Monophonic Piano Music Transcription Using Machine Learning

Presented by:

Longa Maimbo

Prepared for:

Yaaseen Martin

October 25, 2024

Submitted to the Department of Mechatronics Engineering at the University of

Cape Town in partial fulfilment of the academic requirements for the degree of Bachelor of Science in Engineering in Mechatronics.

Abstract

This project focuses on the development of an Automatic Music Transcription (AMT) system specifically designed for monophonic piano music, utilizing machine learning techniques to convert WAV audio inputs into MIDI files suitable for further processing and sheet music generation. Various feature extraction techniques and neural network models were evaluated to identify the optimal configuration. The system demonstrated strong performance on monophonic music, particularly after the application of a post-processing algorithm that improved its transcription accuracy. However, testing on polyphonic music revealed a significant number of false positives, indicating the model's limitations in this context. The findings suggest that while the model effectively identifies onsets and pitches, its performance metrics are affected by inaccuracies in transcriptions. Future work will focus on developing more perceptual metrics for evaluating transcription quality, expanding the training dataset for greater variability, and enhancing post-processing methods, including the potential integration of Hidden Markov Models. Overall, this project lays the groundwork for a robust AMT system capable of handling polyphonic music transcription, with implications for both music analysis and composition.

Acknowledgments

I would like to express my profound gratitude to God for guiding me through every challenge and inspiring me to push the for greater success even in my weakest moments. To my parents, thank you for your endless support and belief in my dreams your encouragement has been a constant source of strength, and guidance to me even in the most uncertain of days. I am especially grateful to my supervisor, Dr. Yaaseen Martin, for your insightful guidance and unwavering support and welcoming attitude towards me throughout this semester . Your mentorship has not only shaped my research but has also taught me the importance of resilience and passion in pursuing my thesis. I would also like to thank my friends who have stood with me through thick and thin they are truly my found family.

Plagiarism Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the **IEEE** convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed and has been cited and referenced. Any section taken from an internet source has been referenced to that source.
3. This report is my own work and is in my own words (except where I have attributed it to others).
4. I have not paid a third party to complete my work on my behalf. My use of artificial intelligence software has been limited to **Formatting and rewording** (specify precisely how you used AI to assist with this assignment, and then give examples of the prompts you used in your first appendix).
5. I have not allowed and will not allow anyone to copy my work with the intention of passing it off as his or her own work.
6. I acknowledge that copying someone else's assignment or essay, or part of it, is wrong, and declare that this is my own work.

[illegible]

Longa

Maimbo

October 25, 2024

Contents

List of Figures

List of Tables

Chapter 1

Introduction

No matter where a person may be in the world, whether they speak the language of the place they are in or not, one form of expression that will always be understood is the music we hear. From a sombre piece meant to instil a feeling of sadness to the most jovial of melodies. Instrumental music is able to clearly communicate what it wants to without any language barrier, it is within this quality that the true power of music lies. However, quite often musicians may craft the most astonishing pieces of music whilst improvising, as such they are unable to recapture the beauty of their piece. This is where the fundamental need for an automatic music transcription (AMT) system lies.

1.1 Background to Study

In recent years AMT has become an increasingly popular tool in music production and analysis. It provides an accessible means through which people can convert audio recordings into musical notation such as sheet music or MIDI files.

With the ever growing use of digital tools in the music industry AMT provides people at every musical skill level with the ability to efficiently transcribe performances they hear live. These technologies facilitate musical education and documentation.

Currently many studies have gravitated towards the use of machine learning models such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks(RNNs) in an attempt to improve the accuracy of AMT systems. While they have seen improvements in certain areas such as monophonic piano transcription, challenges remain particularly in regards to transcribing polyphonic instruments.

Existing AMT systems still struggle to obtain accurate onsets and offsets especially when transcribing polyphonic music that may be rich in harmonics. Therefore is a need for AMT systems that can more accurately transcribe audio.

This project will aim to produce an AMT model that able to transcribe monophonic piano music and will also explore post processing techniques to improve obtained results using custom machine learning models.

By developing a reliable AMT system for monophonic music transcription it could serve as a basis that others may be able to develop into more complex models that could perform significantly better.

1.2 Problem Statement

AMT is a challenging task that aims to convert audio recordings of music into symbolic representations, such as a musical score. Despite advances in machine learning and digital signal processing, accurately transcribing a musical piece remains difficult, especially with polyphonic instruments like the piano.

The difficulty in AMT stems from the highly complex nature of music. Music has many variations in rhythm, pitch, timbre, and other characteristics, which makes accurate transcription a challenging task. In the current state of music information retrieval (MIR), AMT systems often struggle to detect onsets and offsets, and they further struggle with filtering out harmonics.

Even with the advent of machine learning techniques, such as neural networks—particularly architectures like CNNs, LSTMs, and DNNs—AMT systems still fall short of being truly exceptional. This could be attributed to the limited availability of high-quality annotated datasets, especially across different musical genres. Additionally, AMT systems are often sensitive to background noise, which can significantly impact transcription accuracy. Most AMT systems also suffer in accuracy when handling complex polyphonic compositions with overlapping notes.

This study seeks to address the main challenges of onset detection and pitch classification in AMT, with a focus on piano music. By finding the best combination of machine learning architectures and feature extraction techniques, and using post-processing to further improve the accuracy of the transcriptions, this research aims to advance the performance of AMT systems.

Figure 1.1: Audio wave carrying musical information [1]

1.3 Objectives

The ultimate goal of this project is to utilise machine learning techniques to create a piano note classification model that is able to effectively transcribe piano audio into a useful format. This format would allow further processing for analysis or can even be used as sheet music by a musician. In order to achieve this the project must accomplish the following objectives.

- A comprehensive study on the history and current state of AMT with a focus on how machine learning is deployed in this area. A firm understanding of this will provide a solid foundation through which all other objectives may be approached
- An investigation into music theory and its implications in AMT. A deep rooted understanding of music theory should make approaching individual problems easier and could open the avenue for much more creative solutions.
- Reviewing the different feature extraction techniques and their effectiveness in AMT systems. This step is the foundation of machine learning and is therefore of utmost importance.
- Conduct a comparison of how effective different combinations of machine learning architectures and different feature extractions techniques may be in solving this problem. This will help identify the best combination for the study.

Through the accomplishment of the outlined objectives, this study aims to provide a extensive understanding of the machine learning and feature extraction techniques that may be used to solve the AMT problem. This deep understanding of the theory and techniques explored in this project will be used to develop a custom AMT model for monophonic music.

1.4 Scope and Limitations

This project primarily focuses on the transcription of monophonic music using machine learning. Due to hardware, time, and ethical constraints, several compromises were made in order to complete the project in within a reasonable timeframe. While machine learning shows promise for creating highly accurate AMT systems, the computational intensity and lengthy training period required the scope to be scaled back.

