AUTOMATIC MUSIC TRANSCRIPTION

by

OLIVER IGNETIK

A THESIS SUBMITTED FOR THE DEGREE OF

BACHELOR OF ENGINEERING

in

DIGITAL SIGNAL PROCESSING

in the

UNDERGRADUATE DIVISION

of the

AUSTRALIAN NATIONAL UNIVERSITY

2020

Supervisors:
Associate Professor Parastoo Sadeghi, Main Supervisor
Professor Rod Kennedy, Co-Supervisor

Examiners:
Associate Professor Parastoo Sadeghi, ANU
Professor Rod Kennedy, ANU

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Oliver Ignetik

June 5th 2020

Contents

Acronyms

AMT Automatic Music Transcription

CQT Constant Q-Transform

DFT Discrete Fourier Transform

DTFT Discrete Time Fourier Transform

FFT Fast Fourier Transform

MIDI Musical Instrument Digital Interface

MFCC Mel Filterbank Cepstrum

MPE Multipitch Estimation

MIREX Music Information Retrieval Evaluation Exchange

MLM Music Language Model

NMF Non-negative Matrix Factorization

NN Neural Network

ReLU Rectified Linear Activation Unit

STFT Short Time Fourier Transform

Abstract

Automatic Music Transcription

by

Oliver Ignetik

Bachelor of Engineering in Digital Signal Processing

Australian National University

This research paper explores the concept of Automatic Music Transcription. A literature review is conducted to provide a concise overview of the subject, including state of the art methods and how they can be used to better improve user satisfaction of current systems.

In particular, this paper explores the method known as Non-negative Matrix Factorization as applied to time-frequency representations of audio signals. The primary concept that will be reviewed to aid with understanding this technique is the Short Time Fourier Transform.

A secondary avenue of exploration is machine learning algorithms and their application to Automatic Music Transcription systems. A preliminary review is provided to readily prepare the reader for the related discussions and insights uncovered in this investigation.

The design and application of a monophonic Non-negative Matrix Factorization and a polyphonic Neural Network system are presented followed by a discussion of the results. Thereafter, a discussion is presented on how higher level musical knowledge incorporated into future models to improve their accuracy.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 What is Automatic transcription ?

The nature of music signals, which often contain several sound sources that are highly correlated over both time and frequency, means that AMT is still considered an open problem in the literature. Usually an AMT system takes an audio waveform as input, computes a time-frequency representation and outputs pitches over time or ideally a typeset music score. Most approaches are designed to achieve an intermediate goal in AMT, which does not actually resemble musical notation as shown in Figure 1.1.

The capability of transcribing music audio into music notation is a fascinating example of human intelligence. It involves analyzing complex auditory scenes, recognizing musical objects, forming musical structures and checking alternative hypotheses. AMT refers to the design of computational algorithms to convert acoustic music signals into some form of music notation. It is a challenging task and considered an unsolved problem in signal processing and artificial intelligence. This problem is particularly challenging in polyphonic music were even the most advanced systems are far behind meeting the accuracy of trained musicians. [1]

1.2 Key Challenges

Despite significant progress in AMT research, there exists no end-user application that can accurately and realiably transcribe music containing the range of instrument combinations and genres found in recorded music.

There are several factors that make AMT particularly challenging :

1. *Polyphonic mixtures* - inferring musical attributes from a signal containing multiple simultaneous sources with different pitch, loudness and sound quality is extremely difficult. Even the task of disentangling the harmonics of two coinciding pitches is not trivial. For consonant intervals, which are often seen in diatonic harmonies and form basic harmonic buidling blocks, the notes share many of the same harmonics making the seperation of voices even more difficult. [2]

2. *Synchronous sound sources* - musicians pay close attention to metrical structure and rhythmic synchronicity, which violates statistical independence between sources which is often used in Automatic Speech Recognition to facilitate seperation.

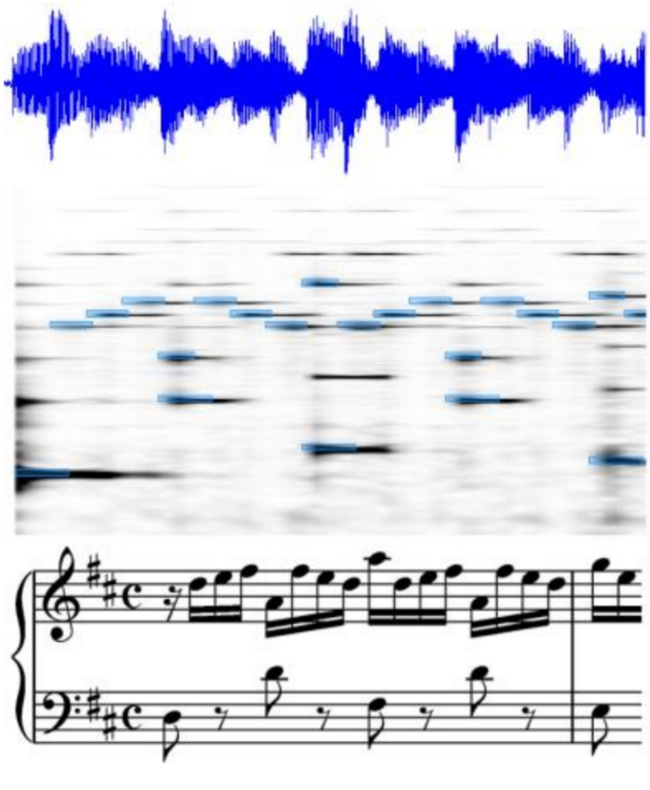

Figure 1.1: AMT process figure taken from NUS ISMIR 2019 tutorial [2]

3. *Lack of ground-truth transcriptions* - the annotation of polyphonic music is extremely time consuming and requires high expertise especially in symphonic pieces were there are many concurrent sound events. Even when there is sheet music available for a particular piece, they are difficult to align with an audio signal. Sheet music at best is considered as a weak label due to the fact subjective interpretation often plays a role. This is true even in the most prudent genres like classical music were musicians strive to pertain to the score as much as possible. [3]

1.3 Commercial Applications

A successful AMT system would enable a broad range of interactions between people and music, including automatic instrument tutoring, dictating improvised musical ideas and automatic music accompaniment, music content visualization and intelligent content-based editing, indexing and recommendation of music and analyzing jazz improvisations and other nonannotated music. Given the potential applications, the problem has attracted commerical interest and a number of AMT software exists. [4]

Commercially available applications include Melodyne (`http://www.celemony.com/en/melodyne`), Transcribe!(`https://www.seventhstring.com/xscribe/`) and AudioScore exist. In context-specific transcription scenarios these applications can reach multipitch detection accuracies of 90% or more. Even some open source academically produced applications can reach similar performance levels. [5] However, given complex ensemble pieces with multiple instruments the performance of such systems is still far behind that of a trained musician.

1.4 Overview

1.4.1 Research Question

Can incorporating higher level musical knowledge improve the performance of AMT systems?

1.4.2 Project Scope

The scope of the thesis will focused on western music with its associated modes and scales. This paper will be restricted to approaches that analyze music produced by pitched instruments such as pianos and guitars. Outside of the scope of the paper will be methods for transcribing percussive instruments such as drums.

1.5 Thesis Synopsis

Chapter 2 serves as a review of important musical concepts and a literature review in music signal processing. Crucial Digital Signal Processing (DSP) techniques are relations are presented. Finally a number of state of the art methods and their characteristics are explored.

Chapter 3 provides details on the system architecture of the AMT systems used in this research project. Chapter 4 presents the crucial results of this research paper and discusses their implications. Chapter 5 discusses the outcomes of the thesis and explores directions for future research.

1.6 Resources

1. Core Python Libraries - see github repo for thesis.yaml file for all dependencies

   a) LibROSA - LibROSA is a python package for music and audio analysis. It provides the building blocks necessary to create music information retrieval systems.
   `https://librosa.github.io/librosa/`

   b) mir_eval - Python library for computing common heuristic accuracy scores for various music/audio information retrieval/signal processing tasks.
   `https://pypi.org/project/mir_eval/`

   c) keras - High level deep learning library for python built on TensorFlow 2.0
   `https://keras.io/`

d) scikit-learn - Machine learning libraries for python.
`https://scikit-learn.org/stable/user_guide.html`

2. Datasets

   a) MAPS - A piano database for multipitch estimation and automatic transcription of music [6]

   b) MusicNet - A curated collection of labeled classical music [7]

   c) MAESTRO - MIDI and Audio Edited for Synchronous TRacks and Organization [8]

3. Work Environment

   a) Anaconda Package Manager

   b) Jupyter Lab NoteBooks

Chapter 2

Background

2.1 Musical Concepts

2.1.1 Pitch and Harmony

The existence of sequences of sounds with well-defined fundamental periods is a very common feature in music. Most musical instruments such as pianos, guitars, flutes and trumpets are constrcuted to allow performers to produce sounds with easily controlled fundamental periods and associated harmonics. Such a signal is described as a harmonic series of sinusoids at multiples of the fundamental frequency and results in the perception of a pitch in the mind of the listener.

