delivery mechanisms in an effort to make the world's vast store of music accessible to all", as defined by Downie [42]. The quote of Downie is explicative of the wideness and of the potential that the Music Information Retrieval has in today's world. Due to the increasing number of streaming services and the consequent availability of mobile music, the interest concerning the Music Information Retrieval is quickly increasing. It is mainly focused on the extraction and the inference of meaningful features from music, on the indexing of music, and on the development of the scheme for the retrieval and the research of data. Of particular interest during this work are those MIR applications referred to feature extraction. Indeed, in the case of Automatic Music Transcription methods, descriptors of audio signals play a central role in understanding musical contents. In fact, AMT systems can be decomposed as MIR tasks linked to mid-level features. Note onset and offset detection, beat tracking or tempo estimation represent the actual research field needed for a complete transcription.
Onset detection has the aim to identify the start point of an event, called onset. More specifically the onset detection needs to identify the starting point of all the events within a musical signal. Depending on the instrument being played, onsets can be divided into three categories: pitched, percussive or pitched-percussive. The first is typical of string instruments or wind instruments; percussive ones are produced by drums; finally, pitched-percussive onsets characterize instruments such as the piano or guitar.
Facing polyphonic music with multiple voices complicates the onset task, since every voice has its own onset characteristic. Furthermore, onsets can be modified for aesthetical purposes using musical effects like the tremolo and the vibrato or other audio effects. Usually, modifications constitute interferences in the onset detection task. For this reason, there are some methods focused on the onset detection suppressing vibrato like the one proposed by Böck [34].
Metrical organization of musical tracks follows hierarchical structure from the lower level, the beat, to the higher one, the time signature. The beat is the reference for each musical event and constitutes the most important rhythmic element. A pre-stated number of beats form a bar, the number of

the beats that need to be present in a bar is indicated by the time signature or meter. On the other hand, downbeats are meant to be the first beat inside a bar and it can be linked to rhythmic patterns or harmonic changes within a musical piece.
Linked to the beat tracking, the tempo estimation task has the aim to recognize accurately the frequency at which beats occurs. Although theory could suggest deriving the tempo of a musical piece from the beat estimation, which is not as easy as it might appear. Tempo hypothesis is needed for a robust and good beat estimation algorithm.

## 3.3 MIDI

MIDI, which stands for Music Instrument Digital Interface, represents a communication digital language. It works with specifics that make possible the communication between different devices inside a cabled network. Finally, the MIDI is a medium to translate events linked to a performance or a control parameter, such as pressing a key on the keyboard. Those messages can be, then, transmitted to other MIDI devices or can be used later on after the recording.

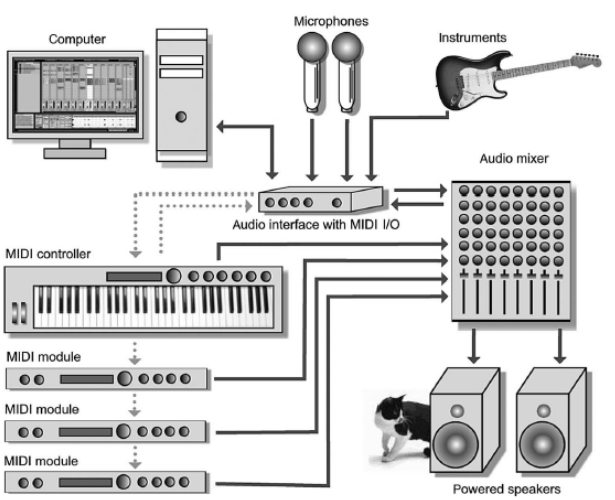


*Figure 3.2: MIDI network scheme*

A basic device within the MIDI environment is the sequencer. A MIDI sequencer can be software or hardware, which is used to record, modify and

send to the output MIDI messages in a sequential way. The MIDI messages are usually divided per track, each one dedicated to a different instrument as required by the producing concept. MIDI Tracks contain events and messages working on a specific channel. Once the performance has been recorded, it is stored and can be arranged or modified also with the help of graphical interfaces depending on the sequencer. Data is then stored in a file or a digital audio workstation to be played back or reused in different ways.
Most used sequencers are the software ones due to their portability through different Operating Systems. They exploit the versatility, the calculus speed and the memory of a personal computer.
From an artistic viewpoint, thanks to its flexibility, the MIDI language is a really important medium for artists. Once the MIDI has been recorded and mastered it can overtake the analogic difficulties and the recorded performance can be edited and controlled. In this dissertation, exploiting the power of the standard, we will use the MIDI annotation to use different piano or instrument sounds on the same performance in order to have more variability inside the dataset.
It is important to remember that the MIDI does not have inside itself any sound information and does not communicate any audio waves or create any sound. It is a language to transmit instructions to devices or programs to create and modify the sound. This is the great strength of the MIDI standard since it permits the files to be very lightweight.
A Standard MIDI file can be of three formats:

1. Format 0: all the tracks of a song are merged in a unique one containing all the events of all the tracks of the original file;

2. Format 1: tracks are stored separately and synchronously, meaning that each track shares the same tempo value. All the information about tempo and velocity of the song are stored in the first track, also called Tempo Track. It is the reference point for all the other tracks;

3. Format 2: tracks are handled independently also for the tempo managing.

Within a MIDI track, every event is divided from other events by temporal data called Delta-time. Delta-time translates into byte, the time between two occurring events, so it represents the duration in Pulse Per Quarter Note between an event and the following one. PPQN is the duration in a

microsecond of an impulse or also called tick per a quarter of note. It is given by the following equation: $\frac{60000000/bpm}{PPQN}$. The Beat Per Minutes represents the metronome time of the song, while the PPQN is the resolution in impulses for a quarter of a note.

### 3.3.1 MIDI messages

The medium for the communication between devices within the MIDI network is called MIDI messages, transmitted along serial MIDI lines at 31,250 bit/sec, where MIDI cable is unidirectional. Data in a serial line follow a unique direction in a conductor cable, while in a parallel line data can be transmitted simultaneously to all the devices connected.
In MIDI messages the Most Significant Bit, left one, is dedicated to identifying the kind of byte. Bytes of MIDI messages could be Status Byte if the MSB is set to 1, or Data Byte if MSB is set to 0.
To permit different kinds of connections between devices and different instruments, guidelines were specified, following those specifications a device can transmit or respond to messages depending on its own internal settings as specified in the figure 3.3. As a matter of fact, there is a different mode in which an instrument can work. The Base Channel is an assigned channel and determines which channel the device would respond to.
Now we will take a deep view of the mode in which a MIDI device can work:

- Mode 1: Omni mode On, Poly mode On, the instrument will listen to all channel and retransmits the messages to the device set at the Base Channel. In this mode, the device acts as a relay of input messages in a poly mode. It is rarely used.

- Mode 2: Omni mode On, Mono mode On, the instrument will listen to all the channel and retransmits the messages to the device or instrument set at the Base Channel, the latter can act just as monophonic device. In this mode, the device acts as a relay of input messages in a poly mode. It is even rarer than the Mode 1 since the device cannot detect the channel nor play multiple notes at the same time.

- Mode 3: Omni mode Off, Poly mode On, the instrument would respond just to the assigned Base Channel in a polyphonic fashion. Data from a different channel from the Base Channel would be ignored. It is the most common mode due to the fact that voices within the multitimbral

device are controlled individually through messages on the channel, reserved for that voice.

- Mode 4: Omni mode Off, Mono mode On, the instrument would listen to the assigned Base Channel, but every voice is able to play a unique note per time. A really common example is the recording system for a guitar, where each data is transmitted in a monophonic way on one channel, one for each string, as a matter of fact, one cannot play multiple notes on a single guitar string.

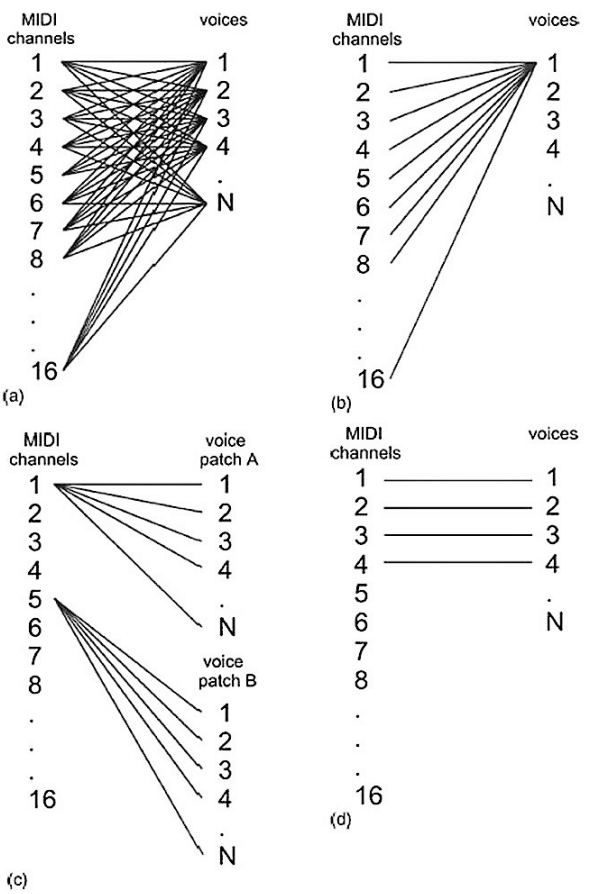


*Figure 3.3: Voice channel assignment of the four modes that are supported by the MIDI: top left Omni on/poly; top right Omni on/mono; bottom left Omni off/poly; bottom right Omni off/mono*

### Channel voice messages

Channel voice messages are used to transmit real-time performance data through a MIDI cabled system. Every time a parameter or a controller of a MIDI instrument is used, selected or changed by the performer, a channel voice message is emitted. Below are specified some of the most used channel voice messages:

- Note-On: used to denote the start of a MIDI note, it is generated every time a key is triggered on a keyboard, a controller or on other instruments. Status Byte contains the Note-On status and the midi channel number; Data Byte to specify which of the 128 MIDI pitch note needs to be played, one Data Byte to denote attack velocity of the pressed key or the pressure, the volume of the note is affected by the latter. MIDI note is contained in the interval from 0 to 127 knowing that in position 60 is placed the C4, to give an example the keyboard has 88 keys and its MIDI note interval comprehends numbers from 21 to 88. In the specific case in which a note has an attack velocity 0, the Note-On events is equal to a Note-Off. This peculiar use of the Note-On message was exploited in the project to modify easily the MIDI files without deleting any of the events.

- Note-Off: is the message to stop a specified MIDI note. The sequence of MIDI events is characterized by a sequence of Note-On Note-Off messages. The note-off command would not cut the sound, but it would stop the MIDI note depending on the release velocity parameter that represents how fast the key was released.

- Program-change: it is a message for specifying a change in the number of the program or pre-set which is playing. The program number usually define the MIDI instrument to play, pre-sets are usually defined by manufacturers or by the user to trigger a specific sound patch or a specific setup.

- All Notes-Off: since a MIDI note could remain played, All Note-Off message can be used to silence all the modules that are playing.

- Pressure/Aftertouch: it renders the double pressure on a key.

- Control-change: it is used to transmit information related to changes in real-time control or performance parameters of an instrument like foot pedals, pitch-bend wheels.

### 3.3.2 System messages

System Messages are forwarded to every device within the MIDI network, so there is no need to specify any MIDI channel number. Every device would respond to a System Message. Three are the types of System Messages in the MIDI Standard:

1. System Common Messages: they transmit general information about the file being played like the MIDI time code, the song position, the song selection, the tune request. Typical System Common messages are: MIDI Time Code Quarter-Frame,Song Select, End of Exclusive messages;

2. System Exclusive messages: are special messages left to the manufacturers, programmers, and design to make other devices of the same brand communicate without restriction of the length of data and MIDI messages customized;

3. Running Status messages: running status messages are a special type of messages used in a situation of redundancy of the same type of message. It permits a sequence of the same message type to omit the Status Byte, that would be the same for each one. If for example, we have a long series of Note-On messages on a specific channel with a Running Status message, we can omit the Status Byte.

## 3.4 Introduction to machine learning and Neural Networks

Nowadays, artificial Intelligence, AI, is widely used in many research areas not only automating routines but also in the field of discerning high-level of information. The true challenge for artificial intelligence was solving those tasks hardly describable for people due to their spontaneous and intuitive nature.
The Deep Learning term is linked to that way to approach an AI problem in which tasks requiring high-level concepts need to be decomposed in many lower-level terms [24].
One of the main reasons for the increasing use of deep learning in the last 20 years was the growing quantity of digitalized data. Training data is really important in deep learning. The process of digitalization of the society and the consequent start of the era of Big Data makes easier the resolution of Machine Learning problems. Indeed, Machine Learning algorithms end with good results when trained on a big amount of data.
Neural Networks are a framework to approach Machine Learning problems. They take inspiration from the human brain system: how it is composed, how it is connected and how its elements interact with each other.
Artificial Neural Networks are composed of interconnected processing units. The goal of the network is to find an approximated function $f^*$ that can map the input $x$ as the target $y$. ANN try to minimize the result of the function $f^*$ so to have $y = f^*(x)$. The processing units are also called artificial neurons of the network and they usually perform a sum on the weighted inputs they receive. The weight of each input depends on how much the input influences the neuron. The output of the weighted sum, called activation value, is usually modified by a bias value which is then passed as input to a transfer function.
The transfer function $\sigma(a)$ applied to the activation value calculated by the neurons is a non-linear function like a hyperbolic tangent, arc-tangent, and sigmoid.

$$\sigma(a) = \begin{cases} \frac{1}{1+e^{-a}}, & \text{sigmoid} \\ \tanh a, & \text{hyperbolic tangent} \\ \max(0, a), & \text{ReLu} \end{cases}$$

Additional non-linearities can be added depending on the purpose of the Neural Networks, for example, the softmax algorithm can be applied to classification problems.
The topology of a network refers to the way in which neurons are connected among each other to accomplish for example a pattern recognition problem. Usually, Neural Networks are organized following a layered structure. In each layer, the set of all the activation values from each neuron is called the activation state of the network, while the output state is the set of all the neurons' output related to a layer.
The training of the Network consists of adjusting the parameters of each neuron, weight and bias, in order to get an approximation of the output as close as possible to the desired output target, provided by the input data at the Network. Thanks to the layered structure, an iterative algorithm can be used during the training, such as the backward propagation of the error based on the gradient descent method. The gradient is the loss function and measures the difference between the real output of a neuron and its desired target output. Depending on the results of the gradient loss, the parameters of the neurons are updated to get the minimal error between the target and the actual output. Different loss functions are available for calculation such as cross-entropy function, the one chosen during the development of the NN used within the AMT system.
The minimization of the loss during the training phase can be performed in three different approaches. Batch Gradient Descent approach tries to minimize all the sets of data. Stochastic Gradient Descent applies single data to the Neural Network. Finally, Mini-Batch Gradient Descent use just a little subset for the training. In addition, to avoid problems such as local minima and to accelerate the training process, optimized Gradient Descend algorithms were developed like Nesterov, Adam, Adadelta or RMSprop [24].
The activation state of a network represents the activation values in a layer and determines the short-term memory of a network. On the other hand, long-term memory is represented by the learning process of adjusting weights [24].