1.4.1 Scope

- **Monophonic Music:** Primary focus of the AMT model built will be monophonic piano music (singular melody on a singular instrument)
- **Modular:** The system should allow easy modification to try different models and addition of new features with the aim to have it be ready to transcribe polyphonic music in the future.
- **Dataset:** The datasets used is a will be well documented and will have amply prepared annotations and variability to ensure proper training of the model for AMT
- **Feature extraction:** Different feature extraction techniques and their benefits will be explored.
- **Post-processing:** Explore post processing techniques that may improve the model's performance without needing to train the model further.

1.4.2 Limitations

- **Monophonic Music:** Primary focus of the AMT model built will be monophonic piano music (singular melody on a singular instrument)
- **Dataset Reduction:** A smaller portion of the dataset was used to reduce training time, thus sacrificing potential robustness for time.
- **Data Splitting and Reprocessing:** Data is compressed and batched before being passed into the model to ensure that it does not use too many computational resources when training.
- **Hyperparameter Tuning:** Due to the large amount of time it took to train each model the no hyperparameter tuning was conducted.
- **Survey Metric:** It was thought that the best approach to determining transcription performance would be to collect data on how many people could recognize the transcribed music. However the project was not ethically cleared to do so.

1.5 Thesis Outline

This report consists of six chapters that outline the overall approach taken in addressing the given problem. **Chapter 1** introduces the study, it gives a background to the problem and identifies current technologies used to approach the problem and their limitations. It continues by establishing the objective that this report should achieve before finally declaring the scope and limitations of the project. **Chapter 2** is a review of the literature and concepts that are fundamental to the topic being researched. **Chapter 3** highlights the method, technologies and requirements that are essential to meeting the objectives of the project. **Chapter 4** is an in-depth analysis of how the solution was ultimately implemented. **Chapter 5** is where the performance of the implemented system is analysed in order to determine whether it meets the requirements outlined. Finally **Chapter 6** closes off the paper with a conclusion of what the report has achieved and suggestions on future work.

Chapter 2

Literature Review

This literature review aims to explore the key aspects of automatic music transcription (AMT) using machine learning techniques, providing an understanding of sound, musical theory, digital signal processing, and various model architectures. By exploring the challenges and advancements in AMT, such as feature extraction techniques and machine learning models.

2.1 Sound

Before diving into how AMT works it is important to establish an understanding of how sound which the basis of the entire system is [2]. Sound originates from a vibrating source which could be anything from vocal chords to a guitar string. For the purposes of this project the sounds produced are generated from the strings inside of a piano. Upon being generated these sounds propagate though different media (depending on the media the behaviour and speed of the sound waves changes) by creating areas of high pressure called compressions and areas of low pressure called rarefactions. Since sound travels parallel to the particles of the disturbed medium carrying them it is thereby classified as a longitudinal wave [3].

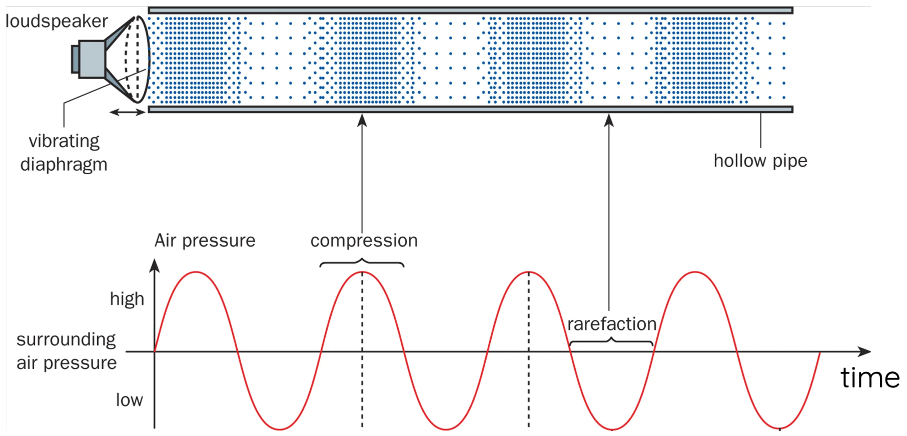

Figure 2.1: Summary of how sound propagates [4]

Sounds fall into two major categories pure tones (single frequencies) and complex sounds which include any sound that is composed of several sinusoidal components at different amplitudes, frequencies and phases [3]. Most sounds in nature are classified as complex sounds, complex sounds can either be musical or non-musical [5]. In order for a sound to be considered as musical it should be made up of a fundamental frequency which is the lowest frequency at which the source of a sound vibrates and it should have overtones known as harmonics which are the integral multiple frequencies of the fundamental frequencies [5]. Sounds containing a fundamental frequency but non-harmonic overtones are not considered to be musical, they are simply classified as noise. Beyond the rather simplistic definition of a musical sound there are more factors that contribute to what makes a sound musical particularly the control and choice of vibrations used to achieve particular pitches, harmonies and rhythms. The pitch of a sound that is perceived is determined by the fundamental frequency of that sound and the method by which this trait may be controlled depends on the instrument, such as vibrating the strings of a guitar while pressing on different frets, or covering of different holes on a clarinet. However when different musical instruments produce the same pitches it can easily be noticed that they still sound

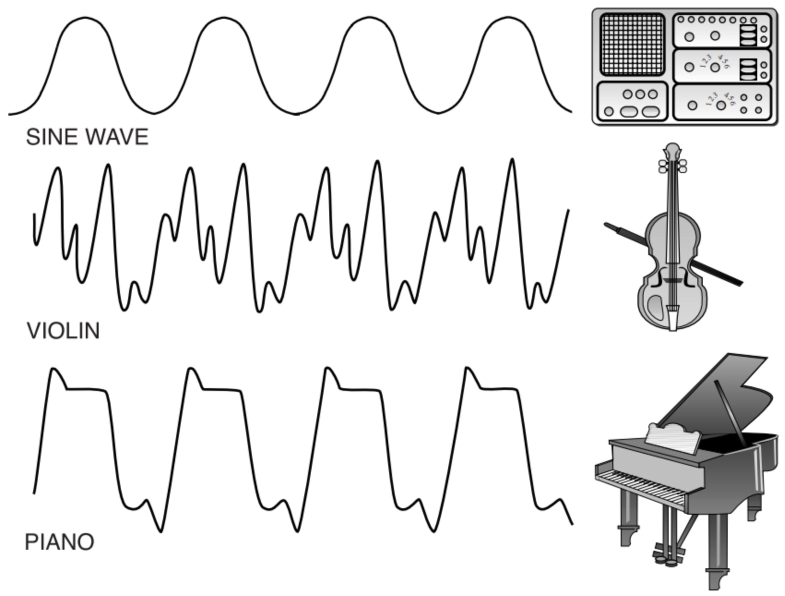

Figure 2.2: Waveform of G4 on a signal generator, violin and piano

different, and this property is known as timbre [6]. The timbre of an instrument is the characteristic sound that it makes even when its pitch and loudness are the same as another instrument and it is due to the method of play, construction and materials used for the instrument [3].