Although different cultures have developed different musical conventions, a common feature is the musical "scale", a set of discrete pitches that repeats every octave. In contemporary western music an "equal tempered scale" is used, which divides the octave into 12 steps on a logarithmic axis called semitones.[9]

$$P_n = P_a(\sqrt[12]{2})^{n-a} \tag{2.1}$$

Where :

P_n = Query pitch
P_a = Reference pitch

In musical theory, the spacing in between these steps are known as semitones and form musical intervals. Different combinations of notes that form intervals result in different harmonic structures or "colours" known as chords. Consonant intervals like a perfect fifth are made up of seven semitones and have a frequency ratio of $(2^{\frac{1}{12}})^7 = \frac{3}{2}$ sounding pleasant to the ear. They share many harmonics and ubiquitous in western music. This is partly the reason why transcription can be so difficult. The tritone is considered dissonant and has a intervallic frequency ratio of $(2^{\frac{1}{12}})^6 = \frac{45}{32}$. This interval sounds jarring to the ear and is associated with musical tension. Tritones provide a harmonic spine for the movement of groups of notes because they are so noticable to the listener.

Musical Instrument Digital Interface (MIDI) is one of the most important tools for musicians. It is a protocal that allows computers, musical instruments and other hardware to communicate. It encodes an audio signal into an multi-dimensional array which contains information about the pitch and onset/offset times of notes. Of particular note is the MIDI pitch which has the formula below :

$$d = 69 + 12\log_2(\frac{f_0}{12}) \tag{2.2}$$

Where :

f_0 = fundamental frequency
d = MIDI number

On a grand piano the lowest note A0 has a frequency of 27.50 Hz and midi number of 21. The highest note C8 has a frequency of 4186.0 Hz and midi number 108.

2.1.2 Tempo, Beat and Rhythm

The musical aspects of tempo, beat and rhythm play a fundamental role. The *beat* can be described as a sequence of perceived pulses that are regularly spaced in time and correspond to the pulse a human taps along when listening to the the music. [10]

The strength or stress of the musical pulse and how it varies determines the metrical signature of a piece of music. Notes are grouped in rhythmic units in each bar according to the time signature.

The term *tempo* refers to the rate of this pulse as is often denoted as *beats per minute* or *bpm*. Musical pulses typically coincide with note onsets or percussive

Figure 2.1: Excerpt from a piano arrangement for the tune Nearness of You with a common time signature of 4 quarter notes per bar

events. In the context of AMT this tasks constitutes finding a *novelty curve* known as onset detection.

2.2 Signal Processing Techniques

2.2.1 Sampling Theorem

The sampling theorem is a consequence of digitizing analogue signals. Sampling an analogue signal stores quantized values of the amplitude of a continuous signal at regular intervals determined by the sampling rate.

The sampling theorem says that to avoid higher frequency components aliasing as lower frequencies components the following must be satisfied. Considering a sampling frequency F_s and Nyquist frequency F_N.

$$F_s > 2 \cdot F_N \tag{2.3}$$

Where F_N is the highest frequency expected in the signal. Frequently a sampling rate of 44.1 kHz is used in audio recording because the range of human hearing is from 20hertz-20kHz.

2.2.2 Discrete Fourier Transform

Consider a finite-length sequence $x[n]$ of length N samples such that $x[n] = 0$ outside the range $0 \leq n \leq N - 1$. To each finite-length sequence of length N, it is possible to associate a periodic sequence.

$$\tilde{x}[n] = \sum_{r=-\infty}^{\infty} x[n - rN] \tag{2.4}$$

This assumption is implied in the mathematics of the Discrete Time Fourier Transform (DTFT), that the signal of interest is periodic in nature even when it has a finite length. The DTFT of such a signal is given by :

$$\tilde{X}[\omega] = \sum_{n=-\infty}^{n=\infty} \tilde{x}[n] \exp^{-j\omega n} \tag{2.5}$$

This sequence is itself periodic with a period N. The Discrete Fourier Transform (DFT) of the original signal finite length signal $x[n]$ can be found by sampling $\tilde{X}$ at $\omega = \frac{2\pi}{N}$ and only considering the values of k within $0 \leq k \leq N-1$:

$$X[k] = \sum_{n=0}^{n=N-1} x[n] \exp^{-j\frac{2\pi}{N}k} \tag{2.6}$$

The DFT is often implemented as the Fast Fourier Transform (FFT) which reduces the order of complexity to $O(N \log N)$ by exploiting symmetries in the transformation. [11] Equation 2.6 will be used frequently throughout this project and extended upon in § 2.2.3.

2.2.3 Short-Time Fourier Transform

As in other audio-related applications, the most popular tool for describing the time-varying energy across different frequency bands is the Short Time Fourier Transform (STFT), which, when visualized as its magnitude, is known as the spectrogram.

Formally, let x be a discrete-time signal obtained by uniform sampling a waveform at a sampling rate F_s Hz. Using a N-point tapered window w (eg. Hamming $w[n] = 0.5 - 0.46 \cdot cos(\frac{2\pi n}{N})$ for $n \in [0, N-1]$) and an overlap of half a window length we obtain the STFT.

$$X[m,k] = \sum_{n=0}^{N-1} w[n] \cdot x[n + m \cdot \frac{N}{2}] \cdot exp\{-j\frac{2\pi kn}{N}\} \tag{2.7}$$

With $m \in [0, T-1]$ and $k \in [0, K-1]$. Here, T determines the number of frames , $K = \frac{N}{2}$ is the index of the last unique frequency value as dictated by the Sampling Theorem. Thus $X[m,k]$ corresponds :

$$f_{coeff}(k) = \frac{k}{N} \cdot F_s \qquad [\text{Hz}]$$
$$t_{frame}(m) = t \cdot \frac{N}{2F_s} \qquad [\text{s}] \tag{2.8}$$

$X[m,k]$ is complex-valued, with the phase depending on the alignment of each short-time analysis window. Often it is only the amplitude $\mid X[m,k] \mid$ that is used. [11]

2.2.3.1 Log-Frequency Spectrogram

Note that the Fourier coefficients of $X[m,k]$ are linearly spaced on the frequency axis. Using suitable binning strategies, various approaches switch over to a logarithmically spaced frequency axis, by using mel-frequency bands or pitch bands as seen in Figure 2.2. Keeping the linear frequency axis puts greater emphasis on the high-frequency regions of the signal, thus accentuating the aforementioned noise bursts visible as high-frequency content. One simple yet important step often applied in the processing of music signals, is referred to as logarithmic compression. Such a compression not only accounts for the logarithmic nature that describes how humans perceive sound but also balances out the dynamic range of the signal. Some variations of the traditional STFT include Constant Q-Transform (CQT) and Mel Filterbank Cepstrum (MFCC). [1]

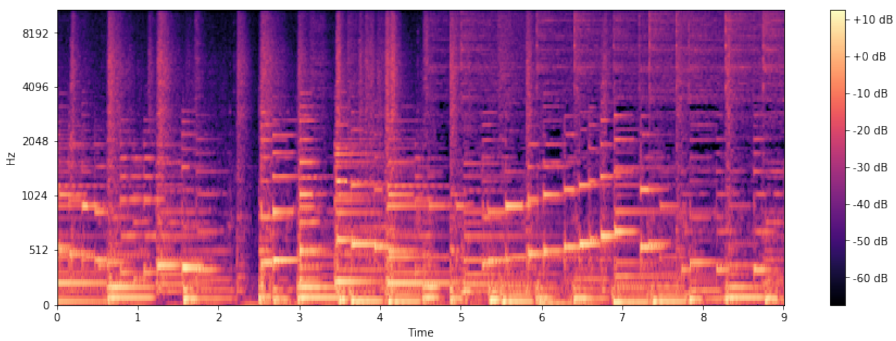

Figure 2.2: STFT of a 10s excerpt from Blues in F - Bill Evans Trio recording

2.2.3.2 CQT

CQT has a number of advantages over STFT in note identification since constant center frequency-to-resolution ratio results in a constant pattern of sounds with harmonic components in the logarithm-scaled frequency domain, which is easier for resolving notes that are played simultaneously. Not to mention the fact that it more closely resembles the human auditory system which makes it ideal in AMT [12].

2.3 State of the Art Methods

Many approaches have been developed for AMT applied to polyphonic music. While the end goal of AMT is to convert an acoustic music recording to some form of music notation, most approaches are aimed at achieving an intermediate goal. Some commercial applications provide the capability of converting a piano-roll representation into typeset music notation. However, the end results are generally musically illogical especially in genres like jazz where notes often fall on the upbeat and rythms are highly syncopated.

AMT approaches can generally be organized into four categories : frame-level, note level, stream level and notation level. Frame level transcription which is also known as *Multipitch Estimation (MPE)* aims at identifying the number and pitch of notes that are present in a frame of music. A frame is generally on the time scale of 10ms depending on the type of analysis window. Note-level transcription not only estimates the pitch in each time frame but also the onset and offset times. Stream level transcription of *instrument tracking* targets the grouping of estimated notes into streams. These groupings typically correspond to different instruments or timbres. Notation level transcription or *audio-to-note transcription* aims to transcribe the music audio into a musical score such as that seen on staff notation. Harmonic and rhythmic structures have to be incorporated into the modelling and as a result the complexity is monumentally higher then MPE approaches. [13]

Readers interested in a comparison of the performance of different approaches are referred to the Multiple Fundamental Frequency Estimation and Tracking task of the annual Music Information Retrieval Evaluation Exchange (MIREX) (`http://www.music-ir.org/mirex`).