A variant of Forward Neural Networks is the Convolutional Neural Network, figure 3.4. Its main characteristic is the fully connected convolutional layer. CNN was chosen in the dissertation as framework for the pitch detection task.

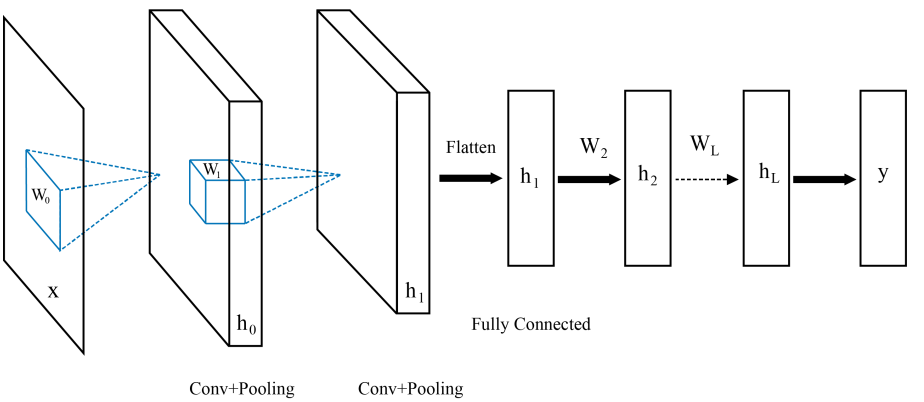


*Figure 3.4: Convolutional Neural Network scheme*

The extension of basic Neural Networks exploits the long and short correlation of the input signal with different types of Networks. Recurrent Neural Networks, figure 3.5, for examples, extend Forward Neural Networks with feedback connections, allowing the connection of previous layers. Feedback connection accounts for different time instant of the input relating the actual input to the past one, representing short time memory.

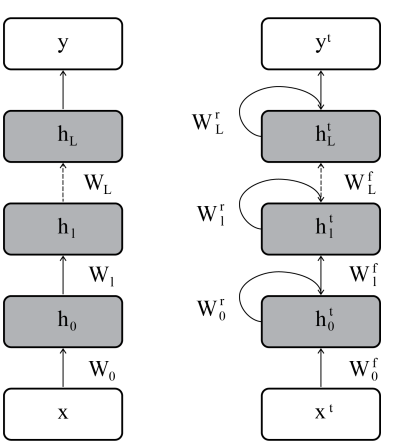


*Figure 3.5: Deep Neural Network scheme. Left: Forward Neural Network; Right: Recurrent Neural Network*

Long-term memory is exploited with the use of Long-Short Term Memory cells, figure 3.6, allowing the Neural Network to learn long-term dependencies. LSTM cells have an internal memory that can be accessed and updated from gates depending on the input they are fed with.

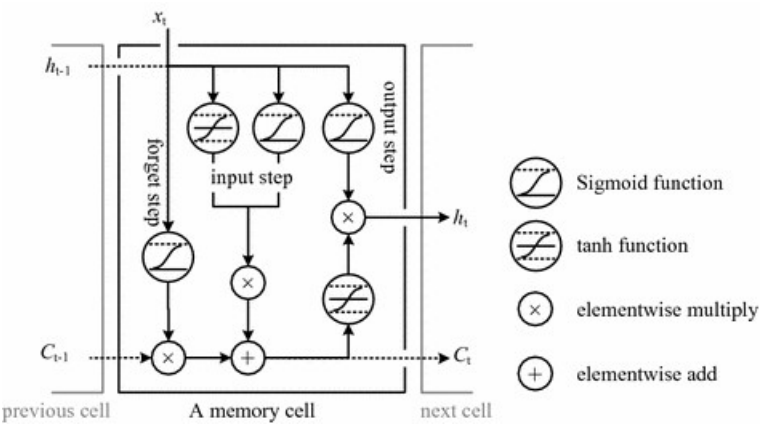


*Figure 3.6: Long Short-term cell*

## 3.5 Madmom library

Due to the emerging of the Music Information Retrieval research field in recent years, audio-based systems for the retrieval of valuable information have become more important. Furthermore, their role is still gaining relevance thanks to the increasing trend of available data.
Audio-based MIR systems in the state-of-the-art are designated as systems which make use of low-level feature analysis for retrieving meaningful musical information from an audio data. The latest audio-based MIR systems exploit Machine Learning algorithms to extract this information. Furthermore, they usually integrate a different level feature extraction sub-system to derive them directly from the audio signal.
Madmom library, as an open-source library, was thought to facilitate research in MIR field both in terms of low-level feature extraction, like Marsyas and YAAFE, and in terms of high-level extraction like MIRtoolbox, Essentia or LibROSA. What makes Madmom different from all the other libraries is the use of Machine Learning algorithms [34].
Madmom library would like to give a complete processing workflow allowing

the construction of both full processing systems and stand-alone programs using Madmom functionalities. Thanks to Processors objects, Madmom converts running programs into a simple call interface.
The use of Processors allows an easy use of complicated and long procedures included in the library as low-level feature extraction ones. High-level features are then used by Machine Learning techniques to retrieve musical information. Madmom includes both Hidden Markov Model and Neural Network methods applied to state-of-the-art algorithms of MIR tasks as onset detection, beat, and downbeat detection, and also meter tracking, tempo estimation and chord recognition. Availability of state-of-the-art techniques permits users to build a complete processing method or just integrate them in stand-alone programs as we have done.
Madmom is an open-source audio processing and Music Information Retrieval library based on Python language. Following the Object Oriented Programming approach, it encapsulates all the information within objects that instantiate subclasses of the NumPy class.
Madmom depends just on three external modules: one for array handling routines, NumPy, one for the optimization of linear algebra operation for FFT, SciPy, and finally one for the speed-up of critical parts, Cython. ML algorithms can be applied without any other third-party modules since they are algorithms pre-trained on external data, which are just tested with input data, allowing reproducible research experiments.
The source code for each file is available on the net and the complete documentation for the API can be found at http://madmom.readthedocs.io.
The Madmom library was extensively employed during the development of the Automatic Music Transcription system. In particular during feature extraction and peak detection phases (the latter applied during the pitch detection). It was exploited for its robust algorithms, and is easy to include within external code.

# Chapter 4

# Methodology

The proposed transcription system consists of three main phases as suggested from the figure 4.1: an initial signal-processing stage, followed by an activation function calculation and finally a peak picking one. In the first phase features are extracted from the audio signal and target values are also derived from the MIDI files. Feature extraction works on raw audio data, while target creation works on that of MIDI. During the activation function calculation phase, the Neural Network is trained on the same input features. The final stage is represented by the detection of onset employing a simple peak picking method. After the training of the Neural Network, the evaluation phase consists of the prediction extraction from the interpretation of the features. Target and evaluation prediction are compared to estimate the system.

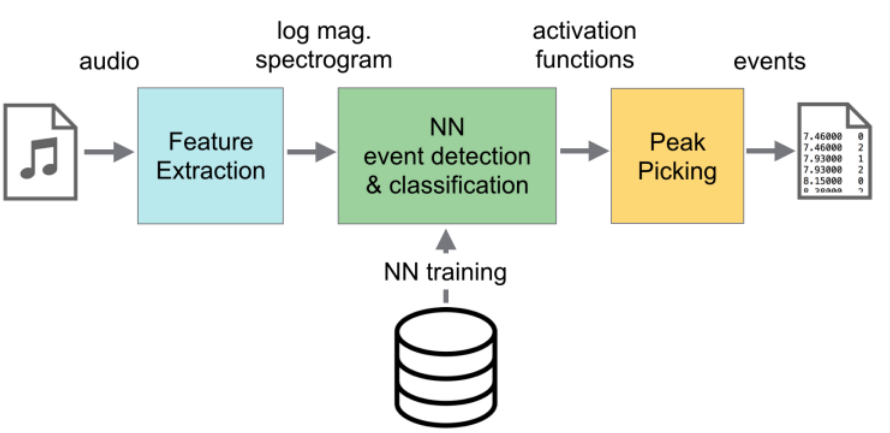


*Figure 4.1: System workflow*

The following sections describe the entire system workflow, narrowing down the analysis to the signal processing and the Neural Network training phases that can be referred to as the core of the method itself.

## 4.1 Design choices and considerations

Automatic Music Transcription systems aim at retrieving a new representation of the input signal starting from a different one, exploiting analysis methods. The Dixon method, for example, uses a time representation of the signal. But, usually, as seen throughout the relevant section 2.4, probabilistic and factorization algorithms rely on a time-frequency one. One of the main reasons which support the choice of undergoing through time-frequency analysis, the method employed to perform our study, is the simultaneous study of a signal under both time and frequency parameters. In fact, their tight coupling helps and supports the signal analysis. From a practical point of view the signal can be transformed from a one-dimensional signal to a two-dimensional one through the Fourier Transform, assuming the signals to be either infinite in time or periodic. A more realistic interpretation of the Fourier Analysis is the Short-Time Fourier Transform used to compute the Fourier Transform on short time-frames. STFT determines frequency content and phase in each local time-frame and was the one chosen in the dissertation. Other time-frequency representations are available for analyzing the signal like Q-transform, Wavelet transforms and Bilinear time-frequency distribution. In practice, Short-Time Fourier Transform and Q-Transform are the most frequently employed ones due to the availability of convenient computational algorithms and deep theoretical knowledge. STFT's main drawback is constant frequency resolution, which may generate problems analyzing lower frequencies. To overcome this issue, usually, a bank of filters of a number of pitches is used (12 per octave in the case of musical signals) with all the filters logarithmically spaced. Indeed the Q-Transform can be seen as a logarithmic-spaced filter to which Fourier Transform is applied.
In the case of multi-pitch system, the evolution in time of the spectral content of a signal is really important to understand. Furthermore, the pitch being played is extracted from the frequency content, and the onset time of a note from the time information.

Among those analysis representation possibilities, a logarithmic-spaced spectrogram was chosen to overcome issues of STFT deriving from low-frequency pitch estimation and from the tuning of the instrument.
The next design decision concerns the choice over which method to employ. As seen from the overview on state-of-the-art techniques, signal-processing methods seem to be robust and inexpensive from a computational standpoint, however, being difficult to generalize due to the use of specific models [40]. Spectrogram factorization and sparse decomposition would be more general than signal-processing techniques, but, on the other side, are more computationally expensive and less robust [25]. Following the above-mentioned considerations, Machine Learning systems seem to meet the requirements needed for a multi-pitch estimation problem thanks to good generalization and robustness of the methods. The main drawback is represented by the computational expense of the networks. Neural Networks, among all the Machine Learning algorithms, are presented as the most promising method. Indeed, results of such systems, as the one proposed by Böck [34] or the Madmom library, and the studies done on different types of Neural Networks proposed by Marlot [40], guided us to this design decision.
The last consideration on the design of the system regards the way in which pitches are estimated. Despite joint multi-pitch estimators being more precise and less prone to errors, they are unfrequently employed due to the complexity of the problem. So, as for most of the Automatic Music Transcription algorithms, a frame-based method was applied to the multiple-F0 estimation phase.