In AMT musical notes are typically identified on the basis of pitch, onset times and offset times. Offset times however are generally difficult to characterise because of their ambiguous nature hence they are often ignored however onset times are paramount to any AMT system and can simply be defined as the position and duration of notes. In AMT onset time is the moment at which the note can be detected [7]. The relevant music theory shall be discussed in the next section.

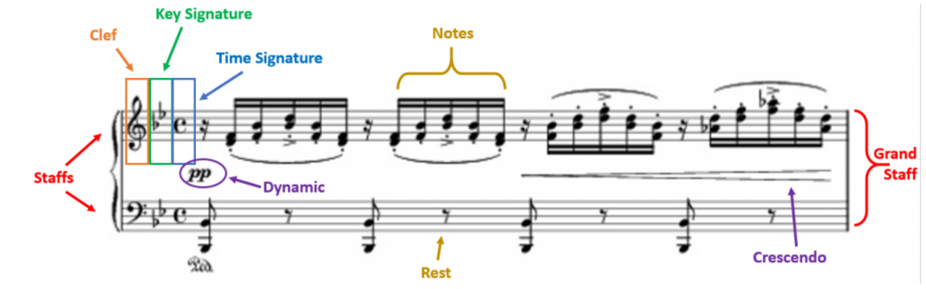

Figure 2.3: Waveform of G4 on a signal generator, violin and piano

2.2 Theory of Music

Musical notation is system that is used to represent music visually through the use of symbols. While musical notation has plenty of nuances that may appear, they are far beyond the scope of this literature review. Therefore the essentials of musical notation will be discussed in this section.

The essentials of musical notation provide the basic framework and structure for any piece of music, but there are some components that must always be present for a score to be considered complete. The Staff is the set of five lines with four spaces that are used for the placement of notes and other musical symbols [3]. Where the notes and symbols are placed on the staff represents the pitch of the note and allows the musician to determine the how high or low the note is [8].

Moving onto what is placed on the staff, there is a clef at the beginning of the staff that serves to indicate the pitch of the notes that are placed on the staff. Furthermore it determines what lines and spaces correspond to a pitch. The most commonly used clefs are the treble clef which indicates the use of higher pitched sounds which on a piano would mean using the right hand side of the piano, and the bass clef which is used for lower pitched sounds which in the case of this report would be the left hand side of the piano [3].

Once the pitch of the piece is determined it is important to determine the key of the piece and this is done using the key signature. The key signature is a set of flat or sharp symbols that are placed directly after the clef at the start of the staff. The purpose of the key signature is to give the key of the music which specifies whether notes are to be played as sharps or flats [9].

The time signature of a piece is placed directly after the key signature and consists of two numbers one on top of the other. The number at the top represents the number of beats in each measure, while the number at the bottom specifies the note value that represents one beat. Having determined the general timing of the piece of music, the notes and rests are now placed into the piece. The notes and rests simply represent the sounds in the music (notes) and the silence (rests) every note and rest has a shape that is indicative of their duration. The notes and rests are what provide the rhythm and melody of a piece by indicating when to play and when to be silent and for how long [3].

2.3 Digital Signal Processing

Audio signals are complex signals that vary over time and frequency hence they are continuous signals meaning that they also contain an infinite amount of data [10]. However this is a problem for a digital device that is expecting a finite amount of information to process. In order for the data collected to be processed and utilised by a digital device it needs to converted into a series of discrete values taken periodically this is known as the digital representation of a signal [10].

2.3.1 Audio Processing

Sampling is the process used to measure the values of a continuous signal at regular intervals and the rate at which these intervals occur is known as the sampling rate. The sampling rate of an audio signal must adhere to the Nyquist-Shannon sampling theorem in order for it to be useful. According to this theorem the sampling rate used must be at least twice the highest frequency of the analogue signal. This done to maintain the integrity of the frequency information(avoid aliasing) [5]. The sampling rate most frequently used for audio recording is 44100Hz meaning the signal is sampled once every 0.023 milliseconds. The reason for this common sampling signal is the fact that the human audible range is from 20 Hz to 20000 Hz thus by Nyquist-Shannon sampling theorem the minimum sampling frequency is 40000 Hz. However the common sampling frequency of 44100 Hz allows space for error.

Having obtained the audio signals there is now a need for them to be analysed, one common way of analysing a signal is the fourier transform. It is used due to the fact that it can represent any finite energy signal as the sum of infinite sinusoidal coefficients weighted using amplitude coefficients [11]. However while the fourier transform is an excellent tool for analysing the frequency content of audio signals it has one major caveat. The fourier transform assumes that a signal is stationary, however this is less than ideal when analysing music signals since they are dynamic [10]. Due to the dynamic nature of music signals it is necessary for the time and frequency to be analysed simultaneously, there are a myriad of techniques that could be used to analyse these signals however three in particular stand out fo the purposes of AMT namely the constant Q transform (CQT), the Mel-Spectrogram and the Mel Frequency Cepstral Coefficients (MFCC).

2.3.2 Constant Q Transform

The constant Q transform is a time frequency analysis method that works on the principle of dividing the frequency axis into a logarithmic scale in which the width of the frequency bins is proportional to their centre frequencies [12]. The reason this system works particularly well in AMT is because the it is logarithmically spaced which is how musical pitch structures are also spaced. The CQT uses a window function to match the the Q factor of the human auditory system which is the measure of the bandwidth of a filter relative to its centre frequency [10].

The CQT has further ingrained its status as a useful tool in AMT systems, because it has been shown to be more efficient than more traditional time-frequency analysis methods such as the short time fourier transform (STFT) [13]. Its success in other sectors of musical analysis such as melody generation [14] further drives this narrative. Therefore given the information in the sources it's clear that with its logarithmic frequency spacing and efficient representation the CQT is a powerful tool for a myriad of musical machine learning tasks including AMT.

2.3.3 Mel-spectrogram

Mel-spectrograms are a common method used to represent audio signals particularly in AMT and music information retrieval (MIR) systems. In literature mel-spectrograms have gained popularity due to several factors such as their ability to capture what is considered to be perceptually relevant information from audio. They do this by approximating human sensitivity to frequency change using the mel scale [15]. Moreover they have been noted as being an efficient means of representing data [16] due to typically having a smaller file sizes than other techniques employed such as the CQT. Therefore they are less computationally expensive to process which can be crucial for deep learning models. Mel-spectrogram being used as input features for AMT system are recognized for being promising for tasks such as polyphonic piano music transcription [17]. With all these factors in mind it is clear that the mel-spectrogram is a strong choice for any AMT system.