2.3.1 Non-negative Matrix Factorization

A large subset of transcription systems employ methods stemming from spectrogram factorization techniques, which exploit the redundancies found in music spectrograms. Non-negative Matrix Factorization (NMF) was first proposed by Lee and Seung [14]

Starting with a non-negative M by N matrix $\mathbf{X}$ the goal of NMF is to approximate it as a product of two non-negative matrics $\mathbf{W}_{M\times R}$ and $\mathbf{H}_{R\times N}$, where $R \leq M$ such the cost function is minimized :

$$C = |\ \mathbf{X} - \mathbf{W}\cdot\mathbf{H}\ |_F \tag{2.9}$$

where $|\ .\ |_F$ is the Frobenius norm. This is actually equivalent to Gradient Descent based minimization of divergence. [15] There are a number of algorithms for finding the appropriate values of $\mathbf{W}$ and $\mathbf{H}$. For example, the generalized Kullback-Leibler divergence between $\mathbf{X}$ and $\mathbf{W}\cdot\mathbf{H}$ is non-increasing under the following updates and guarantees the nonnegativity of both $\mathbf{W}$ and $\mathbf{H}$:

$$H \Leftarrow H \odot \frac{W^T \frac{X}{WH}}{W^T J} \text{ and } W \Leftarrow W \odot \frac{\frac{X}{WH} H^T}{J H^T} \tag{2.10}$$

where the $\odot$ operator denotes pointwise multiplication, $J \in \mathbb{R}^{M\times N}$ denotes the matrix of ones, and the divison is pointwise. [16]

In the context of time-frequency representations and AMT both unknown matrices have an intuitive interpretation. $\mathbf{X}$ in the most basic cases in time-frequency analysis is a STFT of the audio signal. $\mathbf{W}$ encodes the spectral profiles of the R components and is commonly referred to as the dictionary matrix. $\mathbf{H}$ encodes the temporal activity of the each of those components and named the activation matrix.

There are two classes of NMF approches that fall into supervised and unsupervised approaches. In supervised approaches the dictionary matrix is prextracted. For explanatory purposes, one can imagine the applicaiton of such an NMF AMT system. To compile the dictionary matrix a recording of each note played in isolation is recorded and concatenated. Thus each component can be thought of corresponding to individual pitches with their associated harmonic profiles. The NMF-based decomposition is then performed by applying the update rules in Equation 2.10 to find $\mathbf{H}$ to minimize the cost function.

The unsupervised approach involves hyperparameter tuning to discover the optimal value for the number of components. This can be achieved by grid search methods cv-fold testing by splitting up the audio signal into smaller segments. Both approaches are widely used and there have been many studies based on improving performance and accuracy. For a comprehensive overview of a number of these techniques refer to [17–19].

State of the art applications of NMF for polyphonic AMT include work were sparseness constraints were added into the NMF update rules, in an effort to find meaningful transcriptions. [20] Another approach was based on incorporating harmonicity constraints in the NMF model, resulting in two algorithms: harmonic and inharmonic NMF. [21] Additionally the model constrains each basis spectrum to be expressed as a weighted sum of narrowband spectra, in order to preserve a smooth spectral envelope. The inharmonic version of the algorithm is also able to support deviations from perfect harmonicity and standard tuning. Also, another approach proposed a Bayesian framework for NMF, which considers each pitch as a model of Gaussian components in harmonic positions. [22] Spectral smoothness constraints are incorporated into the likelihood function, and for parameter estimation the space alternating generalised EM algorithm is employed.

More recently, one approach proposed an algorithm for MPE and beat structure analysis. The NMF objective function is constrained using information from the rhythmic structure of the recording, which helps improve transcription accuracy in highly repetitive recordings. [23]

2.3.2 Neural Networks

NN are systems that are vaguely inspired by biological neural networks. They are based on a collection of connected units or nodes called artifical neurons. They are able to learn non-linear functions from input to output via an optimization algorithm. The goal of a network is to learn the weights w_{ij} by minimizing the cost function with respect to the training data. [24]

NNs have a number of advantages over traditional machine learning algorithms. One of the main advantages is the removal of the need for feature extraction which is important in unstructured types of data like images or sound. The type of network architecture that will be discussed in this paper is known as a feed forward neural network. Deep networks partially replace the need for feature engineering. The

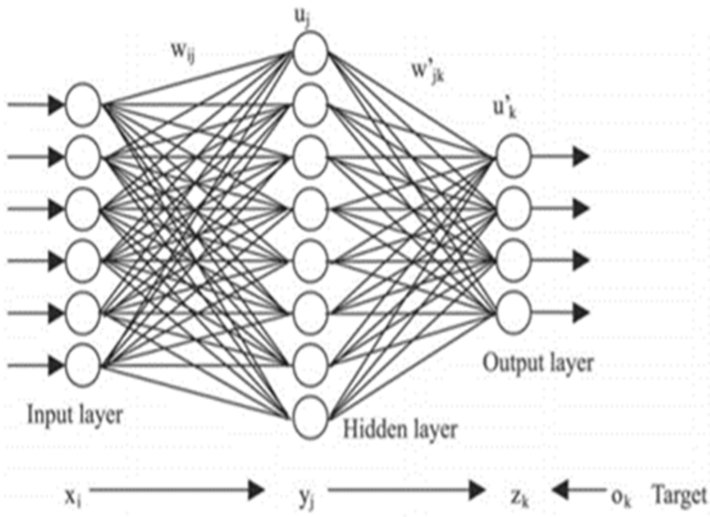

Figure 2.3: An example of a feed forward architecture neural network appropriate for classification tasks. [2]

deeper layers in the network, model increasingly more abstract and intricate features. In the context of music transcription, the layers closer to the input might model individual notes, whilst deeper layers, model features such as chords and harmonic progressions depending on the type of network used.

One extremely important concept that is pivotal for understanding the optimization of neural networks is gradient descent and back propogation. Backpropagation is an iterative optimization process used to train models. The goal is to find the lowest point of a multivariate loss function by incrementally updating each weight

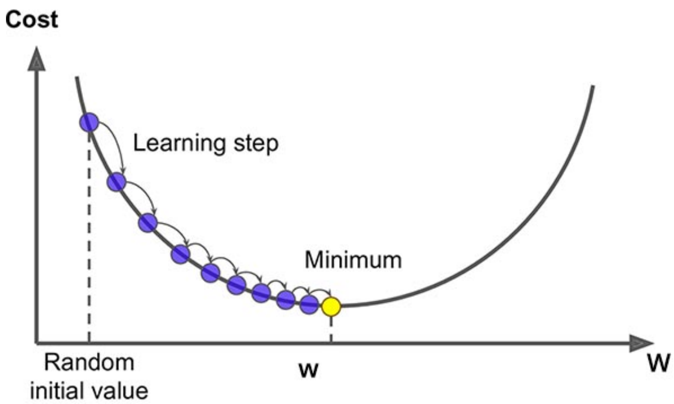

Figure 2.4: An intuitive graphical understanding of gradient descent optimization

Algorithm 1: Gradient descent optimization for one epoch with batch size equal to entire dataset

Data: Training dataset X
Result: Optimal weights that minimize loss function θ

Input: query weight θ_j
Output: Gradient ∇

Function `Calculate Gradient`(θ_j):
  $r_j \longleftarrow$ Slope of the loss function
  $\alpha_j \longleftarrow$ Slope of the activation function
  $\gamma_j \longleftarrow$ Value of the neuron that feeds into our weight
  $\nabla \longleftarrow \gamma_j \; r_j \; \alpha_j$
  return ∇

Input: Initialized weights θ, training Data X, learning rate L
Output: Optimized weights θ

Procedure `Gradient Based Optimization`(θ, X, L):
  for $X_j \in X$ **do**
    Perform forward propagation with X_j to make a prediction
    Use this prediction in back propogation to update weights
    $\nabla \longleftarrow$ `Calculate Gradient`(θ_j)
    $\theta_j \longleftarrow \theta_j - L \times \nabla$
  return θ

in the network by the product of the gradient and the learning rate. Algorithm 1 shows the steps that are used in gradient descent optimization in one epoch. An epoch refers to a complete pass through the training data. In other words all the samples have been passed through the model for training Through backpropagation, the loss or difference between the predicted value and the actual value, is transferred from one layer to another and the weights are modified so that the loss is minimized.

One other crucial concept in NNs and machine learning in general is the concept of overfitting. Typically, datasets are seperated into test and train sets. The model is exposed to the training data and is then used to predict on the test data. When the test accuracy of the model starts to decrease and the training accuracy further increases this is known as overfitting. This is because the model is no longer capturing the underlying relationships in the data but is rather overaccommodating to the subtleties in the training data. The end goal of the network should be to predict on any unseen dataset that it has not been exposed to and perform accurately.

There are a number of important parameters which have to be tuned in a neural network through a process known as hyperparameter tuning. This is typically done using a grid-search algorithm to find a set of parameters which optimizes the cost function. [12] Some of these hyperparamters important for tuning network performance are given below :

2.3.2.1 Important Features in NNs

Several aspects of NNs require further explanation as they are not obvious. These parameters play a pivotal role in the performance and accuracy of the network. This section will briefly discuss each of these parameters and certain trade offs that they present.

1. *learning rate* - the rate at which the weights are updated in the optimization process. If the learning rate is too high the model may fail to converge in optimization.

2. *activation functions* - mathematical equations that determine the output of a node in network. They determine whether it should be activated or not based on the input to the node. A number of common activation functions that are used in different types of problems include : ReLU (Rectified Linear Unit), hyperbolic tangent, softmax and sigmoid activations. Each activation function is employed depending on the type of problem and corresponding optimizer and loss function used.