## 4.2 Workflow

As anticipated in the introduction of this chapter, representation of data is one of the most important design choices. The representation indicates the information on which the method should work and what needs to be fed into the Neural Network. Despite some methods still usnig time representation, the time-frequency one is still the most popular for these kinds of applications following the reasons previously mentioned.
The signal-processing phase comes first in a music transcription system and allows the extrapolation of features from raw input data. The extrapolation can be seen as a different way of interpreting the data. As previously said, the choice of the features play a central role within an Automatic Music

Transcription system. Usually, methods for discarding redundant or non-useful information for the transcription are studied to avoid memory problems linked to the large amount of data the system needs to process.
Therefore, the signal-processing stage plays a key role in the economy of the system, however, the trade-off between performance and precision is still at stake.
During this phase, in addition to the audio analysis, all the information concerning the dataset are extracted from the MIDI files. All information, both derived from audio and MIDI files, is saved in compressed files that later in the this study will be loaded for a quicker re-utilization. Mainly two kinds of structures are saved for each file: the target and the features. The former are built reading MIDI files and converting it into a big matrix. Its dimensions are 88, which represent the number of pitches on a keyboard, and frames within the audio signal, the latter identifying time evolution. MIDI files are scanned in order to discover note events. Each note event is analyzed and then added to the matrix setting to one the value according to the pitch and time frame. The resulting matrix would be similar to a MIDI piano-roll using the number of frames for the time dimension. Throughout the process different targets may be selected depending on the experiment one needs to set, and these are either frame-by-frame, or onset or offset targets. Regarding the features, audio is analyzed through the support of the audio section of Madmom library [34] that allows easier calculation of the Logarithmic-Spectrum. Firstly, the Short-Time Fourier Transform is applied to the audio signal extrapolating its spectrogram using a window of size 1024, 2048 and 4096. To overcome problems linked to low pitch estimation, a logarithmic filter is applied to the spectrogram. Finally, the first order differential of the logarithmic spectrogram is extracted, helping the onset detection phase as it will be explained later in the section dedicated to signal-processing.
The data is saved in numpy structures to be loaded quickly from memory, and they contain names of the musical pieces, target, represented by the conversion of MIDI files into text ones, and features as explained above.
Another important block is represented by the Neural Network training. The pre-processed data is loaded from the memory and divided into three splits, one dedicated to the training (80%), the other to the validation of the training (10%) and the last to the test (10%) before being inserted into the Neural Network.
The Neural Network tries to approximate the distance of feature representa-

tion to the target one, mapping the difference between the target obtained by the results of the function applied to features. The function is modified thanks to parameters and each time they are updated to find the closest approximation, as being measured by the error calculated at each step. The error is calculated every epoch, with a maximum number of 10000, and a patience of five iterations. The Learning Rate is updated following the optimizer chosen within the training, while the output of the network will be represented by the parameters that best fit the approximation compared with the targets.
The post-processing phase is applied to the predictions using Madmom library. During this phase peak-picking function with a moving average and threshold parameters help the post-processing of the predictions. The detection derived from this last passage is compared to the annotation files during the evaluation.
Detections and annotations are compared event by event, where a True Positive value is detected when pitch is the same for annotation and detection files and the onset time of the detection is in a range of $\pm 50ms$ from the annotated one; those events from annotation non-detected are classified as False Negatives, while those non-annotated but detected events are False Positives.
Finally, the test phase of the network consists of applying the trained Neural Network to the features of the test set. The trained Neural Network is loaded with the parameters calculated from the training and validation data, and the output produces the predictions. The predictions are extracted from the features by the Network depending on the target fed into it. If, for example, the onset target is taken, the Network will search for a map between the input features and them.

## 4.3 Audio signal pre-processing

Audio signals are transformed through a pre-processing phase in a compressed musically meaningful representation. It can be used for the approximation and the analysis of the acoustic model during the training of the neural network. Three parallel Short-Time Fourier Transform are applied to the signal frame-wise applying three different lengths of Hamming window to the signal, respectively 1024, 2048 and 4096 frames. The resulting frame rate is 100 frames per second and each window is distant 10ms for a time-length

of 23.22ms, 46.44ms, and 96.88ms as suggested by the Böck's work [34].
The frequency range analyzed was 30Hz 17000Hz using a sampling frequency fixed to 44100Hz, as can be derived from the Shannon theorem. Indeed, the sampling frequency was set to be at least the double of the maximum frequency of the signal so as not to lose any information during the sampling. Furthermore, the linear magnitude spectrogram of the audio signal of each of the three STFT is filtered to compress the representation.
A Bark scale aligned frequency filter-bank with 12 and 24 bands per octaves was used aiming at improving the frequency resolution of the system. The logarithmic representation of the Bark spectrogram with 24 bands per octave improves all evaluation measures of about 10% due to its sharper frequency resolution. Indeed, 2 critical bands are used for the analysis of each semitone space. On the other hand, the 12 bands setup dedicates just one critical band per semitone. However, the semitone spacing aims at reducing the dimension for the feature vector and desensitizes the whole system against minor tuning variation.
Due to the percussive nature of the triggering of piano sound, during the attack phase, a steep rise in energy is therefore detected. This represents an important clue in the estimation of the starting point of the note, the onset. For this reason, first-order differences, meaning the energy differences between preceding frames are included in the vector features to better detect onsets. Delay windows are applied with an overlap of 0.5 and their length depends on the length of the STFT window and is of 2 frames for the 1024 STFT window, 4 and 8 for the 2048 and 4096 ones. The two feature vectors for each musical piece have size 482 for the 12 bands per octave, while 836 for the 24 bands per octave, nearly double.

## 4.4 Neural Network

Deep Neural Networks can be referred to as a powerful machine aiming at learning from a model that can be used for classification or regression tasks. A DNN is usually composed of one or more non-linear transformations, depending on the layers present in the network.

Each layer performs a transformation

$$h_{t+1} = f(W_l h_l + b_l)$$

where parameters $W_l$ and $b_l$ represent respectively the matrix of weights and the vectors of biases for the level l with $0 \leq l \leq L$. The function $h_{t+1}$ is the result of the non-linear function $f$ applied element-wise to the input $h_t$. The first layer of a network, called $h_0$, is represented by the input itself x, while the output of the network, $h_L$, is the result of all the transformation according to the layers. The output yields a posterior probability distribution $P(y|x, \theta)$, where $\theta = W_L, b_l^L$ which can be estimated with the backpropagation algorithm. A DNN for acoustic modelling can be set by introducing a frame of features as input. For example, feeding to the Neural Network a magnitude spectrogram of any frequency representation, the Neural Network will be trained to predict the probability of detecting a pitch in the time-frame t as $p(y_t|x_t)$.
The thesis exploits the Convolutional Neural Network type to preserve spatial structure of input, trained using the RMSprop optimization algorithm in batch mode.
The convolution operation $h_{j,k} = f(\sum_r (W_{r,j} x_{r+k-1} + b_j))$ produces a new feature map of the input applying shared weights across all input lengths within the convolutional layer. The input vector is selected on a region of $m\mathbf{X}n$ to which a max-pooling layer is applied to select the maximum within the region and the weights are represented as tensor of multiple dimensions. At the time t the time window is of $2k + 1$ length and the output posterior distribution is represented by $P(y_t|x_{t-k}^{t+k})$.
The Convolutional Network is able to perform with better precision by incorporating information from different time examples modelling a time context through the frame window.

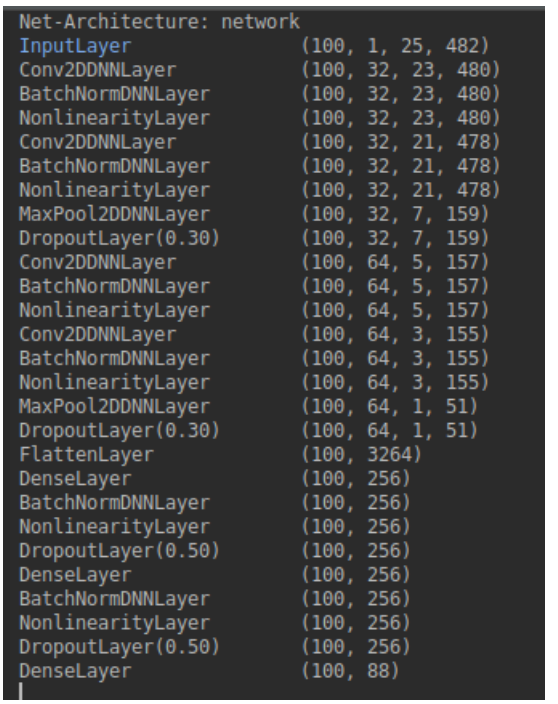


*Figure 4.2: Convolutional Neural Network list of layers used in the thesis*

The figure 4.2 shows the layered architecture of the Convolutional Neural Network for the 12 bands per octave, features vector and the dimension of each layer. It can be observed how the input layer is characterized by the entire number of features (482 for the 12 bands case), and that Convolutional layers are followed by Non-linear layers and Max-Pool layers. The Convolutional layers are built using two different building blocks: the first consists of two layers with 32 3x3 filters and the second consists of two layers with 64 3x3 filters. Both Convolutional layers are combined with batch normalization layers, and each followed by a 3x3 Max-pooling layer and a drop-out layer with the drop-out value $\lambda = 0.3$.
Most popular non-linear functions are the sigmoid one, the hyperbolic tangent and the rectified linear unit:

$$\sigma(a) = \begin{cases} \frac{1}{1+e^{-a}}, & \text{sigmoid} \\ \tanh a, & \text{hyperbolic tangent} \\ \max(0, a), & \text{ReLu} \end{cases}$$

The choice of a function over another one can potentially impact on the

effectiveness of the Neural Network as an approximator [26]. ReLu seems to behave better with a fast gradient convergence, but the initialization of training weights has a substantial impact on the choice of the non-linear function on the transcription [27] [35].
The Dropout prevents the co-adaptation of units increasing robustness to noise. As its results also present better generalization and mitigating overfitting of the network by setting to zero a fraction of the activation values of a hidden layer applied usually for each training case.
Batch Normalization produces activation with a zero-mean and unit-variance distribution for each layer to which it is applied. The normalization at each training set limits the distance between the activation distribution from the normalized one (zero-mean and unit-variance) [29]. The Dense layer can be seen as a dense matrix and vector (W,b) to which non-linear functions are applied. The Dense layer transforms the input through the nonlinearity mapping.
The Convolutional layer defines a number k of kernels $C_k$ to be applied to the weights and biases matrices $(W_c, B_c)_{c=0}^{c=C_k}$. Each input is transformed through the convolution on itself and the kernel in a different feature mapping, through applying also a non-linear function.
The MaxPool layer is used in Convolutional Networks to provide a small amount of translational invariance. It selects the maximum activation value within a restricted area of the input both in time and frequency, reflecting small changes in the tuning of the network.
Global Average Pool layers compute the mean value of the features maps.

## 4.5 Study of coefficients

The following paragraph is dedicated to the description of all the experiments and the studies performed in order to allow the system to work properly. One of the most important studies was made on SoundFonts and focused on the understanding of which SoundFont to employ during the synthesis of the dataset. A simple C major musical scale was synthesized and evaluated on a pre-trained network to understand which of the SoundFonts could be chosen for the following experiments. The results highlighted how SoundFonts rendering the playing of the electric organ and electric piano as unsuitable for the evaluation of network trained on the acoustic piano. This can be explained by considering the massive difference present in spectral components

of sound from different SoundFont classes.

| Piano Type | F-measure | Precision | Recall |
|---|---|---|---|
| Hammond B3 | 0 | 0 | 0 |
| Electric Grand U20 | 0 | 0 | 0 |
| Crazy Organ | 0 | 0 | 0 |
| FM piano | 0.75 | 1 | 0.6 |
| Hammond Organ | 0.7 | 0.68 | 0.73 |
| Electric Fender | 1 | 1 | 1 |