2.3.4 Mel Frequency Cepstral Coefficients

The Mel Frequency Cepstral Coefficients are a common feature extraction technique that are obtained by fourier transforming audio signals and mapping the result onto the Mel-frequency scale [9]. The Mel-frequency scale is a non-linear scale that resembles the human auditory system, once the signal is mapped onto the Mel-frequency scale the power spectrum is calculated and processed by means of filter banks that replicate the human ear's frequency response [10] which is a major benefit to feature extraction techniques in AMT.

Despite MFCCs not being commonly deployed in AMT systems they have been shown to be capable of obtaining promising result such as in [9], this most likely due to factors such as their robustness to noise [10] and its highly compact representation which can significantly reduce computational complexity. MFCCs present themselves as promising feature extraction technique with their notable robustness, ability to capture perceptually relevant information from audio and their compact representation of data that could ease the model training process therefore making them a good choice for feature extraction.

2.4 AMT Machine Learning Techniques

In the most basic sense AMT is the conversion of an audio recording into a musical score. This particular task has received significant attention in the field of music information retrieval (MIR) due to the significant ways it can contribute to fields such as education, audio engineering, etc. With the rise of machine learning and artificial intelligence, a variety of new methods have been proposed that may enhance the efficiency and accuracy of AMT systems. Therefore this section of the literature review will explore the state-of-the-art machine learning techniques employed in AMT.

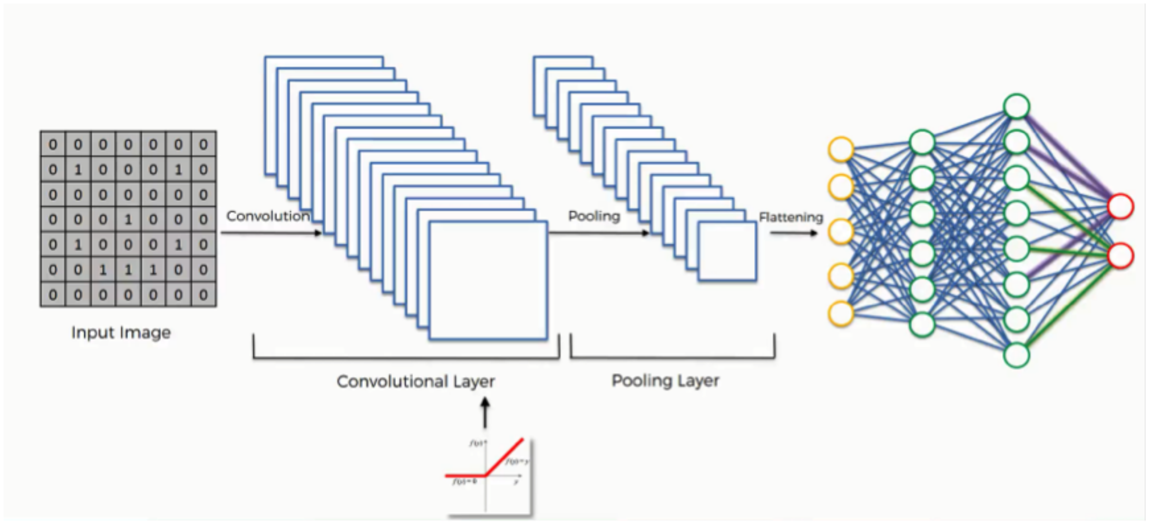

Figure 2.4: Diagram summary of CNN network [18]

2.4.1 Convolutional Neural Networks (CNNs) in AMT

Convolutional Neural networks have gained popularity for feature extraction in AMT due to their ability to detect patterns in audio signals. The typical CNN approach to AMT involves the conversion of audio signals into spectrograms which represent the signal's amplitude across frequencies over time as discussed Section 2.3. The CNNs analyse spectrograms to identify musical characteristics such as the timbre pitch and rhythm of the audio signal. One method used is parallel fusion, this is an approach that employs the use of multiple CNNs to process the different parts of an audio signal in order to capture the various musical features such that each one would focus on a particular characteristic such as the pitch or the timbre and the outputs for each of these CNNs will be fused to each other in order to improve the overall accuracy of the system [19] and this method stands out due to its impressive results.

The results in Table 2.1 display the F1-score obtained for two different data splits on a frame and note level where "I" represents the model tested on a mix of live and

Table 2.1: Transcription Results From [19]

Frame(I)	**Note(I)**	**Frame(II)**	**Note(II)**
77.76	84.16	65.02	68.23

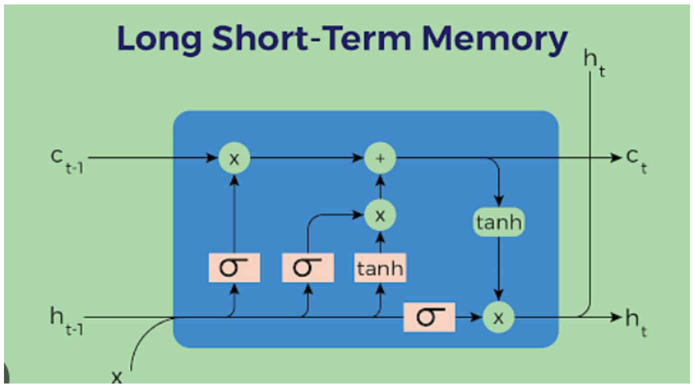

Figure 2.5: Diagram summary of LSTM network [20]

electronically generated piano recordings and ”II” represents the model tested on only live piano recordings. These results demonstrate just how capable CNNs can be at recognising features and their method of use in the system demonstrates their versatility which serves as further argument for their implementation into an AMT system.

2.4.2 Recurrent Neural Networks (RNNs) in AMT

Recurrent Neural Networks are designed to recognise patterns in sequences of data as such they are an excellent choice for handling the temporal dynamics of a musical progression. One type of RNN is the Long Short-Term Memory (LSTM) Network which can retain information over extended sequences, which is essential in music transcription.

LSTM networks are able to store and retrieve information over time using structures known as memory cells and gates (input, output, and forget gates). In AMT LSTM networks are used to allow the system to develop an understanding of the structure of musical sequences, and to predict future notes based on previously played notes [13]. The nature of these networks makes them ideal to be combined with a CNN in order to compensate for the CNN's lackluster performance when handling the temporal dynamics of a provided spectrogram as noted in [15]. The strength of LSTMs is shown in [21] as can be seen in the results of their proposed solution.

Table 2.2: Transcription Results From [21]

Accuracy	**Precision**	**Recall**	**F1-score**
94.64	96.30	95.00	95.60

However it should be noted that the testing and training were all conducted on Carnatic music as a result the consistency in the style and nature of this type of music could easily have contributed the the particularly high scores obtained by the model. This could potentially mean that the model will perform poorly given another style of music.