3. *optimizer* - the type of optimization algorithm employed to train the model and update the weights of the model. One of the most well known optimization algorithms is called gradient descent which has a number of variants such as stochastic and mini-batch gradient descent. Some optimization algorithms are more appropriate for certain types of problems such as regression or classification problems.

4. *epoch* - when an entire dataset is passed forward and backward through the NN only once. Typically setting a higher number of epochs will lead to higher

accuracy in the training set but their is a risk of overfitting.

5. *batch size* - total number of training samples present in a single portion of the dataset that is is used to update the weights in backpropagation. There is typically a tradeoff with batchsize, computational efficiency and accuracy. A smaller batch size requires more time for training but is generally more accurate. A common batch size that is used is 32 which is referred to as a mini-batch.

6. *loss function* - the loss function is used in backpropagation to update the weights so as to increase the prediction accuracy of the model. They are mathematical functions that measure the difference between the predicted output and the actual output. Some common loss functions include mean squared error, hinge, binary crossentropy (See Figure 2.5) and the kullback leibler divergence.

In recent years NNs have had a considerable impact on the problem of music transcription and on music signal processing in general. However, compared to

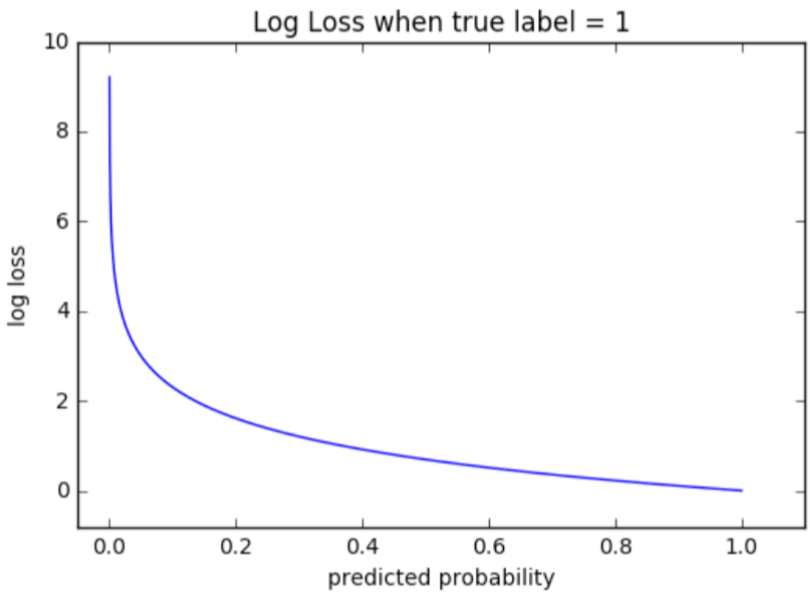

Figure 2.5: Binary cross entropy function that is useful in classification tasks used in conjuction with a softmax activation function on the output

other fields progress on NNs for music transcription has been slower due to a lack of annotated data with labels which is essential for training the models appropriately. [25]

The current state of the art method for piano transcription was proposed in research completed by Google Brain [26]. This approach combines two networks, one which detects onsets and one which finds note lengths. The output from the note onset network is used to inform the second network calculating note lengths.

Despite the appeal of NNs and the promises they hold they are often still outperformed by NMF-based methods for a number of reasons :

1. Lack of annotated labelled datasets - NNs rely on data to be effective. There are only a a small number of annotated datasets which in themselves are restricted to certain types of instruments and genres of music. [3]

2. Adaptablity to new conditions - there are currently no methods to retrain or adapt an NMF-based AMT systems on only a few seconds of audio. As such NMF-based systems can perform considerably better with less data and are easier to adapt.

2.4 Summary

In general there are drawbacks and advantages to both types of approaches outlined in this chapter. NNs can be more effective in context dependent environments whilst NMFs show more adaptablity. Currently there are no methods which are favoured in all situations.

Chapter 3 will introduce an AMT system which uses NMF to transcribe single note melodies and discuss the system architecture. Chapter 3 will also discuss how to apply a NN to musical data obtained from 2018 MusicNet database.

Chapter 3

System Design

The AMT problem can be divided into several subtasks, which include: multipitch detection, note onset/offset detection, loudness estimation and quantisation, instrument recognition, extraction of rhythmic information, and time quantisation. The core problem in automatic transcription is the estimation of concurrent pitches in a time frame, also called multiple-F0 or multi-pitch detection. [4]

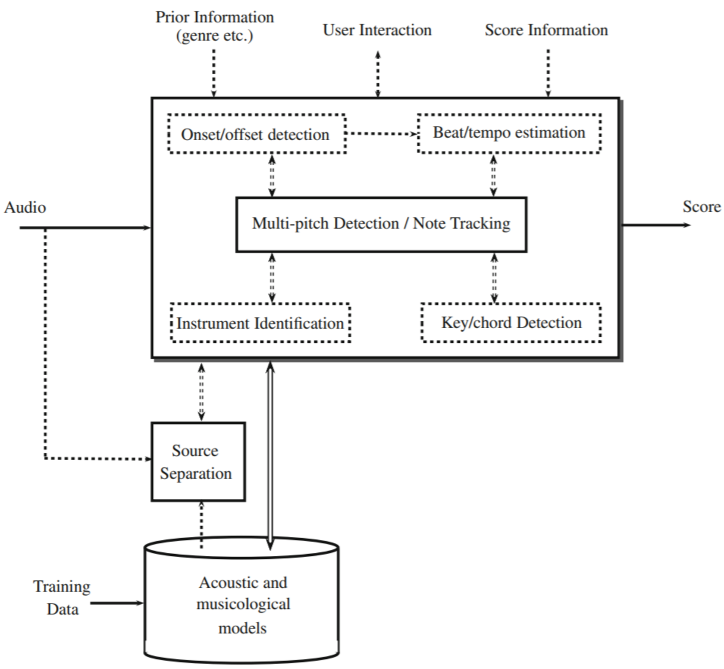

Figure 3.1: Typical architecture of an AMT system. [4]

Onset Time (s)	Offset Time (s)	MIDI pitch
1.12	1.42	23

Table 3.1: Example ground truths format

3.1 Preliminaries

This report will investigate how to use NMF in interpreting an audio recording and extracting music information from this recording. There are two systems and two applications that will be be presented in this section.

1. NMF applied to single line melody - exploration of parameters used to fine tune model accuracy
   a) MAPS Dataset - chromatic scale played on a Bechstein D 280 in a concert hall [6]
   b) Type of architecture - Supervised NMF
2. NN applied to a large music database
   a) Dataset MusicNet recordings and active frame note labels [7]
   b) Type of architecture - Feed Forward neural network with mutli-label classification

3.2 Methodology

3.2.1 NMF application to monophonic AMT

The sample used in this analysis is a recording of a chromatic scale played on a Bechstein D 280 piano in a concert hall environment retrieved from the MAPS database [6].

It should be noted that in this experiment the offset time will be omitted from the investigation as this is much more difficult to infer from the audio signal. The audio data is loaded into the python kernel using LibROSA and then it is passed to the NMF_model class which is the main class for performing the experiments

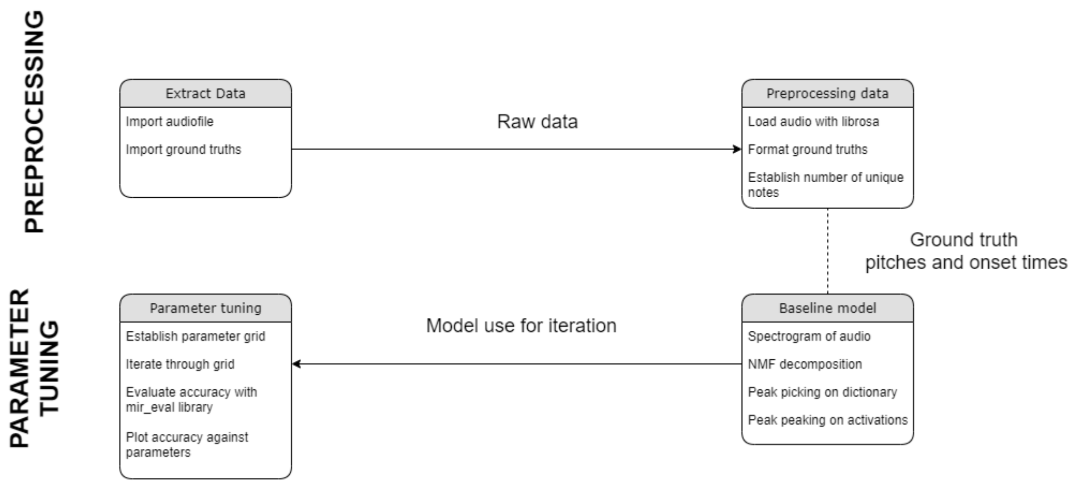

Figure 3.2: System model for NMF AMT

in this investigation. As can be seen in Figure 3.2 the audio data is passed to the NMF class where the constructor takes two important parameters to determine the spectrogram *hop length* and *window size*. The NMF class has the following important methods :

- make_stft - builds the spectrogram based on the hop length and window size
- NMF_decomposition - performs the NMF decomposition
- estimate_onsets - find the note onsets using the peak picking algorithm
- estimate_pitches - find the active pitches in the sample

3.2.1.1 Evaluation

A set of parameters to test in the model is instantiated as a dictionary with name and data keys to allow for flexible access. This dictionary is then iterated through and the models accuracy for each parameter is evaluated using the mir_eval library.