*Table 4.1: Evaluation metrics for SoundFonts tested on a C-major scale*

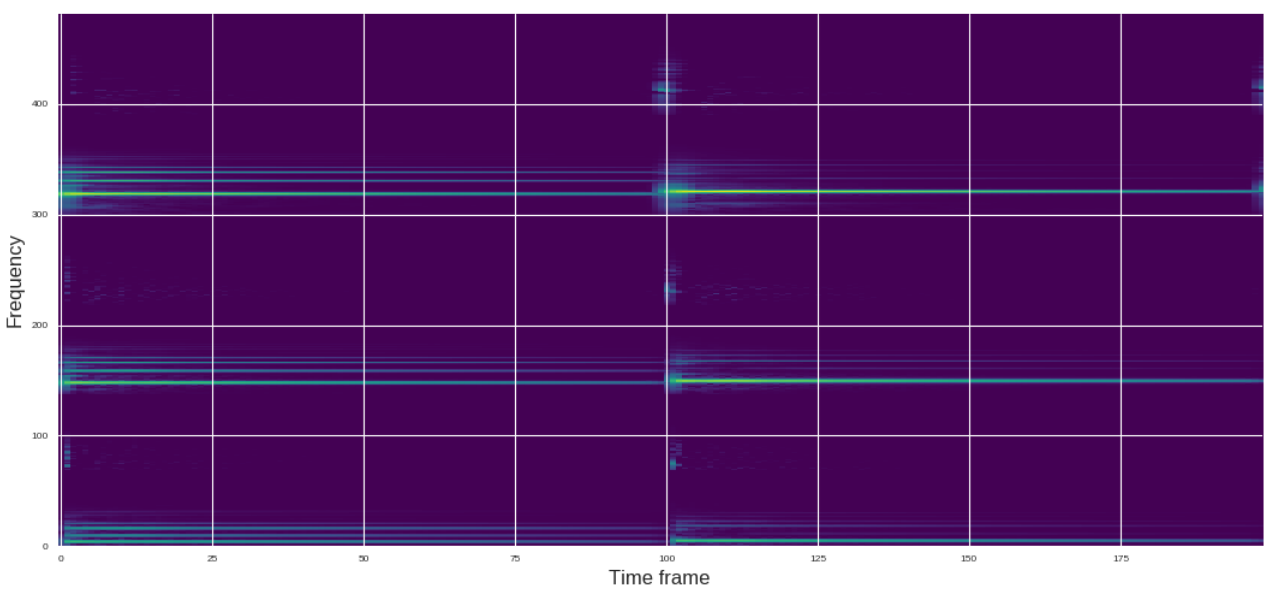


*Figure 4.3: Electric Piano Fender Spectrogram*

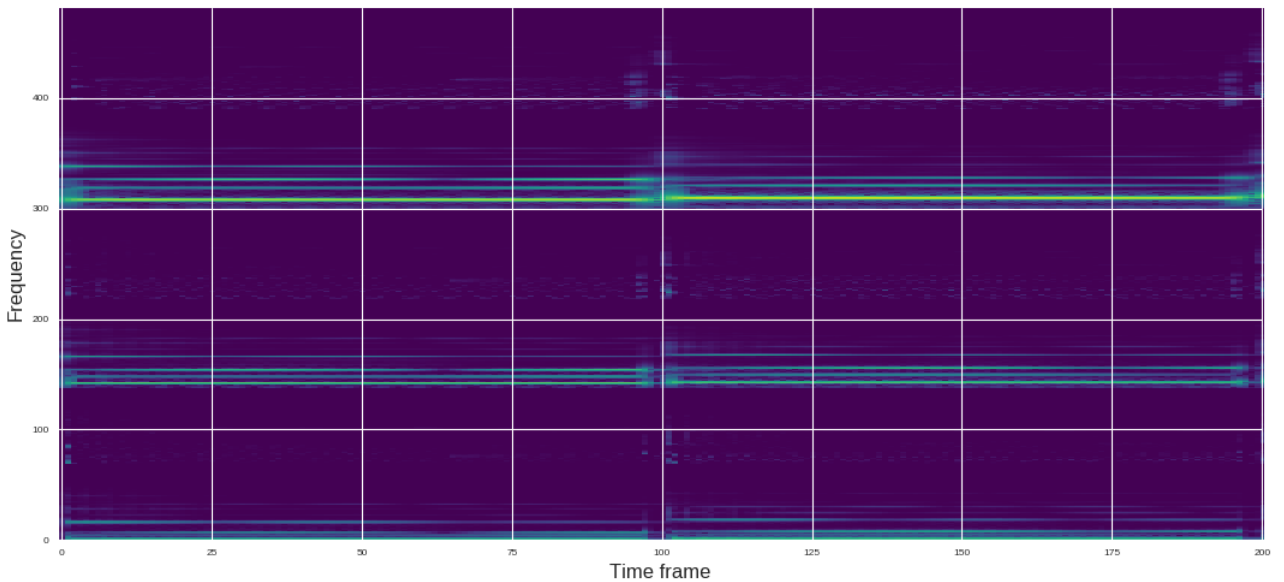


*Figure 4.4: Hammond B3 Spectrogram*

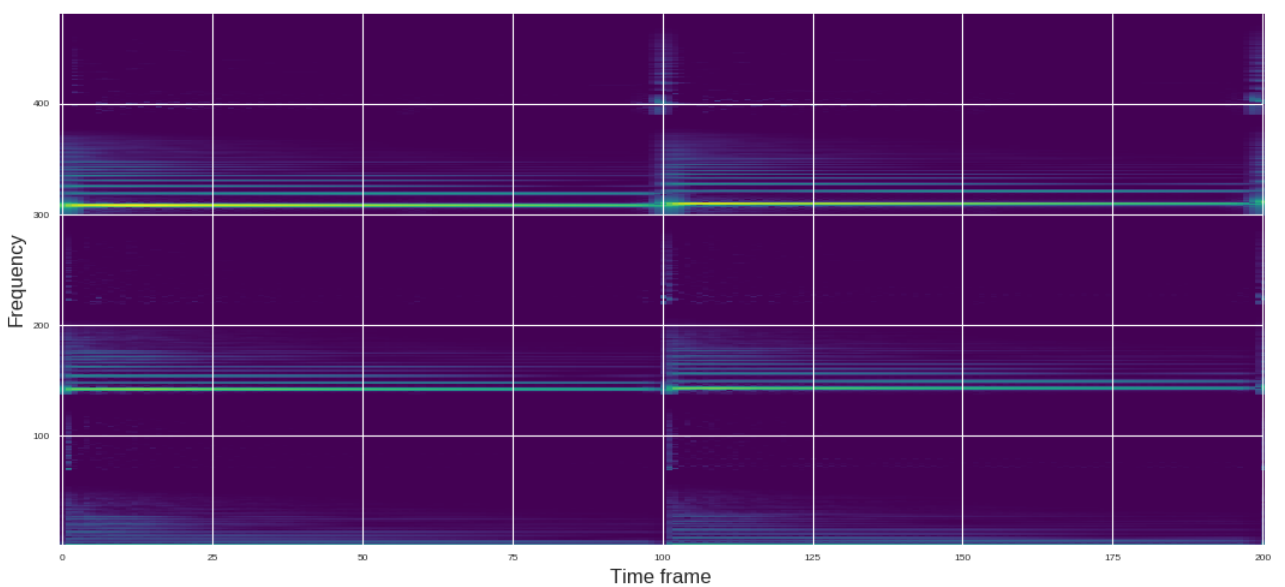


*Figure 4.5: Electric Grand U20 Spectrogram*

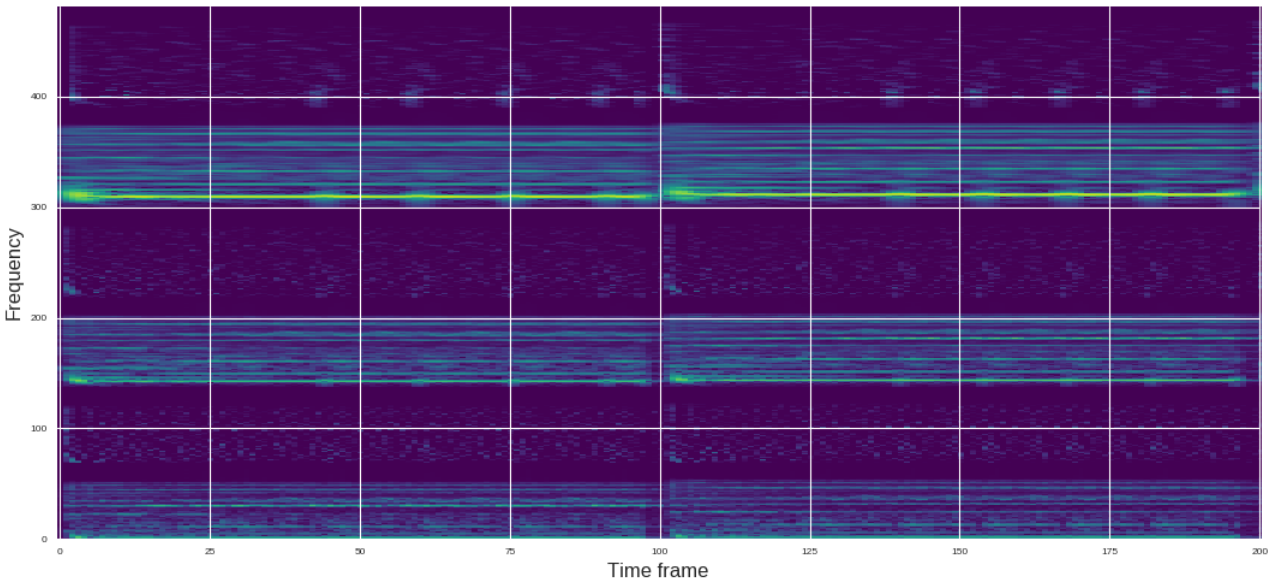


*Figure 4.6: Crazy Organ Spectrogram*

| Measurements | Electric Grand Piano | Fazioli Grand Bright Piano |
|---|---|---|
| F-measure 50ms | 0.96 | 0.96 |
| F-measure 25ms | 0.06 | 0.62 |
| Precision 50ms | 1 | 1 |
| Precision 25ms | 0.07 | 0.7 |
| Recall 50ms | 0.93 | 0.93 |
| Recall 25ms | 0.06 | 0.6 |

*Table 4.2: Evaluation metrics tested with different window size*

The results shown in the table 4.1 can be justified by the spectrogram of each SoundFont. Indeed, the ones presenting a high number of frequency components, usually, have the worse results. On the contrary, the ones with frequency peaks clearly recognizable reach a high value for metrics.
Other studies about the environment in which the instrument was recorded or is designed to render, such as reverberation or bright and clear timbre,

were briefly performed changing the window of evaluation during onset detection. The table 4.2 shows how doubling the window impacts on some values of SoundFonts increase. Furthermore spectrograms show a slight variation in the onset value compared to the previously presented SoundFonts.

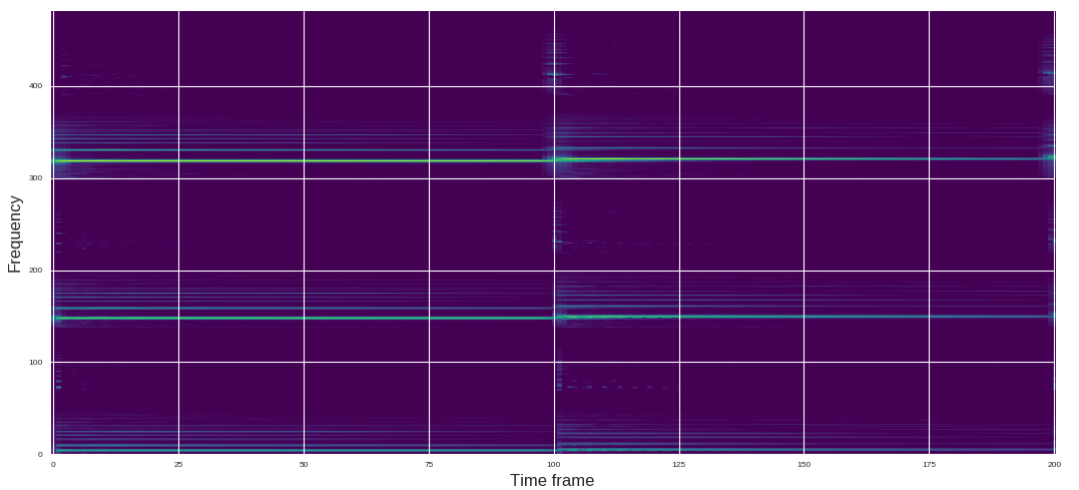


*Figure 4.7: Electric Grand Piano Spectrogram*

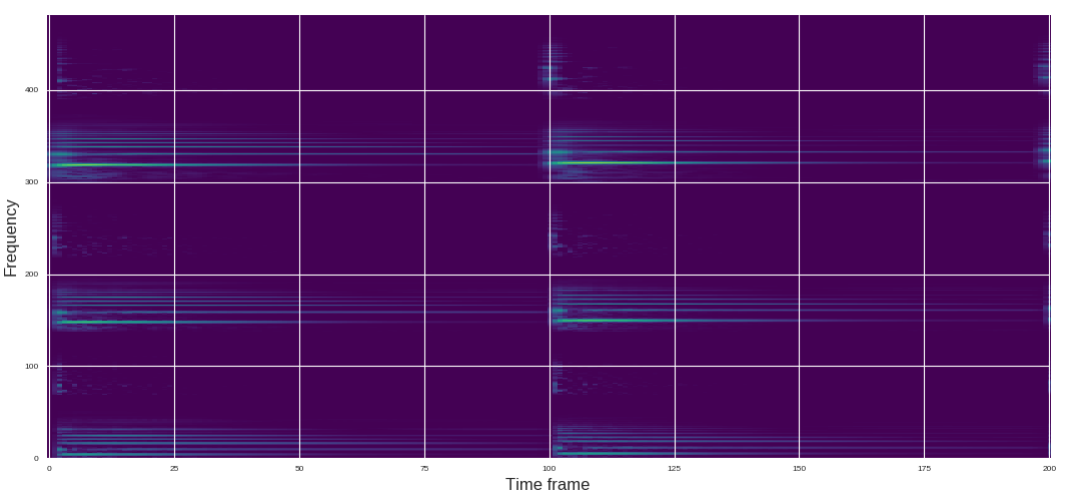


*Figure 4.8: Fazioli Grand Bright Piano Spectrogram*

Coefficients for the post-processing function that needs to be applied after the estimation of the predictions were studied depending on a piano note time envelope. The latter is usually characterized by an accentuate energy burst during the attack phase due to the percussive nature of the sounding mechanism.
Meaningful prediction values from the network have been shown to be in the range of 0.2 to 0.9. In order to exclude non-meaningful interferences caused by octave errors the threshold for the peak-picking function was set to 0.25. The time window employed in the evaluation of MIREX context is 90ms. Although onset detections in the presented work and the Böck's one [34] were evaluated with a shorter window size (50ms), however, results are really sharp on onset time detection as will be explained in the evaluation section. Using a window size greater than 50ms, the evaluation system gains in performances. This is proved by the studies on onset tolerance time interval in Böck dissertation [34], highlighting how his system retrieves satisfying results even within a shorter time window (25ms).

# Chapter 5

# A new dataset for jazz piano transcription

No more appropriate quote is worth mentioning than Bob Mercer's one: "There is no data like more data". It clearly represents the most recent trend of implying an analysis technique which relies on a statistical approach, using a wide variety of files. The trend raises the need to deal with a large amount of data to have enough information to work on.
Indeed, statistical analysis may help resolve many problems related to Music Information Retrieval, but is raising another problem linked to musical data collections which is the licensing of the audio files.
The attempt to create a dataset focused on the jazz music genre is justified by the almost complete absence of a dedicated dataset and by the previously mentioned musical interest in jazz music. Indeed, jazz is characterized by wide a variety of different performing styles and also by a wide employment of improvisation. The latter usually produces a countless number of variations and variants linked to a personal fingerprint. The intrinsic variety that identifies the genre itself is one of the core components that makes its analysis relevant. Furthermore, it can be referred to as a benchmark for transcription tasks which ease the musicological analysis of jazz.
The main purpose of the proposed dataset is to provide a collection of relevant musical data which will provide a contribution to jazz music analysis related to the reasons mentioned above. The dataset will comprise the complete set of data needed for evaluation of the system: the set comprehends audio data and reference data as annotations. The main issues for the creation of a musical database is the collection of audio signals. Problems concerning

licensing and available quality of digital data may arise as well. Furthermore, the annotation building stage is one of the most time consuming and imprecise processes if performed manually. Those last considerations guided us to the choice of taking MIDI file as input to create reliable annotations and high-quality audio data. The discussion, then, will focus on the used technology and workflow which follows the creation of a jazz dataset: the process starts with data collection while finishing with the creation of a complete database. The main macro phases of the creation are the following: firstly, collection and analysis phase, secondly, a separation of piano and non-piano instrument stage, thirdly a synthesis and annotation production one, lastly a mixing stage. Finally, a brief estimation test was done with the help of Madmom library.