Long Short-Term Memory Networks however can be quite computationally heavy as such authors proposed the use of a streamlined version of these systems called Gated Recurrent Units (GRUs) they offer greater computational efficiency and speed by means of using fewer parameters without majorly impacting system performance [14]. Overall LSTMs have shown throughout literature that their ability to understand the temporal nature of music makes them a suitable choice for any AMT system.

2.4.3 U-Net Architecture in AMT

The U-Net architecture is a CNN based technique that was developed for segmentation of images in the biomedical field, however some authors have managed to adapt it to the AMT field and have found successes in using it [15]. In the context of AMT this technique is used to treat an audio signal as an image and segment it into musical components. This system has found popularity in this field primarily due to its ability to handle temporal and spectral features of music along with its robustness to noise hence studies have often found it to achieve high accuracy in several conditions. The U-Net architecture is modeled using an encoder-decoder structure with skip connections. The encoder compresses data to allow the capture of only essential features of data and the decoder takes the data back to its original size, using the learned features to recreate the musical elements of an audio signal [15]. Skip connections between the linked layers of the encoder and decoder enhance the model's ability to utilized detailed information thereby improving note segmentation and transcription accuracy.

2.4.4 End-to-End Systems In AMT

End-to-End Systems are a type of deep learning model that is able to take raw audio as an input and is able to output musical notation without the need for manual feature engineering, thereby simplifying the transcription process. Not much has been researched for this method however such a system would prove to be useful in the field of AMT.

2.4.5 Challenges in AMT

While machine learning has advanced significantly with the advent of machine learning techniques it is not perfect as such the challenges of AMT systems must also be explored. In the current state of the field polyphonic music transcription is commonly noted as the primary problem with AMT systems, this is due to the fact that when multiple notes are played simultaneously there is an overlap of frequencies and harmonics which will often lead to inaccuracies in a transcription model as such highly advanced models need to be employed to overcome this problem. Another caveat of machine learning models in general is the high volume of data that they need for them to effectively work, in the case of this this report this data need to be plentiful and diverse as such it should include different music genres, recording conditions as this increases the models robustness and ability to generalize.

2.4.6 Summary

The literature review traced the evolution of automatic music transcription (AMT) and highlighted some of the main challenges in the field. It covered important concepts such as sound properties, music theory, and key digital signal processing techniques like the Fourier Transform, Constant Q Transform (CQT), mel-spectrogram, and Mel Frequency Cepstral Coefficients (MFCC). These tools, combined with machine learning architectures such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and U-Net, have advanced our ability to extract musical features and interpret the temporal patterns of music. The next chapter builds on the insights from this chapter to propose an effective approach towards AMT.

Chapter 3

Methodology and Design

This chapter of the report discusses the approach taken to develop a piano note classification system using different neural networks by means of leveraging machine learning techniques and digital signal processing where necessary. This chapter begins by deconstructing the overall problem to be solved into its constituent parts in order to simplify the approach, building off of that information the user requirements are extrapolated in order to determine the specifications that need to be met in order for this system to be considered a success. Finally a high-level outline based on the aforementioned specifications of the system is given.

3.1 Software Development Pipeline

The goal of this project is to create an AMT system with a focus on monophonic piano music, having explored the various approaches taken in the literature reviewed in Chapter 2 and drawing inspiration from the many architectures and techniques discussed in these sources, it is evident that based on the chosen approach to utilise neural networks to solve this problem there are certain mandatory steps that must be implemented within the code to take the system to completion. It should however be noted that study of the various literature also showed non-essential but helpful techniques to take into consideration such as the thresholding as suggested by [13] to combat false positives, hence based on all this information the overall task was broken down into the following tasks:

1. Audio Preprocessing
2. Feature Extraction
3. Neural Network Architecture
4. Model Training
5. Postprocessing
6. Transcription
7. Evaluation

Each of these features may be developed separately and combined in the final stages to create one complete system. The separation of these tasks in the initial development of the system also ensures that modifying the model to use different machine learning techniques will be an easier task.

To tackle these tasks an agile approach is taken, due to its circular development ideology which is ideal for this project due to how many elements the overall system consists of [22]. This means that each component will have its requirements clearly defined to allow a more methodical and targeted approach to each problem and from these requirements a set of specifications and acceptable test criteria are outlined to benchmark the performance of each individual components.

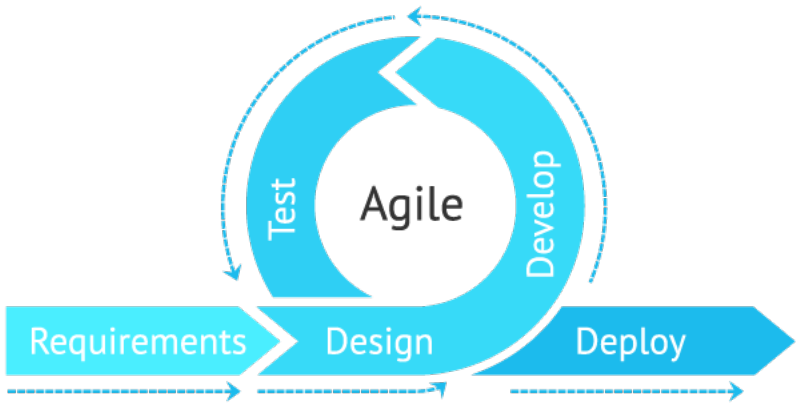

Figure 3.1: Agile software development structure [23]

With the requirements of the software having been defined they are then used as a guideline to the software design process, ensuring that each of the acceptance criteria are met in regards to the design before the module being worked on is completed.

Having designed the structure of the software to suit the requirements and having defined clear acceptance criteria, the most difficult part of development may now begin. This part of the development cycle is where all the functional requirements of the software are to be met to ensure a satisfactory final product, it goes hand-in-hand with the testing phase in order to determine if the system is ready for deployment in the context of this report after the testing phase is complete for a component of the full system the next one is designed developed and tested and once all individual components are completed they are integrated to allow for one cohesive system that looks and performs according to the acceptance criteria.

3.2 Requirements

In order to properly address this problem, it is essential to establish a set of standards that the completed system must meet. These requirements serve as a guiding framework, ensuring that the development process from concept to fully realized model is done diligently.

3.2.1 User Requirements

The utmost goal of this project is to create a system that can take an audio file as an input and is able to transcribe it into a musical score that will allow a musician to reproduce the original piece. The best case scenario for a project of this nature would be to produce a score that when played sounds identical to the original audio file in all aspects, this doesn't necessarily mean that the output score must match the input score perfectly since scores that have minor differences could sound similar or even the exact same score could be played slightly differently due to the interpretations of the musician as noted by [24]. Moreover since traditional metrics struggle to grasp the perceptual quality of the outputted transcriptions [25] it is important to use a more perceptual metric to determine the output quality. To achieve the desired system it is important to first outline the basic expectation of this system in the form of user requirements.