3.2.2 NN Application to polyphonic AMT

A feed forward NN will be used to transcribe music from the musicnet database. The pieces are polyphonic which means the complexity is much higher due to multiple instruments being played at the same time.

Parameters	Value
Input nodes	252
Output nodes	88
Optimizer	Adam
Output activation	Sigmoid function
Hidden activation	ReLU function

Table 3.2: Critical Baseline model parameters

A baseline model is established which will be tuned by scanning the hyperparameters of the model. Once optimal parameters have been established the model will be trained for a larger number of epochs to learn the dataset more comprehensively. The model will be evaluated using an appropriate metric for multilabel classification tasks. The model will be implemented using Keras with a Tensorflow backend and a high level overview of the model is shown in Figure 3.3 The defining features of the model architecture are provided in Table 3.2

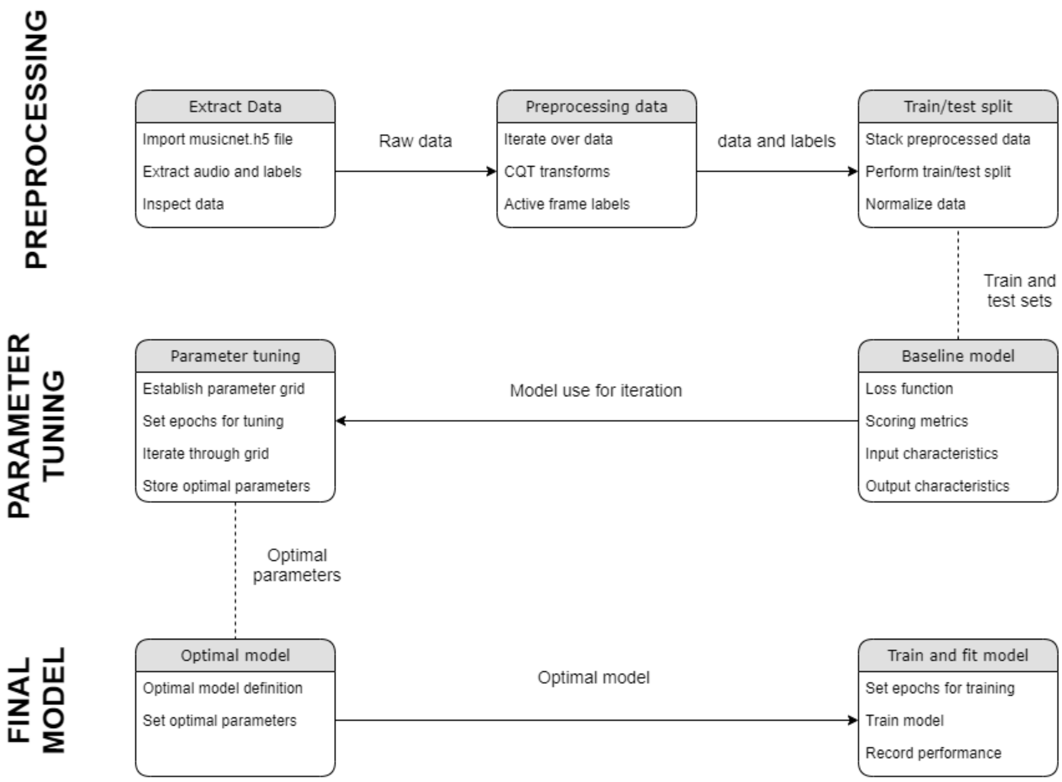

Figure 3.3: System model for NN AMT

3.2.3 Dataset

All of the data for the model comes from the musicnet database [7]. The outstanding feature of this dataset is that each audio signal comes with the associated ground truth onset and offset times and pitches. The data is split into training, validation and test sets based on a 60-20-20 percentage ratio.

3.2.3.1 Features

The audio signal is transformed into the CQT domain which is closely related to the STFT but is better for multipitch scenarios due to the higher fidelity in note resolution. The CQT is downsampled from 44.1 kHz to 16 kHz with a hop length of 512 corresponding to 32 ms frames. Consequently there are 252 CQT features per frame. Therefore, a data matrix with the size of $252 \times$ number of frames is obtained from the CQT transformation.

3.2.3.2 Labels

The ground truth labels are sampled at each frame to determine the active notes in the audio. There are 88 possible notes on the piano and as such an array with length 88 constitues the active and inactive notes at each frame in the audio, with 1 representing active and 0 representing inactive.

3.2.3.3 Loss Function

A weighted binary cross entropy will is used as the loss function for optimizing the wieghts of the model. A normal binary cross entropy function (shown in Equation 3.1) is not appropriate in this case as the ratio of inactive notes to active notes notes is on average 10-15 throughout the database.

$$L^i(y_i, \hat{y}_i) = -(y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)) \tag{3.1}$$

This means that the model would overfit to the inactive notes to minimize the loss function which is essentially the easy way out. A weighted binary cross entropy function accounts for this by informing the model to fit more for active notes.

3.2.3.4 Activation functions

Rectified Linear Activation Unit (ReLU) functions will be used as the activation function for the hidden layers and a sigmoid activation function will be used on the output layer as is required in multilabel classification tasks. The sigmoid activation function is shown in Equation 3.2.

$$F(x) = \frac{1}{1 + e^{-x}} \tag{3.2}$$

3.2.3.5 Evaluation

Accuracy is not an appropriate metric in this kind of multilabel classifiction task due to the imbalance between inactive and active notes. A better measure to use that takes into account both the precision and recall of the model is the f-measure as shown in Equation 3.3. The f-measure used in this experiment is based on the f1-score from the sklearn.metrics libary but is modified so it is appropriate for the multilabel nature of this problem.

$$Precision(P) = \sum_{i=0}^{N} \frac{TP(i)}{TP(i) + FP(i)}$$

$$Recall(R) = \sum_{i=0}^{N} \frac{TP(i)}{TP(i) + FN(i)} \tag{3.3}$$

$$F\ measure = 2 \times \sum_{i=0}^{N} \frac{P \cdot R}{P + R}$$

Chapter 4

Results

4.1 NMF approach to monophonic AMT

4.1.1 Discussion

In this section the results obtained by the NMF model will be discussed. Generally the model performed well achieving and accuracy of 0.90 or higher dependent on the values of different parameters. The four most influential parameters tested in the model will be discussed in this section.

4.1.1.1 Window Size

Through inspecting Figure 4.1 (a) it is shown that the overall accuracy of the model decreased as the window size increased. This can be explained by referring back to Equation 2.8 which encapsulates the concept of frequency time resolution trade off in spectrograms. The ideal time window size to use in this situation is 3004 samples as this translates to a frequency resolution of 14.68 Hz. This is equivalent to a semi-tone in music which is part of the "equal tempered scale" as shown in Equation 2.1.

4.1.1.2 Peak Picking Threshold Value

One of the crucial parameters in the model is the use of a peak picking algorithm that depends on a number of parameters. Most notable of these parameters is the delta threshold value which translates to the sensitivity of the model to ambient noise. The NMF model returns a dictionary matrix and an activation matrix. The

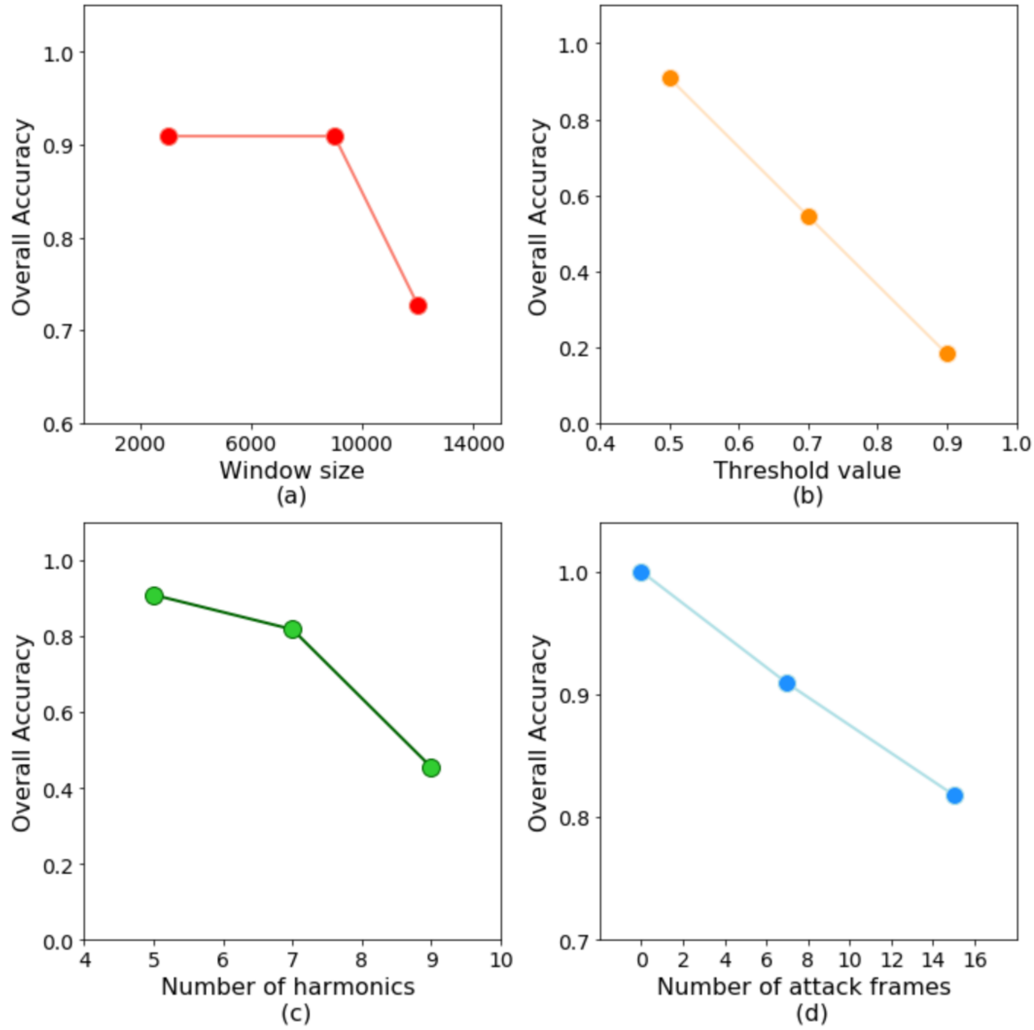

Figure 4.1: Hyperparameter tuning with accuracy metrics for monophonic AMT system