## 5.1 State-of-the-art Datasets

The increasing interest in Music Information Retrieval tasks raises the need for a musical data collection to be solved through statistical analysis of data. Furthermore, music licensing is a contemporary problematic, since it represents the licensed use of copyright music and it is intended to ensure the protection of the owner's work [32].
Musical annotated dataset can be focused on different problems, so it can have different structures. The main problematic that this dissertation has confronted refers to fundamental frequency extraction as well as onset detection. For the sake of completeness, other datasets will be presented.
The Structural Analysis of Large Amount of Musical Information project presented by Smith et al. [31] SALAMI is a dataset focused on structural annotations containing 2400 different types from 1400 musical recordings of a wide variety of music. The mentioned project attempts to balance all the genres from jazz to classical, with a good number of non-Western music, outstandingly rare. Structure refers to the partitioning of a musical piece into sections, grouping similar or repeated segments. This dataset could deliver great contributions to music theorists and musicologists due to its focus on the structural organization of music. Algorithms for automatic production of structural description represent an active area operating in this manner [31] and the SALAMI project tries to exploit their potential. However, their test demands the creation of human-annotated ground-truth dataset. Finally, Smith et al.'s project can be applied to a variety of different studies on music

perception, formal styles and musical parameters according to the artist or genre.
The MedleyDB [33], instead, is a dataset developed either for the melody extraction, sound source separation or automatic mixing. In particular, during automatic instrument recognition, annotation of instrument activations are exploited. It consists of 122 songs of non-specific musical genre, 108 including melody annotation.
Other datasets have focused on MIR task as automatic music tagging or music recommendation or artist recognition. One of those is the Million Song Dataset built by Ellis et al. [21]. It contains a collection of audio features and metadata for popular music tracks for a total of 280 GB data and a million song files with more than 44000 unique artists.
Three main reasons have directed most of the datasets and transcription methods towards the choice of focusing on piano: firstly, wide availability of digital audio; secondly, availability of score annotation mostly in classical music, and lastly, wide knowledge linked to piano source production. Within the wide variety of genres the classic one is extensively used due to the availability of musical scores that represent a checkpoint for annotations derived from MIDI.
The only dataset with different classified genres is Real World Computing one [43], which collects 315 musical pieces from pop, jazz as well as classical. It was built thanks to the powerful collaboration of the RWC Music Database Sub-Working Group and the RWC Partnership of Japan. The performances were recorded to obtain audio files and MIDI files. A dedicated set for individual instruments is available and contains variations of playing styles, dynamics and different instrument manufacturers.
Other databases, with higher fruibility than RWC due to their public availability, are focused on classical music since the wide collection of digital data and annotations appertaining to that genre. One of the most important is the MIDI Aligned Piano Sound [3], employed also for the evaluation of this work. MAPS dataset is divided into four-set each one containing audio files, annotation, and MIDI. ISOL set is dedicated to isolated monophonic notes, RAND contains chords created randomly, UCHO focused on usual chords recurrent in Western music, while the MUS set is made up of piano music pieces. The MAPS dataset provides a large amount of sound at different levels of structure, from isolated notes to complete melody, on high-quality recordings, 16-bit sampled at 44100 Hz. The latter are generated either

thanks to the help of automatic generating processes from the synthesis of MIDI files or to the use of MIDIfied piano like the Disklavier [3].
Other used datasets are LabROSA, Mozart By Batik and MIDI from the midi page. All of them are nearly totally related to classical music also following the availability of musical scores that permit the check of the produced MIDI annotations.
Starting from the Laboratory for the Recognition and Organization of Speech and Audio (LabROSA), it is explicitly built for music transcription, classification and similarity estimation.
On the other hand, Mozart dataset collects 13 sonatas from Mozart played by Batick on a Boësendorfer Midified piano. More than one annotation format is available, and they are derived either from the MIDI match to score alignment or from the conversion of score files '.scr' to MIDI. Furthermore, different audio files, in addition to the Boësendorfer played, are available in the form of synthesized MIDI. Finally, MIDI Maestro dataset is available for the test. MIDI files from the http://www.piano-midi.de/ page are synthesized with GrandConcertMaestro SoundFonts, which collects high-quality SoundFonts. The page offers a total of 267 MIDI files from classical music including artists like Bach, Beethoven, Chopin, Tchaikovsky, and annotations are extracted from them.

## 5.2 Design choices and considerations

The selection of the data has been the first decision to address. As explained in the scope of the thesis, one of the aims of this work is to build up a complete jazz music database. The focus of the presented dataset and the transcription method was moved towards the jazz genre for the motivations mentioned above. A first phase of research within the literature of piano transcription method was needed to detect those works already focusing on the jazz transcription and then understand which data they are working on. The research has pointed out that most of the piano transcription methods are focusing on classical music due to the availability of datasets focused on that kind of music. For this reason it has been important to understand how to properly build up a reliable and precise dataset.
A standard way to build up a musical database complete with annotation is to produce the ground truth a-posteriori from the audio signals. This process, besides being time-consuming, usually produces a considerable level of

inaccuracy. The problem can be approached either in a manual way or semi-automatic way, however it can still be affected by erroneous values for the annotated parameter. Unlike the large part of dataset creation approaches, MAPS dataset was created from MIDI files, constituting the core of annotations. Audio files are synthesized through an automatic process from MIDI files.
Aligning with the MAPS approach, this work wants to offer reliable annotation extracted from MIDI files, and for this reason, they were chosen as the building block of the database. The source from which the files were taken was the MIDKAR website (http://midkar.com/) that was created with no professional or commercial intent. It contains about 8000 MIDI files divided into musical genres. The Jazz split from the MIDKAR page was analyzed to detect piano solos and to check the correctness of the MIDI. Due to the low number of piano solos, we opted for the file with the accompaniment part. It followed that a proper processing able to separate the piano from the accompaniment part had to be developed. Since MIDI files can be built in different ways with regards to the synchronization of the tracks, to simplify the automated synthesizer system, just MIDI files of type 1 were taken as input of the process.
The final dataset contains the related audio signal for each musical piece synthesized from its MIDI file, and the text annotation describing the musical events present in the piece (time of onset of the note and pitch of the note are marked). A complete set containing this information needs to be given to the transcription system as input: the audio for the time-frequency representation extraction, the MIDI one in order to build the ground truth for the algorithm and finally the annotation one to evaluate the system.
The use of MIDI files as a main building block of the dataset leads to the forced choice of the employment of SoundFonts for the sythesis of the MIDI. Besides a robust and reliable extrapolation of annotation, MIDI and SoundFonts technologies are lightweight and allow the researcher to easily modify the dataset to different study cases, changing the organization and the SoundFonts.

## 5.3 Technologies used

This section is dedicated to the description of the main technologies employed during the dataset creation process. If MIDI standard was deeply discussed before and also Madmom library was introduced in preceding sections, it is therefore necessary to explain the rationales behind the Mido python library, TiMidity, FFmpeg software and SoundFonts file type. However, this section will focus on the last three and not on the Mido library because of the straightforward nature of it. The Mido python library [44] was employed for the analysis of the MIDI files, while a combination of the two software and SoundFonts technology was exploited for the synthesis of MIDI files and mixing.

### 5.3.1 Timidity++

TiMidity (http://timidity.sourceforge.net/) is a free software synthesizer, distributed under GNU general public license, that runs under Linux OS. TiMidity can read different types of data in addition to MIDI (.mid SMF is Standard Midi Files), recomposer files, MFI (Made for iPod is a licensing program for developing hardware and software in Apple devices) and module files (MOD music, tracker music). This type of file stores digitally recorded samples and pattern of music data such as a spreadsheet containing the number of the notes, the instrument, and the controller message.
Another important feature that has been exploited in this thesis project is the support of SoundFonts type files that helped render the synthesized MIDI sound.
The main feature of TiMidity can be referred to as the ability of MIDI files to play without a hardware synthesizer and the conversion of them into PCM waveform data. MIDI instruments are substituted by type of files such as Gravis Ultrasound compatible patch (Gravis Ultrasound is a soundcard compatible with IBM known in the early '90 for the good reproduction quality of MIDI files) or SoundFonts.
Supported audio file types are identified by the extensions .wav .au .aiff. TiMidity can also display much useful information about the playing file, regarding sound spectrogram of a playing music piece and track information of a MIDI file.

### 5.3.2 SoundFonts

SoundFonts refer to a technology used to play MIDI files using sample-based synthesis. The first version was developed in the early '90 by E-mu System and creative Labs without any specification. It was used by the soundcard Sound Blaster AWE (Creative Lab), then upgraded to the known 2.0 version. SoundFont 2.0 redefines the representation of audio data using additive unit of real-world blocks, creating layers of an instrument as well as adding stereo samples. From the introduction of the 2.1 version which adds some technical specification, it has enabled the configuration of SoundFonts within the MIDI controller. Since MIDI do not contain any audio files, they represent just an annotation of instruction to be reproduced by synthesizer. Synthesizers then make use of the wavetable, that is sampled sound, to reproduce all the instructions contained in the MIDI file. SoundFont standard wants to provide a portable and universal interchange format for wavetable synthesizer samples. Thanks to the use of generator and modulators in addition to the special unit, it can be easily extended and portable using hardware independent parameters supporting a wide range of technologies.
A SoundFonts file contains samples in PCM waveform (like WAV format). Those samples are then mapped in sections within the octave interval and with loops using musical effects like vibratos or velocity change. Since this format is widely applied to MIDI, it usually contains 127 instruments and a dedicated track for percussion and sound effects.

### 5.3.3 FF-mpeg

FF-mpeg (https://www.ffmpeg.org/) is a software suitable for recording, conversion and reproducing of file audio and video. Based on the library libavcodec, it permits not only encoding and decoding but can also transcode, mux and demux, streaming, filtering and playing of any audio or video format. Originally developed for Linux OS and then extended also to other OS it contains three main tools: a command line tool for converting multimedia files between formats called ff-mpeg, a simple media player called ff-play and finally a multimedia stream analyzer called ff-probe.
The main instrument used within the dissertation is ff-mpeg that reads a number of input files of different characteristics. Inputs can be either regular audio files or network streams or grabbing devices and write to an arbitrary number of outputs. Ff-mpeg calls libavformat library (for demuxers) to read

input files and get packets containing the encoded data.When multiple inputs are used simultaneously, the tools' attempts to keep them synchronized. Encoded packets are fed to the decoder to get the uncompressed frames (raw video or PCM audio) that can be processed by filtering and finally re-encoding before the muxing of them to get the final output file. All those passages need the use of the specific library contained in ffmpeg. Libavcodec library contains encoders and decoders for both audio and video codecs and it is used in the middle part before applying any filtering to data. Libavformat is also very used in the first part of the handling of a file for demuxers and muxers of multimedia container formats. Then the core of filtering is libavfilter containing the media filters. Finally, libswresample performs highly optimized audio resampling and a sample format conversion operation.

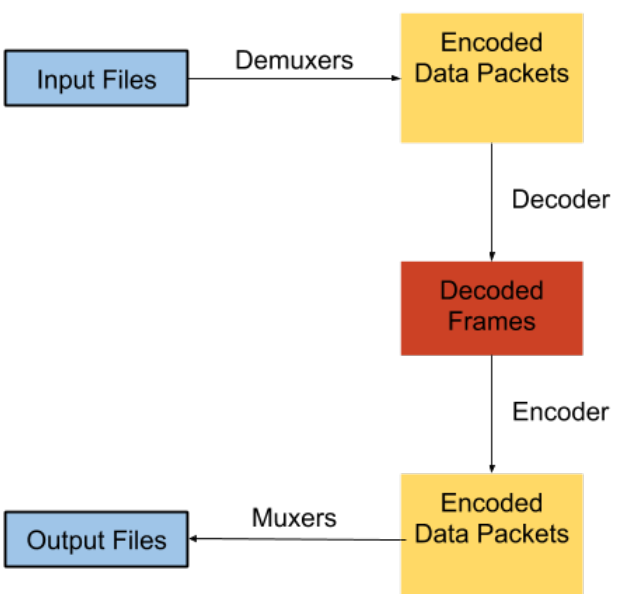


*Figure 5.1: FF-mpeg operational scheme*

## 5.4 Workflow

The workflow of the creation of the musical database is straightforward and can be categorized into four main phases: input data collection and refinement, track division in piano and non-piano musical pieces, annotation production and synthesis of piano and non-piano MIDI files, mixing of synthesized audio tracks. All stages needed for dataset creation were preceded by a soundfont collection one.

### 5.4.1 Soundfont collection and organization

The proposed dataset can be exploited by researchers in many ways and leaves them free of choosing which kind of data to work on. The employment of MIDI and SoundFonts technologies has a great potential for the mobility of the dataset in different study cases. For this reason, the collection of SoundFonts was an important pre-step to define how to build the dataset and how to conduct different experiments.
SoundFonts were downloaded from different sources, but the main one was the Merlin site (http://www.SoundFonts.gonet.biz/). They were split into the same number for the three different sets, Acoustic, Electric, and Electric Organ. For the accompaniment case, just a full orchestra file per each set was available on the site. As a matter of fact, the building of more SoundFonts for an entire set of instruments would have been a time-consuming procedure. Moreover, the latter is not discussed in this paper. Concerning the piano sound source files, each set contains 12 of them dedicated to the piano type indicated above. Most of them were downloaded from the Merlin web site, also Fazioli and Electric pianos, but the organs were retrieved from a different source.
The above-mentioned subdivision would lead to a three-cross validation experiment for the dataset, maintaining a good variance within each set.