UR-ID	**Piano Note Classification User Requirements**	**S-ID**
UR-01	The model system must handle audio data and corresponding annotation files	S-01, S-02, S-03
UR-02	Preprocess audio into a form that the model can interpret	S-03, S-04
UR-03	Model should hand multi-pitch, onset and offset detection	S-06, S-12, S-07
UR-04	Model must output data quickly	S-05, S-13
UR-05	System must evaluate transcription accuracy using multiple metrics	S-08, S-09
UR-06	System must produce quality transcription	S-10, S-11, S-14

Table 3.1: AMT System User Requirements

3.2.2 Functional Requirements

The user requirements create something of a basis for the necessary functionality of the piano note classification model and it is from this basis that the specifications of the AMT system may be derived.

S-ID	Piano Note Classification Specifications	A-ID
S-01	Model must support WAV files for optimum quality	A-01, A-03
S-02	Model must support MIDI, TSV and CSV annotation files	A-02
S-03	Ability to preprocess data into mel-spectrograms, CQTs and MFCCs	A-03, A-04
S-04	Data Normalisation for input features	A-04
S-05	Batching for large data inputs	A-05
S-06	Utilise DNN, CNN and LSTM model architectures	A-06, A-07
S-07	Train Model with labeled onset, offset and MIDI pitch Data	A07
S-08	Measure precision Recall and F1-score of transcriptions	A-08
S-09	Implement visual onset check for manual transcription quality measure	A-09
S-10	Accurate tempo	A10
S-11	At least 60% true notes must be found	A-11
S-12	Combined loss function for onset, offset and pitch must be used	A-07, A-12
S-13	songs of average length must be generated within a few minutes	A-13
S-14	Implement secondary onset filter to refine predictions	A-08, A-11

Table 3.2: AMT System Specifications

3.2.3 Acceptance Test Procedures

The specifications give clear guidance as to exactly how the system is supposed to work. However going beyond having a system that works it is important to have a system that works well. This is the reason behind acceptance test procedures they are meant to set the standard of performance expected form the built system. The standards for the AMT system being built are as follows.

A-ID	**Piano Note Classification Acceptance Test Procedures**
A-01	Inputted WAV files must be used to create output MIDI file
A-02	NumPy array for ground truth must be produced from annotations
A-03	WAV file must have features extracted into NumPy array
A-04	Min-Max scaling correctly applied on extracted features
A-05	Large inputs of data must be split to save space
A-06	Build and train DNN,CNN and LSTM models
A-07	Code is optimised to extract his data from provided annotations
A-08	Obtain an F1-score that is better than in initial testing phase
A-09	Plot graphs on prediction vs actual with overlap
A-10	Tempo must be double, half or within 5BPM of true tempo
A-11	Minimum recall score of 60%
A-12	Final model must be capable of detecting onsets offsets and pitch
A-13	A 3 minute song must be generated within a 60 seconds
A-14	Post-processing improves results of transcription

Table 3.3: AMT System Acceptance Test Procedures

3.3 Software Choices

With the abundance of possible solutions to this problem it is important to decide which one is able to provide the greatest benefits with the least drawbacks. For the purposes of AMT the best choices were concluded to be Python, C++, Java and JavaScript. The most important aspects of these programming languages in regard to this project will be ranked on a scale of 1 to 5 (in which a 1 indicates weak performance or a lack of necessary tools and support, while a 5 indicates strong performance and an abundance of tools and support) and their weighted score will determine which language is best suited for the purposes of this project.

Based on the scores from the above table it is clear that python offers the greatest advantages to disadvantages ratio. This is because python has an abundance of

Feature	**Python**	**C++**	**Java**	**JavaScript**
Speed	2	5	4	3
ML Library Support	5	3	3	4
Music Notation Library Support	5	3	3	3
Audio Signal Library Support	5	2	3	3
Easy of use	5	2	3	4
Total	22	15	16	17

Table 3.4: Feature Comparison Table

well documented and regularly updated libraries for machine learning, audio signal processing and musical notation all of this combined with the comparatively simple syntax and readability that python offers all combine to make python the ideal language for this project despite it's slower speed which for the purposes of this project is still fast enough. With the abundance of Python libraries available for the different aspects of theis project i.e. audio analysis, music analysis and machine learning the libraries Librosa, pretty_midi, pandas and TensorFlow respectively are chosen due to their popularity in literature and their thoroughly detailed documentation. All code for this project was done in a Visual Studio Code using the python notebook format to allow for each component of the code to be tested independently and to allow for easier interaction with visualised data. Muse score was utilised to manipulate and examine MIDI files due to its popularity and suite of features.

3.4 Software Structure

With the various choices and requirements of the system clearly defined, it is now essential to establish the necessary structure that will be utilized during the implementation phase. This structure will guide the development process, ensuring that all components of the system are integrated effectively and function as intended.

3.4.1 Training Algorithm

The first step to any machine learning model is developing a suitable process through which data may be prepared and the models may be developed through training. For the purposes of this project it should be noted that three means of feature extraction and three types of neural networks were used making for a total of nine different models. The general training algorithm is as shown in Figure 3.2

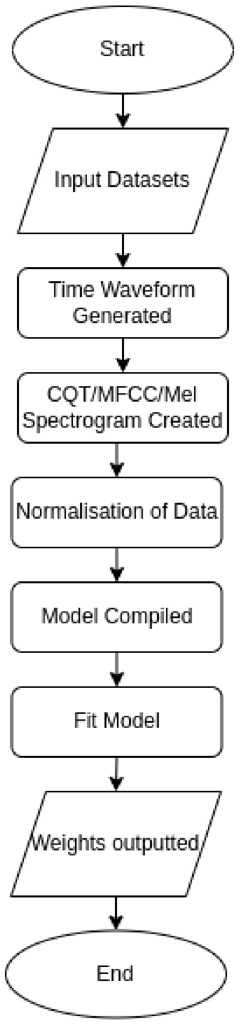

Figure 3.2: Training Algorithm Flowchart

3.4.2 Prediction Algorithm

Having completed training of the Neural Networks a structure for the general use of the system must now be defined for the end user. The flowchart in Figure 3.3 gives an overview of how the final system will operate.