dictionary matrix (example shown in Figure 4.2) can be thought of as DFTs of the seperate notes that together construct the final audio signal. The threshold function accounts for ambient noise and is context dependent on the recording environment. This means for each sample recording and differing instrument the threshold function has to be tuned. It can be seen in Figure 4.1 (b) that by increasing the threshold the overall accuracy decreases. Thus the optimal range for this recording environment is approximately 0.5.

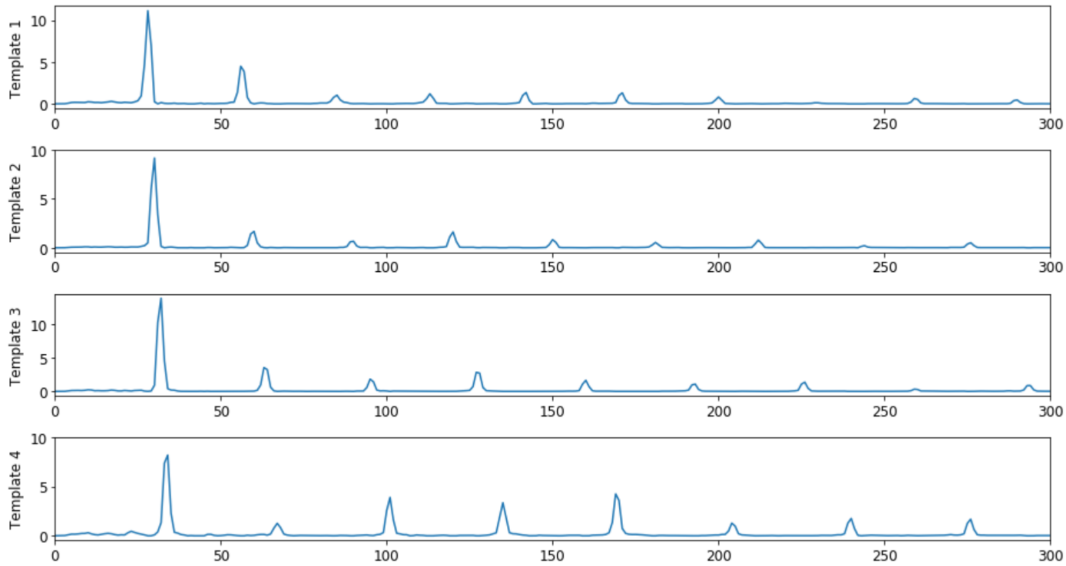

Figure 4.2: Example of dictionary components

4.1.1.3 Number of harmonics

The number of harmonics is another crucial parameters in finding the fundamental frequency. As discussed in § 2.1.1 a pitched sound comprises of a series of harmonics that are integer multiples of the fundamental frequency. Infact only a number of harmonics are needed for the pitch to be percieved in the mind of the listener. By inspecting Figure 4.1 (c), it is evident that only a small number of harmonics are needed for good performance in the model. However, it is noted that as the number of harmonics included in the calculation of the fundamental frequency has an inverse relationship with overall accuracy. This can be explained by a concept known as inharmonicity. This phenomenon is present in many pitched instruments where the harmonics are not perfect integer multiples of the fundamental frequency due to properties of the instrument that vary of time such as string tension and hammer velocity in pianos.

4.1.1.4 Number of attack frames

The last important parameter that will be discussed in this section is the number of attack frames. This parameter is important for predicting the onset time of each note event. Figure 4.3 is the activation matrix showing the temporal evolution of the notes. Typical pitched events have four phases that musicians would be well aware

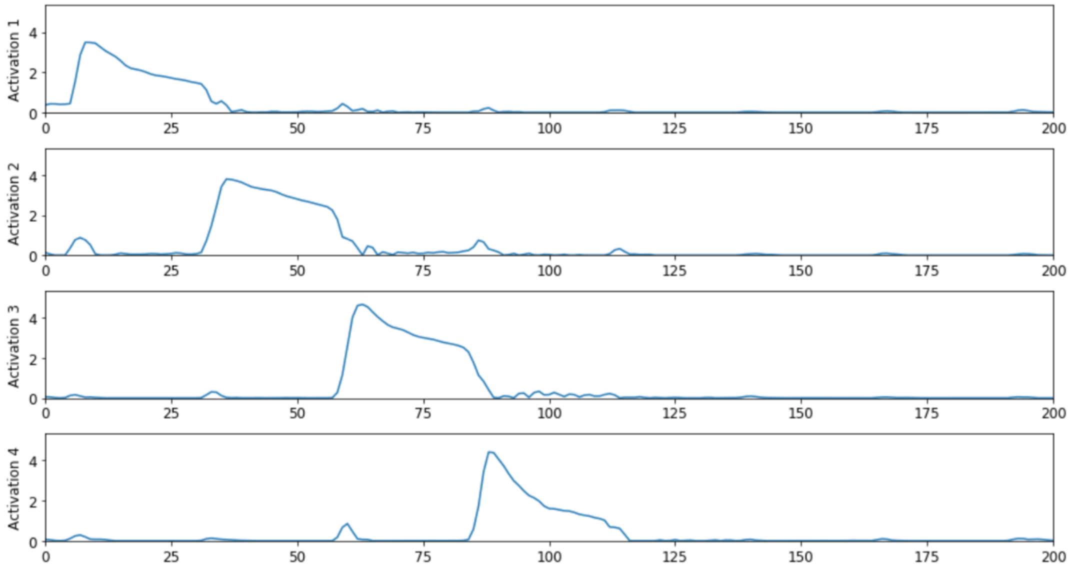

Figure 4.3: Example of activation components

of attack, decay, sustain and release. The pitched event will be best percieved in the mind of the listener during the sustain period as a stable frequency of oscillation will be achieved. This experiment, however, revealed that disregarding the attack frames infact increased the model performance as shown in Figure 4.1 (d). The number of attack frames also depends on the type of recording environment and the type of instrument used. So given a different instrument the number of attack frames may have to be changed.

4.1.2 Summary

In summary the NMF model performed extremely well for monophonic AMT however is not appropriate for polyphonic transcription. The crucial points of discovery are:

- NMF model provides high accuracy metrics for monophonic music
- Fast performance and compile time
- Needs fine tuning dependent on recording environment and instrument type
- The attack frames had a negligible effect on model performance

Parameters	Value
Neurons per layer	512
Dropout Rate	0.91
Layers	3
f1-score	**0.75**

Table 4.1: Optimal parameters for model

4.2 NN approach to polyphonic AMT

4.2.1 Discussion

In this section the results obtained by the NN model will be discussed. Generally the model performed well achieving a validation f1-score of 0.75 after 100 epochs performed on the training data. The most important insights will be discussed including the Hyperparameter tuning, choice of loss function and metric of performance. The optimal parameters are shown in Table 4.1.

4.2.1.1 Hyperparameter grid search

In order to investigate the effect of the model parameters on performance a number of parameters were investigated as shown in Table 4.2. As the number of layers or number of neurons goes up the compile and training time of the model goes up so this must be carefully considered when using NN models. The dropout rate is important to set to a value of 0.1 to ensure that the model does not overfit to the training data. The ultimate goal of the model is to apply it to "unseen data" to fully test its predictive power. This means the model is highly dependent to genre, recording environment and instrumentation. This model would not perform well on a music database that is vastly different as it would need to be retrained which takes considerably more time then using an NMF model which is one major drawback. Training this model for 100 epochs to achieve the observed f1-score had a relatively long time of completion.