### 5.4.2 MIDI collection and refinement

The Dataset creation process starts with collecting input data. As anticipated in the section dedicated to the design choices 5.2, the musical database was built from MIDI files in order to derive reliable annotation, avoiding errors derived from the manual annotation process. MIDI files were downloaded from the MIDKAR web page (http://midkar.com/), which provides a classification regarding the genre division. This last feature is of real help to the intent of selecting music belonging to the jazz genre. Due to the low number of piano solo tracks available among the downloaded files, more than 650, it was decided to maintain both solos and complete musical pieces. The latter has raised the necessity of introducing an additional step within track divisions in order to produce MIDI files with just piano playing.
A refinement phase was added to check the nature of input to simplify the automatic process of tracks separation and synthesis. The analysis of MIDI files was performed with the help of MIDO python library [44]. It aims at checking the main characteristics of MIDI files concerning the presence of the piano, the format of the files and consistency of message sequence.
First of all, musical pieces need to have at least a piano part, MIDI files were inspected to understand if a piano is played and, in the meantime, statistics about which kind of piano was played among the available ones that were extracted. Exploiting the structure of MIDI files, inspection of the instrument is connected to the program number indicated by program_change messages. Programs dedicated to the piano are from 1 to 8 (remembering that in the informatics field the start number is 0 the interval goes from 0 to 7) comprehending in increasing order Acoustic Grand Piano, Bright Acoustic Piano, Electric Grand Piano, Honky-tonk Piano, Electric Piano 1, Electric Piano 2, Harpsichord, Clavinet.
The format type of MIDI is an additional important parameter to check. It indicates how tracks are organized inside the file. Indeed, as explained in the dedicated section 3.3, format can be of three types. To simplify the tempo analysis and to have a fixed structure, we considered just Format MIDI file 1, that is also the most used of the three types.
Finally, an inspection of the MIDI messages composing the files was made in order to avoid problems linked to the inconsistent building of the data. In fact, MIDI from MIDKAR are not official files and are created by amateurs and can have a structure not properly defined with MIDI messages

erroneous or inconsistent. All files with multiple program_change messages within the same track, meaning that in the same track more than one instrument is playing simultaneously, were discarded to avoid problems concerning the synthesis and the separation of the tracks.
To summarize the MIDI input must be a file format type 1, and must have at least one program piano playing, but cannot have multiple program-change messages within the same track. From the more 650 files collected from MIDKAR, about 100 were discarded finishing with a set of 550 MIDI files.

### 5.4.3 MIDI separation

After the refinement, MIDI division phase can be performed. The MIDI program analysis shows a massive quantity of piano tracks using the Acoustic Grand Piano, while a minimal part of them were performed on the Electric Grand, Electric 2 and Honky-tonk. The set containing Harpsichord and Clavinet is nearly non-existent, as expected from a jazz genre musical piece, while for the Bright Acoustic piano and Electric 1 piano the number increases to almost 50 files.

| **Piano type** | **Number of tracks** |
|---|---|
| Acoustic Grand Piano | 304 |
| Bright Acoustic Piano | 31 |
| Electric Grand Piano | 2 |
| Honky-tonk Piano | 8 |
| Electric Piano 1 | 64 |
| Electric Piano 2 | 16 |
| Harpsichord | 1 |
| Clavinet | 5 |
| Multiple piano | 119 |

*Table 5.1: Distribution of piano program*

Since the distribution of piano programs did not help and considering the most used piano in the jazz genre, it was decided to divide the whole set into three subsets each one of about 185 dedicated to a different type of piano: one to the Acoustic, one to the Electric, one to the Electric organ, like the Hammond organ widely used in jazz music. The idea of dividing into three subsets comes from the need to know on which kind of piano the system is working. It opens also an option for a three-set cross-evaluation and it can be also used to understand how well the system reacts during the evaluation varying the type of piano that is playing.
To make effective this division and to produce real piano solos, MIDI programs for piano were aligned depending on the split the files belonged to. The tracks in case of multiple piano programs were merged inside the same track. The dataset results have three separate groups; each of them focused on one of the three types of chosen piano. In the Acoustic piano set, all the piano tracks play on the program 0 (informatic notation), in the Electric piano on the program 4 and in the Electric Organ they perform on the piano program 7. This simplification was made also to accelerate the splitting and synthesis processes.
The splitting phase, as a matter of fact, consists of discerning which MIDI tracks have piano programs playing and which have not. Piano solos and accompaniment MIDI files are created from the original one. Despite the modification of the piano program in MIDI, controlling the program while playing takes into account all the eight programs dedicated to the piano, so as to be more generalizable in the future during the splitting of the dataset. At the end of this phase, the dataset consists of MIDI files divided into piano, accompaniment and mixed set, and each of these set is split into groups dedicated to specific piano type, Acoustic, Electric, Organ.

### 5.4.4 Annotation and audio files production

The next step, still derived from the MIDI analysis, is the production of both text and beat annotations. Text annotations are built as a succession of midi note events specifying onset time of the note and pitch. While beat files group tempo information extracted from the tempo track, useful for the analysis as beat tracking and down-beat tracking. Indeed, in MIDI format 1, the first track is dedicated to temporal related information, and for that reason is called also Tempo Track. All data extracted from that specific track are saved in a '.beats' file. All the other tracks are dedicated to playing instruments, with a specific case for drums usually having a special playing track set to the $9^{th}$ track. The analysis for the annotation building is focused on program_change and note_on messages. The former indicates the program number of the instrument and the track number on which the instrument needs to be played. The latter specifies the note and the reference track on which it needs to be played, indicating pitch number, onset time and velocity, linked to its degree of loudness.
The overall information about notes is collected in the annotations files recognized by '.piano' suffix for piano set, '.acc' for the accompaniment one and '.mixed' for the mixed one.
Thanks to the use of MIDI files, the annotation creation process is more reliable than a manual procedure, as it is extrapolated directly from the MIDI files. Furthermore, the employment of MIDI allows portable and mutable ways of producing high quality audio signal.
With the help of TiMidity software (http://timidity.sourceforge.net/), a free software synthesizer, it has been possible to derive audio for the piano and accompaniment folders. The mixed set that will be derived from the mixing of the already synthesized accompaniment and piano audio files. For this reason, during the synthesis procedure all possible initial periods of silence must be rendered, otherwise, the synchronization of the two tracks would be lost during the mixing phase.
Software option were set to synthesize monaural audio files and they were saved in '.wav' format rendered with personalized SoundFonts to achieve a good variance within a single set.
The synthesis phase plays a central role in the setting of the experiment since one of the aims of the work was seeing how different types of the same instrument affect the reliability of the transcription method and its perfor-

mance in cross-validation evaluation. As anticipated in section 4.5, some of the SoundFonts were discarded due to a really negative influence on the system caused mostly by the frequency content and their quality. However, the adopted synthesis approach can be easily modified and leaves open new settings for future experiments just changing the SoundFonts.

### 5.4.5 Mixing

The last phase of the workflow is the mixing one, where accompaniment and piano audio files are mixed together to form the originally merged musical signals. FF-mpeg (https://www.ffmpeg.org/) is the chosen software for that. Among the wide variety of functionalities offered by the software, the ffmpeg one was the only used within the work. The output resulted in a single mix of the two parts, accompaniment and piano.
At the end of the procedure the Dataset will contain, as explained in the figure, two main types of file the audio and the annotation one. Within the annotation are contained the MIDI files, classical text annotations and the beat ones. Furthermore, as mentioned above, each of these file types is divided into piano, accompaniment and mix sets, which in turn are themselves split into acoustic, electric, and organ sub-sets respectively indicated as split0 split1 and split2.

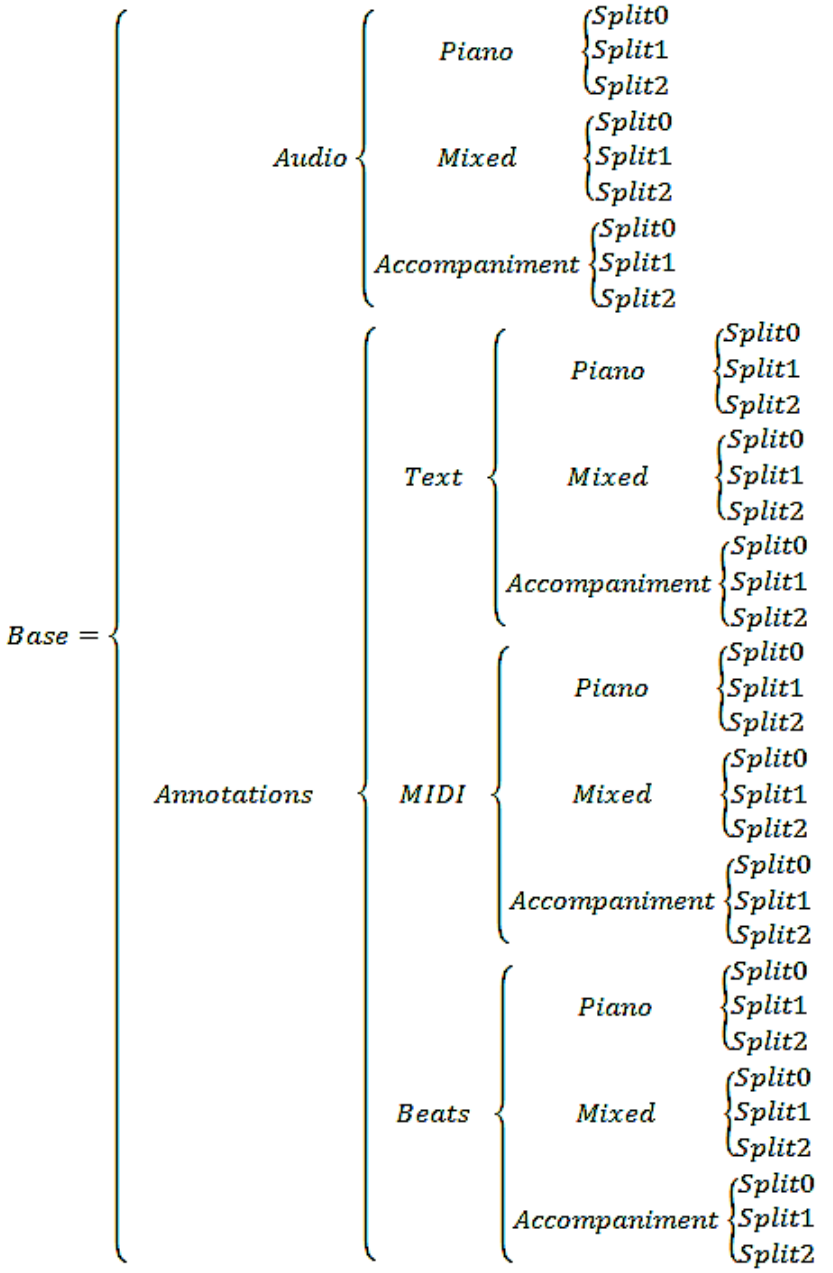


*Figure 5.2: Database scheme*

## 5.5 Transcription madmom

Madmom library was explained before in a dedicated section. It consists of a library tailored to the features extraction of different levels with the help of Machine Learning algorithms. The latter features allow the library to integrate state-of-the-art techniques to solve Music Information Retrieval tasks such as onset detection and music transcription without relying on third-party modules. Machine Learning part exploits pre-trained algorithms.
The employment of Madmom as a complete processing system is justified by the latter feature and the use of processor objects. Indeed, what was done with the dataset was a pre-evaluation of the data, applying transcription methods proposed by Böck in his work [34] on the audio files. The piano transcriptor relies on the algorithm based on a Recurrent Neural Network analyzing a Short-Time Fourier Transform taken with 2048 and 8192 windows. The STFT output is then filtered with a semitone filter-bank in order to compress the representation of data. As described in the section 3.4, the Network is built with bidirectional Long Short-Term Memory cells in order to increase the temporal context modelled by the algorithm. The output of the transcription results in annotated files contains onset time, pitch of the note and velocity, as in the dataset text annotation.
The detections computed from the Madmom program are then evaluated against the annotations extracted from the MIDI files and the table 5.2 summarizes all the results obtained. They show how different splits vary performances depending on the piano type chosen for the synthesis process. Indeed, the best performing split is the first one dedicated to an Acoustic piano; the Electric piano split has results comparable to the Acoustic one; while results for the Electric Organ split deteriorate a lot, decreasing by about 30-35%.

| | Total notes | True Positives | False Positives | False Negatives | F-measure | Precision | Recall |
|---|---|---|---|---|---|---|---|
| Split 0 | 259872 | 120281 | 85831 | 139591 | 0.516 | 0.584 | 0.463 |
| Split 1 | 205251 | 89907 | 81875 | 115344 | 0.477 | 0.523 | 0.313 |
| Split 2 | 262042 | 35327 | 125044 | 226715 | 0.167 | 0.220 | 0.135 |

*Table 5.2: Madmom results for the three splits*

# Chapter 6

# Evaluation

The following sections are dedicated to the description of metrics for each type of evaluation that can be performed. Among those, the frame-based and note-based evaluation methods will be explained, including metrics used in MIREX context. Furthermore, the experiments applied to the network is explained in all details presenting the results and a comparison made with state-of-the-art approaches.