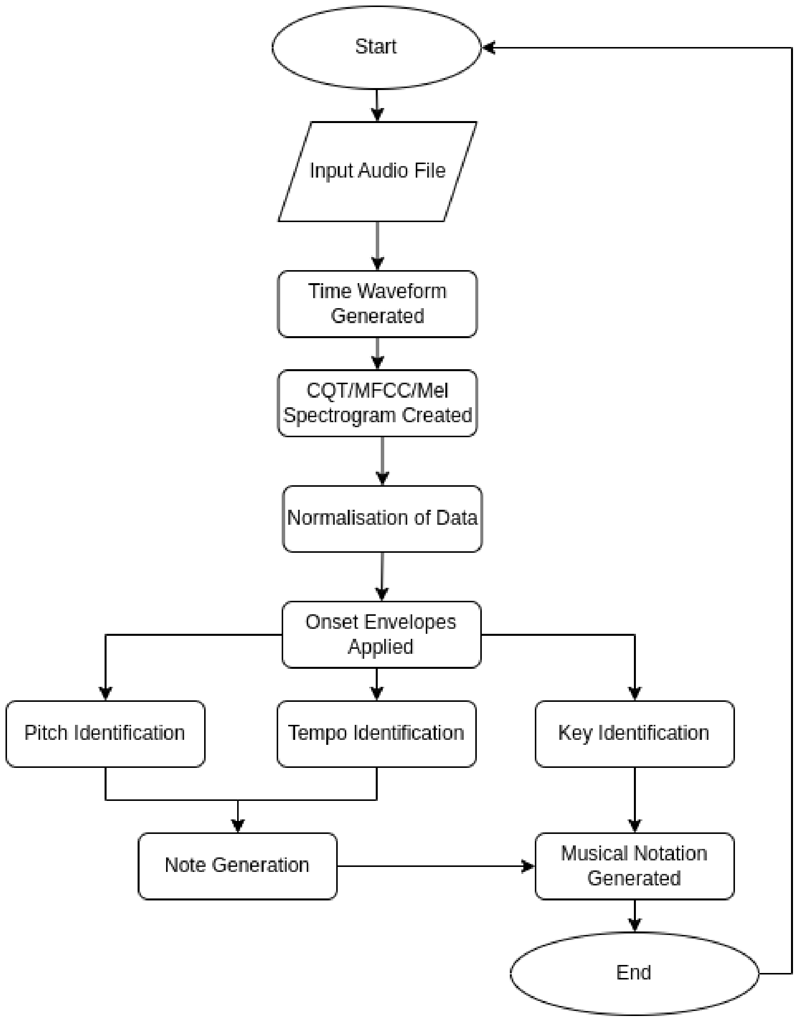

Figure 3.3: Prediction Algorithm Flowchart

As demonstrated by the diagram an input audio file is loaded into the system, once it has been loaded into the system the waveform of the audio file is generated to be used as input before feature extraction to ensure an accurate representation of the audio file [26]. After the waveform is generated feature extraction using a selected method from CQT, MFCC and Mel Spectrograms is completed and the data is stored. The stored data is then normalised to ensure consistent scaling across the data as suggested by [16] to mitigate issues arising signal amplitude variation now that the features are amply prepared they are fed into the system which then uses its trained weights to predict where the onsets of the notes are and from this information it can then determine the tempo, pitch and key of a piece and uses this information to generate a musical score for the end user.

3.5 Chapter 3 Summary

This chapter acted as an overview to the approach that will be taken in implementing the final model. It gave a clear rationale as to why the decisions in software and tools were made and highlights the approach taken towards solving the individual problems that constitute the overarching AMT problem. Therefore it has provided the foundation on which the final implementation will be built in Chapter 4.

Chapter 4

Implementation

The implementation of this piano note classification model is built on several key stages as outlined in Section 3.1 that transform the raw audio that is inputted to the system into a symbolic representation of the music in the form of a MIDI file. This system utilises deep learning techniques to detect and correctly identify musical notes from monophonic audio and the use of further processing in order to refine predictions made on the initial output. The final output of the system is a MIDI file that may be used for playback, as guidance for a musician on how to play the piece or even just further analysis to determine where the model is lacking and may be improved.

This section outlines the step-by-step implementation process, starting from the initial preprocessing and feature extraction, followed by the development and training of the of the different neural networks that were used for note detection and identification. Additionally it will delve into the post-processing techniques that were used in an attempt to improve transcription accuracy. Finally the process of converting the model's predictions into a MIDI file, that attempts to accurately recreate the original audio is detailed. Overall this section provides a comprehensive overview of the methods and theory used in the development of this piano note classification model.

4.1 Data Preprocessing and Feature Extraction

The dataset used for this project is the *MIDI Aligned Piano Sounds (MAPS)* dataset. Due to the limited computing resources and time available only a section of the dataset was utilised particularly the musical pieces denoted by folders labelled ”MUS”. All the required files from the ”MUS” folders in the different denominations, the first step to this process was to separate the data into 3 sets namely, training, testing and validation data [27]. The training, testing and validation splits are 70%, 20% and 10% respectively. Selection of the data for each split was done at random to ensure a variety of song lengths, and recording conditions all become a part of each split.

Once the dataset has been split into the necessary categories the features of the WAV files may be extracted, these features act as training data because they carry vital information that may be used to identify individual notes of a musical piece. The Constant Q Transform (CQT), Mel-Spectrogram and Mel-frequency Cepstrum Coefficients are the three different methods used for feature extraction that are proposed as it is a well suited time-frequency representation of musical signals due to it's frequency axis being linear in pitch and lower dimensionality as compared to other methods [13]. All audio files were converted from stereo to mono by calculating the mean of the two channels as this simplifies the feature extraction process, once calculated their constant Q transform, Mel-Spectrogram or MFCC features are extracted from the mean audio signal. The CQTs, MFCCs and Mel Spectrograms are all configured to cover 7 octaves or more which is in line with the number of octaves a standard 88 key piano can produce sounds from [6].This is done to ensure that the CQT can capture a wide range of frequencies thereby helping it accurately detect pitches from the piano music including the subtle nuances between pitches that may be close in frequency value.

Listing 4.1: Feature extraction code for Mel-spectrogram

```python
try:
        sampling_freq, stereo_vector = wavfile.read(wav_path)
except Exception as e:
        print("Wave file error:", wav_path, str(e))
        continue

stereo_vector = stereo_vector.astype(np.float32) / np.max(np.
    abs(stereo_vector))

mono_vector = librosa.to_mono(stereo_vector.T)  # Converts
    and normalizes the signal

mel_spectrogram = librosa.feature.melspectrogram(y=
    mono_vector, sr=sampling_freq, n_fft=2048,
hop_length=hop_length, n_mels=n_mels)
mel_spectrogram = librosa.power_to_db(mel_spectrogram, ref=np
    .max).T
```

As the code extracts the features from an audio file it finds the corresponding CSV file containing information on the onset, offset and pitch at any given time in the audio file is used as a ground truth label that is aligned to the features as they are extracted. The total number of possible pitches is 88 as this is the standard number of keys on a standard piano, thus the label per frame can be represented as a vector of dimension 88, this means that the pitched played in a frame is set to one while the rest are set to zero. At the end of this process there should be a file containing CQT features that has a matching label file.