4.2.1.2 loss function and metric of performance

The behaviour of the curves in Figure 4.4 indicates that the use of the weighted cross entropy function in combinations with the modified f1-score were well suited to

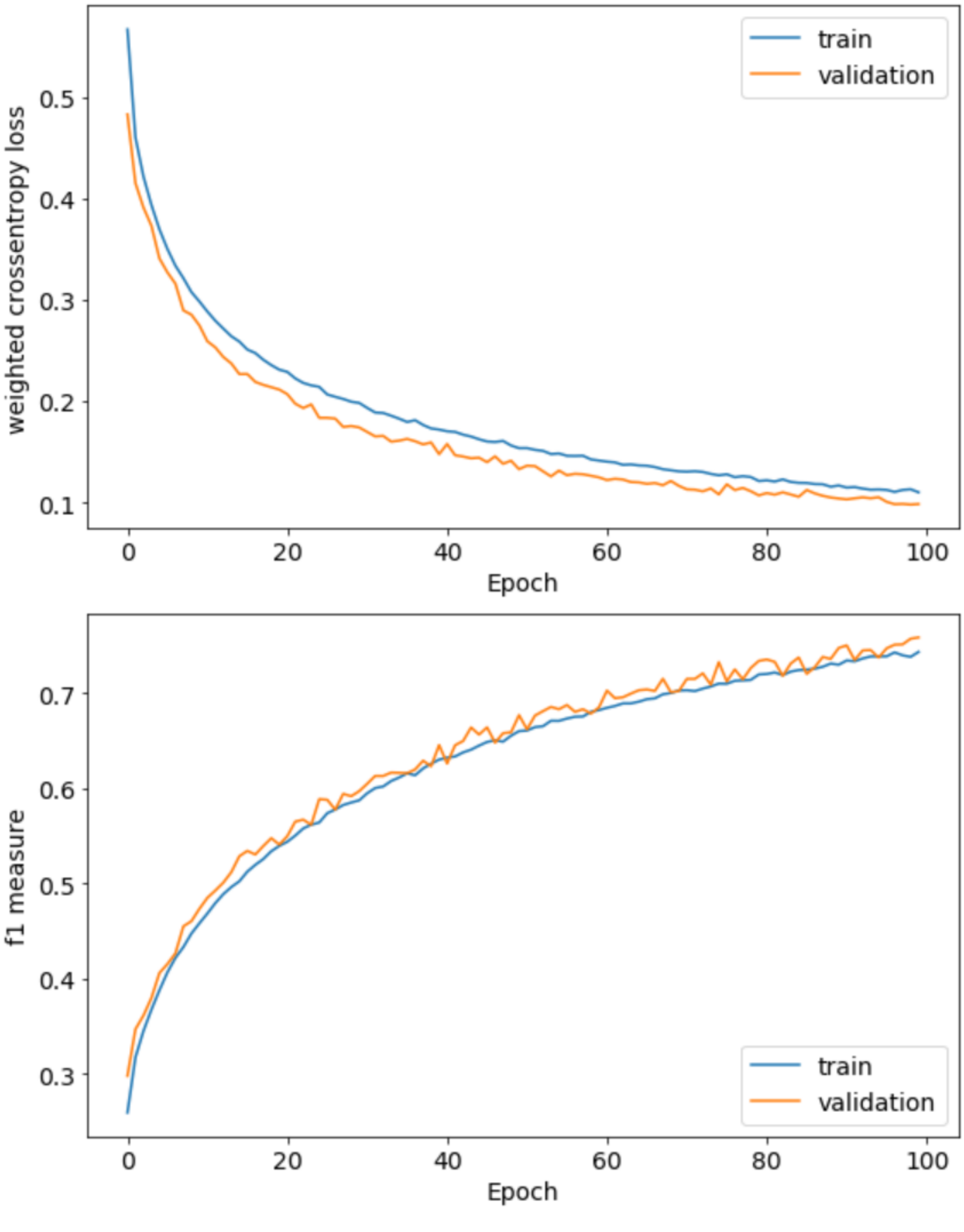

Figure 4.4: Loss function and f1 measure curves for neural network model with optimal parameters

	validation scores	
Parameter	**f1-measure**	**loss score**
layers		
2	0.374	0.374
3	0.382	0.368
4	0.370	0.378
Dropout Rate		
0.0	0.444	0.311
0.1	0.382	0.369
0.2	0.350	0.406
Neurons in hidden layers		
128	0.359	0.398
256	0.381	0.371
512	0.399	0.346

Table 4.2: Hyperparameter tuning for NN approach on polyphonic AMT

this type of problem. Initially a regular binary cross entropy with binary accuracy was used however the model performed very poorly in that it overfit to inactive notes and provided a false sense of accuracy due to this class imbalance.

4.2.2 Summary

In summary, the NN model performed well for polyphonic AMT however further work is needed to convert the output to an interpretible form that more closely resembles the piano MIDI roll representation. As it stands the results are still considered a mid-level representation that is far from resembling a musical score which is the ultimate goal of AMT systems. The crucial points of discovery are:

- Stable loss curve illustrates use of proper accuracy metrics and loss functions
- f1-score was suitable for polyphonic music case
- Takes time to retrain data
- Output is not yet in piano MIDI form

Chapter 5

Conclusion

5.1 Future Research Directions

Despite significant progress in the field of AMT as can be seen by inspecting the MIREX results [27] of recent years. The performance of even the most recent systems falls well below that of a human expert particularly in symphonic music where this is many simultaneous instruments. [28]

5.1.1 User Informed Transcription

The fact that current AMT systems do not reach the same level of precision in extracting information from music audio signals as trained musicians do, suggests that a human in the loop system whereby the user provides input to the system to attain satisfactory results.

Humans are extremely good at instrument identification, note onset detection, and segregation. While computers are capable of performing operations quickly on extremely large datasets. [2] *Semi automatic approaches* may be able to to obtain results faster then human transcription and more accurate then fully automatic approaches. [2]

Tha main effort in this research avenue should be to the type of input that users can provide which is most beneficial to the system and how to incorporate high level abstract musical concepts.

One approach which incorporates user feedback requires the indentification of the type of instrument and scale or notes used. [29] The technique used in this approach performed considerably better then a fully automatic approach using the same type of algorithm. Another approach required the user to hum the melody which was used to help extract it from the mixture signal. [30]

5.1.2 Score Informed Transcription

The musical score of a piece can provide invaluable information for AMT systems to exploit. In certain situations, take for example classical performances, a method known as score-to-audio alignment can be used. [31] Automatic music tutoring applications are becoming more popular in recent years with the advent of such programs as Fender Play, Yousician and more taking advantage of this idea. In the use cases correctly played passages need to be identified along with mistakes made by the student. However, these applications are based around correcting local mistakes in pieces and do not correct major changes in performances such as the form of a piece. Finally, the more challenging problem of lead-sheet informed transcription is almost unexplored with no notable published papers at this time. A lead sheet can be thought of as a blueprint to an improvisor indicating only the melody and harmonic progression. These are very weak labels and make incorporating the information they provide extremely difficult. To conclude while this problem has been explored for certain instruments such as the piano there are many other instruments still yet unexplored and the task of lead-sheet informed transcription remains unexplored.

5.1.3 Context specific transcription

The ultimate goal of a complete multi-instrument AMT system without specific knowledge of any contextual parameters such as instrumentation or recording conditions is not yet achievable. However, considerable progress has been made by incorporating contextual parameters into existing pitch detection algorithms. For example multipitch detection accuracy in context-specific piano transcription can now exceed 90%. [5]

Most transcription algorithms that are based on heuristic procedures even deliberately disregard specific timbral characteristics in order to enable independence of instrumentation in pitch detection. Even those transcription methods that are tested on specific datasets are not tailored to that particular instrument.

Transcription systems typically model a wide range of instruments employing a

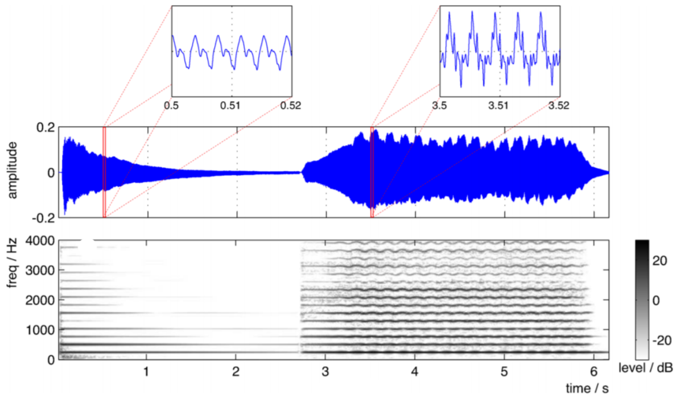

Figure 5.1: Middle C (262 Hz) played on a piano and a violin. The top pane shows the waveform, with the spectrogram below. Zoomed-in regions shown above the waveform reveal the 3.8-ms fundamental period of both notes [1]

single set of algorithms, assuming that it can be applied equally well to different kinds of instruments and situations. The NMF approach presented in this paper had no information about instrument-specific harmonic profiles incorporated into the model. Depending on the sound production mechanism of instruments, the the characteristics of the harmonics require the introduction of instrument specific parameters in the common models used. For example the spectral characteristics of a note produced on a violin are quite different to those produced on a piano.