## 6.1 Evaluation metrics

Evaluation of Automatic Music Transcription systems can be performed following a frame-based approach or a note-based one.
Despite the conceptual differences, the metrics for the evaluation of the two methods are the same. What changes is just the way of detecting True Positives, False Positives, False Negatives. Respectively True Positive indicates the number of correct detections; False Positive the number of redundant detections; Flase Negative the number of missing detections. The formal description of metrics follows the frame-based notation, including the specification for the $n^th$ frame. Note-based notation refers to a note event comparison, instead of a frame-by-frame one. However, they maintain the same formulation also for the note-based approach.
A common metric for the overall accuracy was defined by Dixon [10] as:

$$Acc_1 = \frac{\sum_n (N_{tp})[n]}{\sum_n (N_{fp}[n] + N_{fn}[n] + N_{tp}[n])}.$$

In the MIREX competition [36] a variant of $Acc_1$ measure considering just one octave is employed, and it is called Chroma Accuracy. Other accuracy metrics were proposed to focus on the number of pitch substitutions

$$Acc_2 = \frac{\sum_n (N_{ref}[n] - N_{fn}[n] - N_{fp}[n] + N_{subs}[n])}{\sum_n (N_{ref}[n])}$$

with $N_{ref}$ representing the number of pitches in the ground-truth at frame n. The number of substitutions is given by $N_{subs}[n] = min(N_{fn}[n], N_{fp}[n])$. Equally important for the evaluation of transcription system are precision, recall, and f-measure metrics defined as:

$$\textbf{Precision} = \frac{\sum_n (N_{tp}[n])}{\sum_n (N_{tp}[n] + N_{fp}[n])} \qquad \textbf{Recall} = \frac{\sum_n N_{tp}[n]}{\sum_n (N_{tp}[n] + N_{fn}[n])}$$

$$\textbf{F-measure} = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision}$$

Frame-based evaluation compares the prediction extracted from the transcription method to the ground-truth in a frame by frame fashion. The step between the frames, specified in the MIREX competition for the this kind of evaluation, is 10ms.
Note-based evaluation, according to MIREX specifics, takes into account each note event. The correctness of the prediction depends on the tolerance interval. Indeed, a predicted note event is evaluated as correct if its onset falls within a range of $\pm 50ms$ compared to the ground-truth onset, and the detected pitch needs to be between a quarter of tone of the ground-truth one meaning a $\pm 3\%$ of the fundamental frequency. The 6% indicates the semitone interval, while 12% an entire tone.
In the case of note-based approach, there is the possibility of taking into account the duration of the note. The offset of the note is the only parameter that can be employed for duration evaluation. This kind of evaluation refers, as the note onset evaluation explained above, to a tolerance interval. From onset detection, the offset one is the only difference in the pitch window. Indeed, the latter is of $\pm 20\%$ from the ground-truth frequency value, against the $\pm 3\%$ of the onset. The offset evaluation, on the opposite, respects the same time boundaries of the onset one.

In the end note events are counted as True Positive if it falls into the above-mentioned tolerance interval. False Positive, as for frame-based approach indicates the number of note events not present in the ground-truth. Finally, a note event is classified as a Flase Negative if it is not detected.

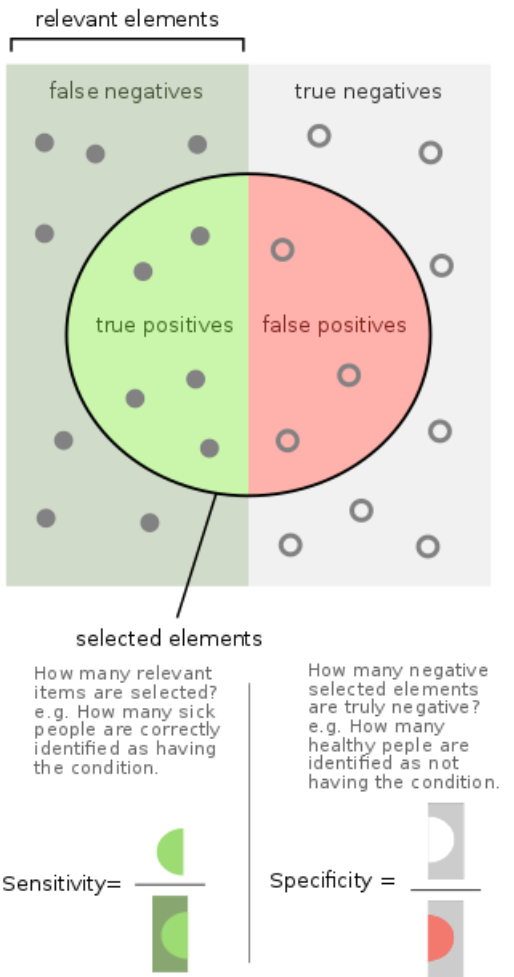


*Figure 6.1: Sensitivity and Specificity metrics*

## 6.2 Results

The evaluation of the proposed system will be performed following a note-based onset approach since the majority of the transcription system is note-based. However, the system was built to be tested with other approaches, both frame-based and note-based onset and offset.
Among the available musical databases, MAPS, MIDI Maestro, LabROSA and Batik, 30 pieces of MAPS instrument sub-set for the training of the Neural Network were chosen to conduct the experiment. With a greater number of musical pieces, problems concerning the time involved and the physical resource allocation raised during the training of the network. Those kinds of problems affected the results of networks trained on a great number of tracks, decreasing measurements by 30%. The subset was divided into the training part containing 80% of the whole set, and validation and test parts, both 10% as in Emiya et al. work [32].
Experiments on both feature representation features settings, 12 bands per octave and 24, were performed, with sensible improvements increasing the number of bands and then the frequency accuracy. The doubling of the number of bands resulted in a nearly doubled features vector, but only a time increment of the 30%. All the experiments on the 24 bands setting took about 35 epochs for an overall time of 30 minutes for the training and half a minute for the validation phase.
The results are not the same as the state-of-the-art approaches, but the proposed system can be improved with the use of different kinds of Neural Networks. Marlot [40] in his work outlines how Neural Networks accounting for the time context having better results than ones without any modelling like Convolutional Neural Networks.
However, results for that little set are really encouraging ones, summarized in the table 6.1 with about 10% improvement doubling the number of bands per octave in the frequency analysis regarding mean F-measure.
Another big improvement can be verified in the number of False Negatives, reduced by a factor of $\frac{1}{3}$ from the original 12 band per octave system thanks to a refined frequency resolution. Also the Recall measurement is affected by the increase in the number of used bands, resulting in an improvement of about 15%.

| | | True Positive | False Positive | False Negative | F-measure | Precision | Recall |
|---|---|---|---|---|---|---|---|
| 12 bands | Sum | 2735 | 63 | 1888 | 0.73710 | 0.97748 | 0.59161 |
| 12 bands | Mean | 1367 | 31 | 944 | 0.67654 | 0.96309 | 0.53998 |
| 24 bands | Sum | 2174 | 172 | 674 | 0.83712 | 0.92668 | 0.76334 |
| 24 bands | Mean | 1087 | 86 | 337 | 0.83701 | 0.92588 | 0.76442 |

*Table 6.1: Evaluation metrics for MAPS dataset*

The network trained on the MAPS dataset was applied for the transcription and evaluation to the acoustic piano jazz set with the outcome of 0.50 for what concerns the f-measure metric, underlining a variation depending on the complexity of the single music piece. The table 6.2 collects all the sensitivity metrics for the jazz dataset, that are 20-30% lower than the MAPS one. The number of False-Negatives is considerably higher compared to the one detected in the MAPS set. A concrete reason can be found in the intrinsic complexity of the jazz genre due to improvisation. Furthermore, we should highlight how well the method can be applied to different environments keeping in mind that the MAPS dataset is mostly related to classical music, while the new jazz dataset includes a wide variety of sounds, musical figures and styles derived directly from the specific analysis of jazz piano performances.

| | True Positive | False Positive | False Negative | F-measure | Precision | Recall |
|---|---|---|---|---|---|---|
| Jazz Sum | 1390 | 851 | 1789 | 0.51292 | 0.62026 | 0.43724 |
| Jazz Mean | 695 | 425 | 894 | 0.50191 | 0.60723 | 0.42777 |
| MAPS Sum | 2174 | 172 | 674 | 0.83712 | 0.92668 | 0.76334 |
| MAPS Mean | 1087 | 86 | 337 | 0.83701 | 0.92588 | 0.76442 |

*Table 6.2: Evaluation metrics for Jazz dataset*

Finally, results can be satisfying and the prediction figure, plot against the target. It shows a really sharp detection of onsets, as retrieved from the studies done on the different SoundFonts in section 4.5, not forgetting that the precision in pitch estimation is also quite high. Despite some errors derived from octave mismatch, there is still a good connection between detected notes and the spectrogram which can be understood from the simplest diagram of simple scales 6.2 6.3. When analyzing more complex ones, it is more difficult to understand which are the frequencies since multiple notes can be played at the same time or closely together one to the other causing an overlap of frequency. Figure 6.4 shows a really clean prediction diagram thanks to simplicity of the performance. Although in the other plots 6.5 and 6.6 predictions are still sharp in terms of note onset, MIDI note errors can occur due to multiple note playing.

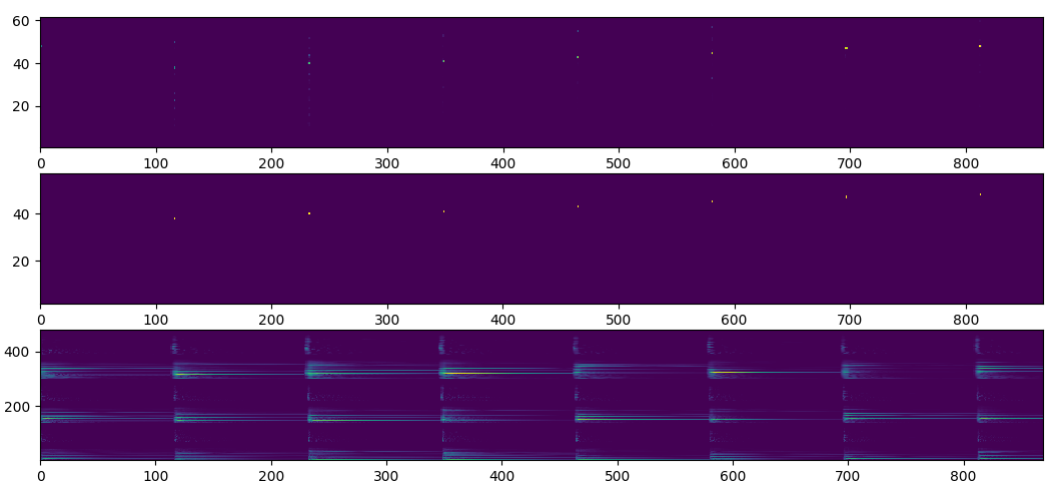


*Figure 6.2: Diatonic A major scale. Top: Predictions; Middle: Target; Bottom: Spectrogram*

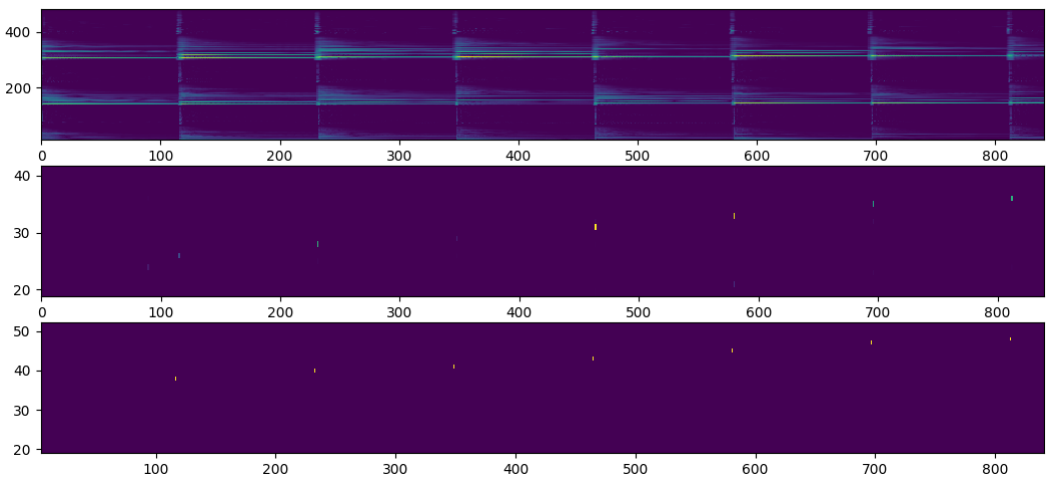


*Figure 6.3: Diatonic C major scale. Top: Spectrogram; Middle: Target; Bottom: Predictions*

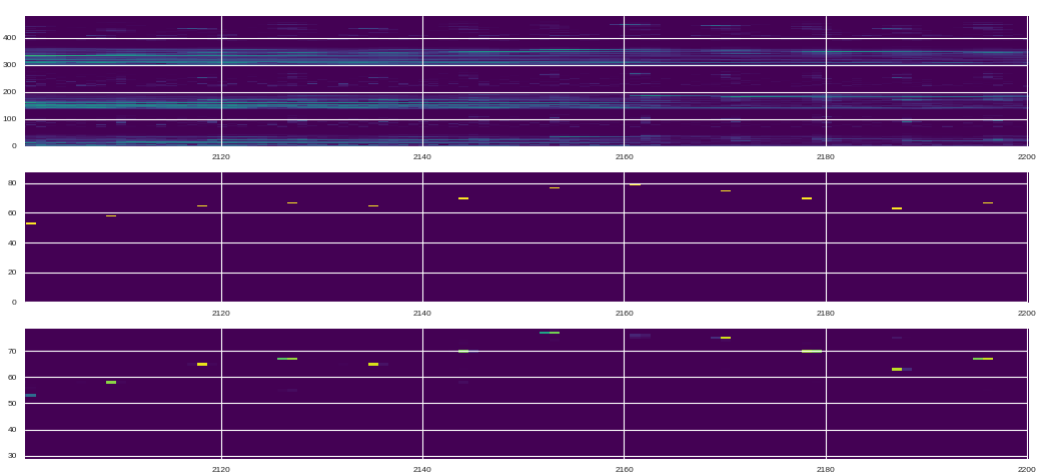


*Figure 6.4: Simple jazz performance. Top: Spectrogram; Middle: Target; Bottom: Predictions*

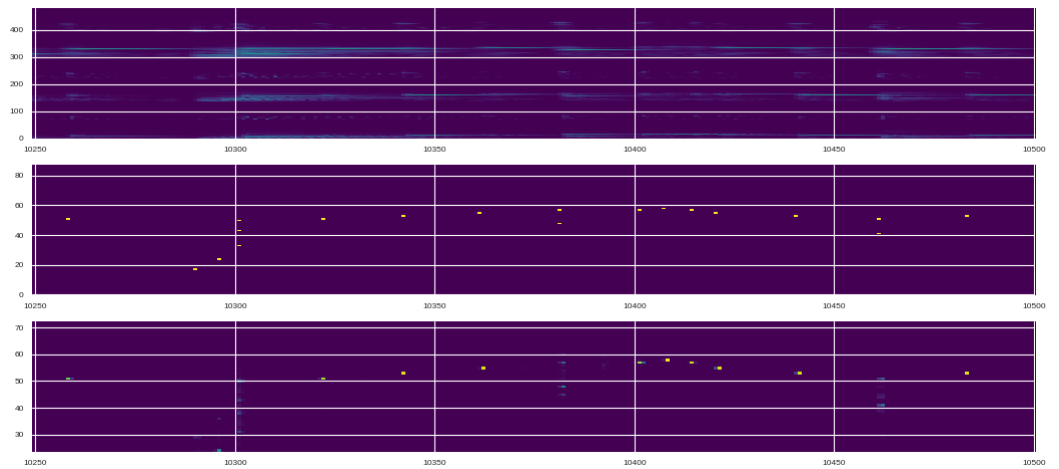


*Figure 6.5: Articulated jazz performance. Top: Spectrogram; Middle: Target; Bottom: Predictions*

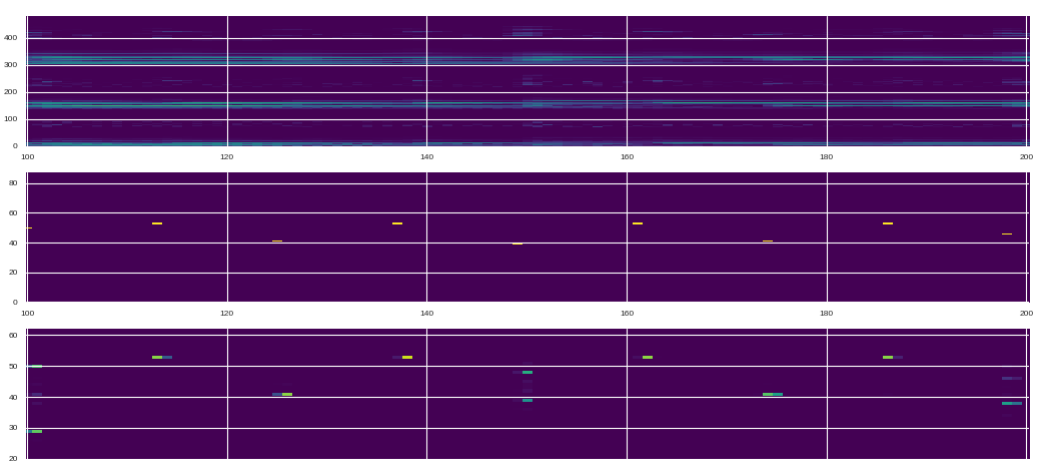


*Figure 6.6: Jazz performance affected by octave errors. Top: Spectrogram; Middle: Target; Bottom: Predictions*

## 6.3 State-of-the-art comparison

Results of the proposed work are compared to other approaches, comprising the Böck one, which seems to be the most promising. Accuracy measurements were taken following the one proposed by Dixon [45]. The table 6.3 collects the accuracy measurements for different state-of-the-art algorithms. Unfortunately, the proposed method results in a low accuracy measurements concerning the jazz set due to already explained reasons. The intrinsic difficulties of the genre bring the algorithm up to a high number of False Negative detection compared to the results retrieved by the MAPS experiment. Indeed, performances of MAPS set are comparable to the other state-of-the-art approaches, if not better. It can be also observed from the table 6.3 that our system also has performances close to the Boogaart and Lienhart one [38], which was trained with a single MIDI instrument. This is remarkable, since our system is not trained specifically for a single instrument.
The same trend can be observed in the table 6.4, where the jazz set results in worse outcomes than MAPS and other approaches.

| Experiment | ACCURACY |
|---|---|
| Jazz | 0.358 |
| MAPS | 0.863 |
| Böck [34] | 0.856 |
| Poliner and Ellis [21] | 0.623 |
| Boogaart and Lienhart [38] | 0.874 |

*Table 6.3: Accuracy measure*

| Experiment | PRECISION | RECALL | F-MEASURE |
|---|---|---|---|
| Böck [34] | 0.640 | 0.728 | 0.680 |
| Marlot [40] | 0.794 | 0.722 | 0.754 |
| MAPS | 0.837 | 0.925 | 0.764 |
| Jazz | 0.501 | 0.607 | 0.427 |

*Table 6.4: Sensitivity metrics*

# Chapter 7

# Conclusion and future works

In this thesis a methodology for automatic music transcription based on machine learning has been presented. The proposed technique is applied to jazz music. The peculiarity of jazz music is enclosed in the variety of styles and in particular how frequently performances are influenced by musicians' skills and way of playing. The coexistence of accompaniment and improvisation parts play an important role within the thesis. The latter is focused on the transcription of polyphonic piano pieces, which in jazz can play the role of first voice performing improvisation lines, or accompaniment instrument following those lines suggested from the lead sheet score. However, improvisation is still the most interesting part in music for AMT systems. In jazz, it plays the main role within a performance, in which, in turn, each musician can have a moment dedicated to express their own virtuosity. Indeed, improvisation is based just on the score sheet, that provides the main melodic lines of a musical piece. But it leaves the musician free to interpret and recompose new melodies upon the lead sheet, sometimes even not following any harmonic rule.
From the results we can observe how the number of False Negatives have decreased by 1/3 with a consequent increase of F-measure and Recall by about 10% varying the number of bands used for the spectral analysis. Despite the doubling of the number of bands and the near doubling of physical resources, the time taken for the training of the Neural Network increases by just 30%. Improvements on two of the three sensitivity metrics are justified in the better frequency resolution, achieved by increase the number of bands in which it is divided for each octave. For an instrument like the piano, rich in frequency content, a better frequency resolution is fundamental for the

precision of such systems. Indeed, Böck [34] and Kelz [29] exploited the 36 bands per octave setting in their works.
Results of the proposed transcription system are quite interesting and still leave a great deal of opportunity to improve the method also with the use of different kind of Neural Networks.
Despite using techniques to reduce the quantity of data derived from the feature extraction phase, problems emerging from the massive amount of information and time consumed limited a complete evaluation of the method.

## 7.1 Future works

The system was prepared to deal with many dataset organizations, and it can be evaluated on all these datasets in a separated way or in a merged one. The problems composing a unique massive dataset are time and resource allocation. Indeed, the same problem raised during the utilization of the jazz dataset for the training and the evaluation, since the whole jazz set reached a number of about 500 musical pieces. The large amount of information available within the dataset allows the setting of different experiments including the onset detection of multi-instrument tracks (available in the accompaniment audio and annotated files).
The target extraction performed during the dataset preparation allows the system to be evaluated in different modes. Throughout this research, we proposed just an onset note-based type of evaluation. However, it can be extended to frame-based or note-based onset and offset approaches just changing the target on which the network needs to act. A really approximate frame-based evaluation was extracted from the study of simple scales, showing slightly lower results (5%) against note-based evaluation. Furthermore, the original three-cross validation experiment can be evaluated to understand how different types of the same instrument can affect the transcription. From the three-cross evaluation, inference on the influence of tonal components and frequency contents of a piano sound can be derived.
A complete open future task, as anticipated in the evaluation section, would be the use of more promising types of Neural Networks. As seen also in other works, Neural Networks modeling temporal contents really helps the audio analysis field due to the high correlation of signals. The section 2.5 analyzes the results of different methodologies applied to multi-pitch estimation, highlighting a promising trend concerning the use of Neural Networks. This

is also confirmed by Marlot's studies [40] on different Neural Networks and Böck [34] method results in comparison with other approaches.

# Bibliography


[1] Moorer J.A. "On the transcription of musical sound by computer". In: (1977).

[2] Plumbley M. D.; Abdallah S. A.; Bello J. P.; Davies M. E.; Monti G.; Sandler M. B. "Automatic Music Transcription and Audio Source Separation". In: (2002).

[3] Emiya Valentin; Bertin Nancy; David Bertrand; Badeau Roland. *MAPS-A piano database for multipitch estimation and automatic transcription of music*.

[4] Berliner Paul F. *Thinking in Jazz*. 1994.

[5] christianparlet. *John Coltrane - Blue Train (Live in Stockholm 1961)*. Youtube. 2012. URL: `https://www.youtube.com/watch?v=JHKSvKFlfak`.

[6] everythingchangesmoi. *John Coltrane - Blue train*. Youtube. 2009. URL: `https://www.youtube.com/watch?v=XpZHUVjQydI`.

[7] Emmanouil Benetos; Simon Dixon; Dimitrios Giannoulis; Holger Kirchoff; Annsi Klapuri. "Automatic Music Transcription: breaking the glass ceiling". In: (2012).

[8] Benetos Emmanouil; Ewert Sebastian; Weyde Tillman. "Automatic transcription of pitched and unpitched sound from polyphonic music". In: (2011).

[9] Klapuri Annsi. *Signal Processing Methods for the Automatic Transcription of Music*. 2004.

[10] Benetos Emmanouil; Dixon Simon; Giannoulis Dimitrios; Kirchhof Holger; Klapuri Anssi. "Automatic music transcription: Challenges and future directions". In: (2013).

[11] Bello Correa Juan Pablo. *Towards the Automated Analysis of Simple Polyphonic Music: A Knowledge-based Approach*. 2003.

[12] Goto Masataka. “A real-time music-scene-description system: predominant-F0 estimation for detecting melody and bass lines in real-world audio signals”. In: (2003).

[13] Barry L. Vercoe; William G. Gardner; Eric D. Scheirer. “Structured Audio: Creation, Transmission, and Rendering of Parametric Sound Representations”. In: (1998).

[14] Martin Keith D. “A Blackboard System for Automatic Transcription of Simple Polyphonic Music”. In: (1995).

[15] Martin Keith D. “Automatic Transcription of Simple Polyphonic Music: Robust Front End Processing”. In: (1996).

[16] Klapuri Annsi. *Multiple Fundamental Frequency Estimation Based*. 2003.

[17] Goto Masataka. “A Predominant-F0 Estimation Method for Polyphonic Musical Audio Signals”. In: (2004).

[18] Vincent Emmanuel; Bertin Nancy; Badeau Roland. “Adaptive harmonic spectral decomposition for multiple”. In: (2010).

[19] Smaragdis Paris; Raj Bhiksha; Shashanka Madhusudana. “A Probabilistic Latent Variable Model for Acoustic”. In: (2006).

[20] Graham Grindlay; Ellis Daniel P.W. “A probabilistic subspace model for multi-instrument polyphonic transcription”. In: (2010).

[21] Poliner Graham P.; Ellis Daniel P. W. “A Discriminative Model for Polyphonic Piano Transcription”. In: (2006).

[22] Sigtia Siddharth; Benetos Emmanouil; Cherla Srikanth; Weyde Tillman; d’Avila Garcez Artur S.; Dixon Simon. “An RNN-based Music Language Model for improving Automatic Music Transcription”. In: (2014).

[23] de Cheveigné A. *Multiple F0 estimation*. 2005.

[24] Goodfellow Ian; Bengio Yoshua; Courville Aaron. *Deep Learning*. 1988.

[25] Benetos Emmanouil. “Automatic transcription of polyphonic music exploiting temporal evolution”. In: (2012).

[26] Hornik Kurt; Stinchcombe Maxwell; White Halbert. “Multilayer feedforward networks are universal approximators”. In: (1989).

[27] He Kaiming; Zhang Xiangyu; Ren Shaoqing; Sun Jian. “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification”. In: (2015).

[28] Duan Zhiyao; Han Jinyu; Pardo Bryan. “Harmonically informed multi-pitch tracking”. In: (2009).

[29] Kelz Rainer; Dorfer Matthias; Korzeniowski Filip; Bock Sebastian; Arzt Andreas; Widmer Gerhard. “On the potential of simple framewise approaches to piano transcription”. In: (2016).

[30] Saito Shoichiro; Kameoka Hirokazu; Takahashi Keigo; Nishimoto Takuya; Sagayama Shigeki. “Specmurt Analysis of Polyphonic Music Signals”. In: (2008).

[31] Smith Jordan B. L.; Burgoyne Ashley J.; Fujinaga Ichiro; De Roure David; Downie Stephen J. “Design and creation of a large-scale database of structural annotations”. In: (2011).

[32] Emiya Valentin; Badeau Roland; David Bertrand. “Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle”. In: ().

[33] Bittner Rachel;Salamon Justin; Tierney Mike; Mauch Matthias; Cannam Chris; Bello Juan. “MedleyDB: A multitrack dataset for annotation-intensive MIR research”. In: (2014).

[34] Böck Sebastian. *Event Detection in Musical Audio: Beyond simple feature design*. 2016.

[35] Glorot Xavier; Bengio Yoshua. “Understanding the difficulty of training deep feedforward neural networks”. In: (2010).

[36] URL: %7Bhttp://www.music-ir.org/mirex/wiki/MIREX_HOME%7D.

[37] Yeh C.; Robel A.; Rodet X. “Multiple fundamental frequency estimation of poliphonic music signals”. In: (2005).

[38] Gregor van den Boogaart;Rainer Lienhart. “Note onset detection for the transcription of polyphonic piano music”. In: (2009).

[39] Richard Vogl; Matthias Dorfer; Knees Peter; Widmer Gerhard. “Drum transcription via joint beat and drum modeling using Convolutional Recurrent Neural Networks”. In: (2017).

[40] Marlot Matija. “A connectionist approach to Automatic Transcription of polyhonic piano music”. In: (2001).

[41] Hartmann William M. “Pitch periodicity and auditory organization”. In: (1996).

[42] Stephen Downie. “Music Information Retrieval”. In: (2003).

[43] Goto Masataka; Hiroki Hashiguchi; Takuichi Nishimura; Ryuichi Oka. “RWC Music Database: Music Genre Database and Musical Instrument Sound Database”. In: ().

[44] URL: %7Bhttps://mido.readthedocs.io/en/latest/index.html%7D.

[45] Dixon Simon. “On the computer recognition of solos piano music”. In: (2000).