5.1.4 Evaluation Metrics

Some notes are more musically important than others and as such some errors are more noticable to human listeners then others. For example, in certain genres like pop music, wrong notes within the scale are less noticable then those from notes outside the scale with alot of tension. Most AMT approaches are evaluated using the set of metrics proposed for the MIREX Mutliple-F0 Estimation and Note Tracking public evaluation tasks. While the metrics do provide insight into building successful AMT systems they do not correspond with human perception of transcription accuracy. Take for example, a repeated missed note compared to a repeating jumping octave or a note in a completely different register. Some ideas that have been proposed include observing how music teachers grade music dictation exams and better understanding the cognitive processes behind processing music in humans. [13]

5.1.5 Music Language Models

AMT systems can model both the acoustic sequences and the underlying notes over time. It provides the main link between music signal processing and symbolic music processing. Some approaches have attempted to incorporate what are known as Music Language Model (MLM)s to better improve the transcription accuracy. These models attempt to encode higher level musical structures such as metric signature, scales and chords and key signature. Key is a high-level musical cue that provides useful information prior to transcription about the combination of potential notes and chords. This can be achieved by giving more wieght to predictions made within the same key. Furthermore, musicological models could be used to describe longer-term relationships in audio recordings such as song structure and modulations between keys. This alludes to the fact that most existing AMT systems are data driven and often the errors they make are not musically meaningful. In the analogous field of Speech Recognition acoustic and language models are applied with great success. However these models can not be directly applied to AMT systems because :

- Music is polyphonic
- Music rhythm involves much longer temporal dependencies
- Music harmony arrangement involves rich music theory

1

Some of the most promising applications of these ideas made use of Recurrent Neural Networks which can better model long term dependencies in time. These approaches were noted to achieved 10% improvements in frame-level transcription accuracy with respect to similar models that did make use of MLM. Further work is needed to better encode high level musical structures into current systems. [32, 33]

5.1.6 Parameters for NNs

There needs to be substantial work in discovering appropriate loss functions to be used in optimization of NN models. The choice of loss function is directly related to the activation function used in the output layer of the NN. This paper showed that despite treating an AMT problem as multi-label probelm with a softmax activation function on the output layer binary crossentropy was not a suitable loss function to be used in optimizing the model. This was shown to be related to the class imbalance between inactive and active notes in each frame. However, by using a weighted function that takes into account this imbalance the model had stable loss curves. In recent times investigations in to recurrent neural networks have shown the most promise as they are believed to be modelling long term interdependencies in between notes which may be improving model performance by capturing harmonic relations in the deeper layers of the network. [32, 33]

5.2 Conclusion

There are several issues in the AMT problem and if these are not addressed the performance of current systems will never be sufficient for certain applications. This paper has reviewed the current state of AMT research in certain key areas and identified major challenges and outlined promising directions for future work.

A potential way forward in the field is to make use of more information in the form of incorporating high-level musical conventions, instrumental characteristics or explicit user input to resolve ambiguties. In Chapter 4 it was discussed how context can be used to inform high-level models that are more powerful then generalized models as they can encode important information about key, instrument identities and metrical structure that can inform the pitch detection algorithms.

To potentiate progress in these research avenues, expertise from several disciplines will be needed such as audio engineering, musicology and acoustics. Furthermore there needs to be a greater emphasis on incorporation of end-user applications that provide crucial feedback on how musicians interact with AMT systems and what are the most salient features in music recordings.

The work outlined in this paper illustrates key approaches and techniques that have been developed in the rapidly evolving field of music signal analysis, but as discussed there is much room for improvement and for new inventions and discoveries, leading to more powerful and innovative applications. For the moment, human listeners remain far superior to machines in interpreting information in music signals. However, by addressing the major challenges presented this gap will be greatly reduced by unlocking the full potential of music signal processing techniques.

Bibliography

[1] M. Müller, D. P. Ellis, A. Klapuri, and G. Richard, "Signal processing for music analysis", *IEEE J. Sel. Topics Signal Process*, vol. 5, no. 6, pp. 1088–110, 2011.

[2] E. Benetos, "Automatic music transcription", Tutorial presented at National University of Singapore, University of London, January 2019. [Online]. Available: `http://c4dm.eecs.qmul.ac.uk/`.

[3] L. Su and Y.-H. Yang, "Escaping from the abyss of manual annotation: New methodology of building polyphonic datasets for automatic music transcription", *Proc. Int. Symp. Computer Music Multidisciplinary Research*, pp. 309–321, 2015.

[4] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription : Challenges and future directions", *J. Intelligent Inform. Syst.*, vol. 41, no. 3, pp. 407–434, 2013.

[5] A. C. Z. Duan and B. Wohlberg, "Context-dependent piano music transcription with convolutional sparse coding", *IEEE/ACM Trans. Audio, Speech Language Process*, vol. 24, no. 12, pp. 2218–2230, 2016.

[6] V. Emiya, R. Badeau, and B. David, "Multipitch estimation of piano sounds using a new probalistic spectral smoothness principle", *IEEE Transations on Audio, Speech and Language Processin*, To be Published. [Online]. Available: `http://www.tsi.telecom-paristech.fr/aao/en/2010/07/08/maps-database-a-piano-database-for-multipitch-estimation-and-automatic-transcription-of-music/`.

[7] J. Thickstun, Z. Harchaoui, D. P. Foster, and S. M. Kakade, "Invariances and data augmentation for supervised music transcription", in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018.

[8] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, “Enabling factorized piano music modeling and generation with the MAESTRO dataset”,, 2019. [Online]. Available: `https://openreview.net/forum?id=r1lYRjC9F7`.

[9] W. Ye, “Perceptual features in music signal processing”, YouTube video of NUS lecture recording, 2019.

[10] F. Lerdahl and R. Jackendoff, “Generative theory of tonal music”, *MA : MIT Press*, 1983.

[11] A. V. Oppenheim and R. W. Schafer, “Discrete-Time Signal Processing”, 3rd ed. Pearson, 2010.

[12] L. Li, I. Ni, and L. Yang, “Music transcription using deep learning”,, 2019.

[13] M. Schedl, E. Gómez, and J. Urbano, “Music information retrieval: Recent developments and applications”, *Foundations Trends Inform. Retrieval*, vol. 8, pp. 127–261, 2014.

[14] D. Lee and H. Seung, “Leaning the parts of objects by non-negative matrix factorization”, *Nature*, vol. 401, pp. 788–791, 1999.

[15] P. Smaragdis and J. Brown, “Non-negative matrix factorization for polyphonic music transcription”, *IEEE Workshop Applications Signal Processing Audio and Acoustics*, pp. 177–180, 2003.

[16] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, “Automatic music transcription”, *IEEE SPS Journal*, vol. 36, no. 1, 2019.

[17] A. Cheveigne, “Multiple f0 estimation”, *Computational Auditory Scene Analysis*, 2006.

[18] M. Christensen and A. Jakobsson, “Synthesis lectures on speech audio process”, in. San Rafael, CA: Morgan and Claypool, 2009, ch. Multi-pitch estimation.

[19] A. Klapuri and M. Davy, “Signal Processing Methods for Music Transcription”, New York: Springer, 2006.

[20] A. Cont, “Realtime multiple pitch observation using sparse non-negative constraints”, *In 7th international conference on music information retrieval*, 2006.

[21] E. Vincent, N. Bertin, and R. Badeau, "Adaptive harmonic spectral decomposition for multiple pitch estimation", *IEEE Trans. Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 528–537, 2010.

[22] N. Bertin, R. Badeau, and E. Vincent, "Enforcing harmonicity and smoothness in bayesian nonnegative matrix factorization applied to polyphonic music transcription", *IEEE Trans. Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 538–549, 2010.

[23] K. Ochiai, H. Kameoka, and H. Sagayam, "Explicit beat structure modeling for non-negative matrix factorization-based multipitch analysis", *Int. conf. audio, speech, and signal processing*, pp. 133–136, 2012.

[24] I. Goodfellow, A. Courville, and Y. Bengio, "Deep Learning", Cambridge: MIT Press, 2016.

[25] R. G. C. Carvalho and P. Smaragdis, "Towards end-to-end polyphonic music transcription: Transforming music audio directly to a score", *IEEE Workshop Applications Signal Processing Audio and Acoustics*, pp. 151–155, 2017.

[26] E. Elsen, C. Hawthorne, J. Song, A. Roberts, I. Raffel, and J. Engel, "Onsets and frames: Dual-objective piano transcription", *Proc. Int. Society Music Information Retrieval Conf.*, 2018.

[27] "Music information retrieval evaluation exchange (mirex)", 2011. [Online]. Available: `http://www.music-ir.org/mirex`.

[28] X. Serra, M. Magas, E. Benetos, M. Chudy, S. Dixon, A. Flexer, E. Gómez, F. Gouyon, P. Herrera, S. Jorda, O. Paytuvi, G. Peeters, J. Schlüter, H. Vinet, and G. Widmer, "Roadmap for music information research", 2013.

[29] H. Kirchhoff, S. Dixon, and A. Klapuri, "Shift-variant non-negative matrix deconvolution for music transcription", *Int. conf. audio, speech, and signal processing*, pp. 25–128, 2012.

[30] P. Smaragdis and G. Mysore, "Separation by humming: User-guided sound extraction from monophonic mixtures", USA: New Paltz, 2009.

[31] S. Wang, S. Ewert, and S. Dixon, "Identifying missing and extra notes in piano recordings using score-informed dictionary learning", *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 25, no. 10, pp. 1877–1889, 2017.

[32] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, “Modelling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription”, *Proc. Int. Conf. Machine Learning*, pp. 1159–1166, 2012.

[33] S. Sigtia, E. Benetos, and S. Dixon, “An end-to-end neural network for polyphonic piano music transcription”, *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 5, pp. 927–939, 2016.

Appendix A

Code for NMF approach

A.1 Instructions

1. Please visit the project github repo
2. Download the jupyter notebook file named *NMF-final-report.ipynb*
3. Download the audio file named *MAPS_ISO_CH0.3_F_AkPnBcht.wav*
4. Download the ground truths file named *MAPS_ISO_CH0.3_F_AkPnBcht.txt*
5. Download the *thesis.yml* file to use the same environment and packages used in the project

Appendix B

Code for NN approach

B.1 Instructions

1. Please visit the project github repo
2. Download the jupyter notebook file named *NN-final-report.ipynb*
3. Download the hdf5 file named *musicnet.h5* from musicnet website
4. Download the *thesis.yml* file to use the same environment used in the project