4.2 Normalisation

After the initial preprocessing has been handled the data must be normalised, in this case min-max scaling is utilised which scales values to a range of 0 to 1.

$$X_{\text{scaled}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{4.1}$$

Normalisation in this case is done in order to ensure that all features contribute equally to the model as opposed to having features with larger values dominate the training process [28], this is especially important for audio/spectrogram data due to the fact that they may overfit or become biased towards parts of the data with large values if it is not normalised. Moreover the use of a consistent normalisation method on all the data categories may improve generalisation and finally normalisation ensure that the data is centered around zero which makes it easier for neural networks to learn patterns [29].

4.3 Network Architecture and Training

This project approaches the task of piano note classification through the use of different neural network architectures namely Convolutional Neural Networks (CNNs), Deep Neural Networks (DNNs) and Long Short Term Memory networks (LSTMs). Each model has been configured with the goal of effectively learning from the training data to provide accurate note and timing predictions. All models have been implemented using the Keras deep learning library and tensorflow as the backend.

To ensure that comparisons of the systems are as fair as possible each one has been configured to roughly the same parameters and the same amount of training which in this case means that each neural network consists of four layers with 256 units in each layer, they use an adam optimiser due to its efficiency in speeding up the training process by means of accelerating stochastic gradient descent as suggested by [16] and it is also capable of adjusting the learning rate to which may help the model learn to find complex solutions [30], all hidden layers activation is configured to ReLU to enhance computational efficiency in this highly complex task [31] and their output activation is configured to sigmoid due to its non linearity which allow the networks to learn more sophisticated patterns [32] with drop out rates of 20% to prevent overfitting [33] since this is a multi-label classification with continuous valued outputs it was determined that mean squared error should be employed as the loss function [25]. Due to the time constraints during this project hyperparameter tuning was not used as a means of improving results thus across all experiments consistency is maintained for the hyperparameters.

4.4 Post-Processing

The post-processing in this code refines predicted note events by combining the model's pitch predictions with onset detection performed using Librosa. Onset detection, identifies the times when note onsets occur in the audio. These detected onset times are critical for aligning the note predictions with actual musical events. By ensuring that predicted notes appear only near these onsets, the code filters out false positives, allowing only valid notes that are temporally aligned with real onsets.

The frame-based pitch predictions are processed to detect whether notes are active (value > 0.5) or inactive (value < 0.1). For each predicted note, it is validated by checking if its start time aligns with a detected onset using a 0.05-second window as the threshold [34]. If the note meets this condition, a note object is created and added to the MIDI track. Additionally, the duration of each note is limited by the prediction threshold, and notes shorter than 0.02 seconds are discarded. This post-processing step ensures that predicted notes are musically relevant, as they are tied closely to real onsets, improving the accuracy and reducing false note activations.

4.5 Transcription

The transcription process for any file begin with the audio file to be tested being converted into the appropriate feature extraction representation such as CQT, mel-spectrogram or MFCC upon being converted the features are exported as a NumPy array file. The exported NumPy array file is then normalised using the technique described in Section 4.2 and is then exported again.

Considering that the audio has been amply prepared for the transcription process it is now loaded into the code along with the appropriate model for the chosen feature extraction method, the model is then used to predict the likely pitches, onsets and offsets of the notes. Once the predictions have been completed the predictions are exported as a NumPy array in which a column represents a time step and each row represents a pitch. The predictions are loaded into the model along with the original wav file, the prediction (matrix) values are used to indicate the probability of note activity for each pitch at each time step.

The wav file is used to estimate the tempo of the piece in beats in per minute by detecting the position of beats in the audio as frame indices then the time values for the first and last beats are used to calculate the tempo.

Listing 4.2: Tempo calculation code

```
def calculate_tempo(y, sampleRate):
        tempo, beats = librosa.beat.beat_track(y=y, sr=
           sampleRate)
        first_beat_time, last_beat_time = librosa.
           frames_to_time((beats[0], beats[-1]), sr=
           sampleRate)
        tempo_calculated = 60 / ((last_beat_time -
            first_beat_time) / (len(beats) - 1))
        if np.isnan(tempo_calculated):
                tempo = 120  # Use a default tempo value
        else:
                tempo = round(tempo_calculated)
        while tempo > 200:
                tempo //= 2
        return tempo
```

The formula used assumes that the tempo of the piece is constant, and further aims to correct irregularities by keeping the tempo limited to a maximum of 200 BPM. This is to prevent issues arising from tempo overestimation which is commonly caused by audio artefacts.

Now that the tempo of the piece has been calculated a new MIDI file is initialised for the track. The tempo is added to the track, then the note information is extracted from the prediction matrix to generate the MIDI notes. For every column of the transposed matrix, the model checks for the onset and offset of each note, this is done by comparing the value of the prediction at each time step to the threshold values highlighted in 4.4 i.e a value>0.5 is considered to be an onset and a value < 0.1 is considered to be an offset. Therefore when a note's onset is detected, its corresponding start time is recorded and when the note's offset is detected its corresponding end time is recorded, and the duration of the note is calculated. As a measure taken against false positives only notes longer than 0.02 seconds are taken into account. After this information is produced the post processing method described in 4.4 is used to refine the predictions and MIDI file upon completion of this process two files are outputted from the final prediction data, a text file containing the start time, end time, and pitch of each note and the MIDI file containing the corresponding MIDI notes. The text file provides a human-readable version of the note data, while the MIDI file can be used for playback in MIDI-compatible software.

4.6 Evaluation

The process of calculating the evaluation metrics (precision, recall and F1-score) begins with the conversion of the original MIDI notes and the predicted notes from the model into frame-based arrays. These arrays represent whether a pitch is active in each frame. The 2D arrays generated are then flattened into 1D arrays where each value is a representation of whether a particular note is currently active or not in a frame. using these flattened arrays the precision, recall and F1-score are calculated by comparing the predicted notes to the ground truth thereby giving a detailed assessment of the models performance.

Listing 4.3: Tempo calculation code

```
#convert notes to frame-based arrays
original_frames = notes_to_frames(original_notes, num_frames,
    frame_duration)
predicted_frames = notes_to_frames(instrument.notes,
   num_frames, frame_duration)

y_true = original_frames.flatten()
y_pred = predicted_frames.flatten()

precision, recall, f1_score, _ =
   precision_recall_fscore_support(y_true, y_pred, average='
   binary')
accuracy = accuracy_score(y_true, y_pred)
```

4.7 Summary

This chapter has given a provided a detailed walkthrough on the rationale behind how the final system was implemented in accordance with the requirements and specifications outlined in Chapter 3. The focus may now shift towards how the culmination of these techniques and choices performs. Therefore the model they resulted in will be evaluated in the next chapter to determine if it satisfies the defined acceptance test procedures.

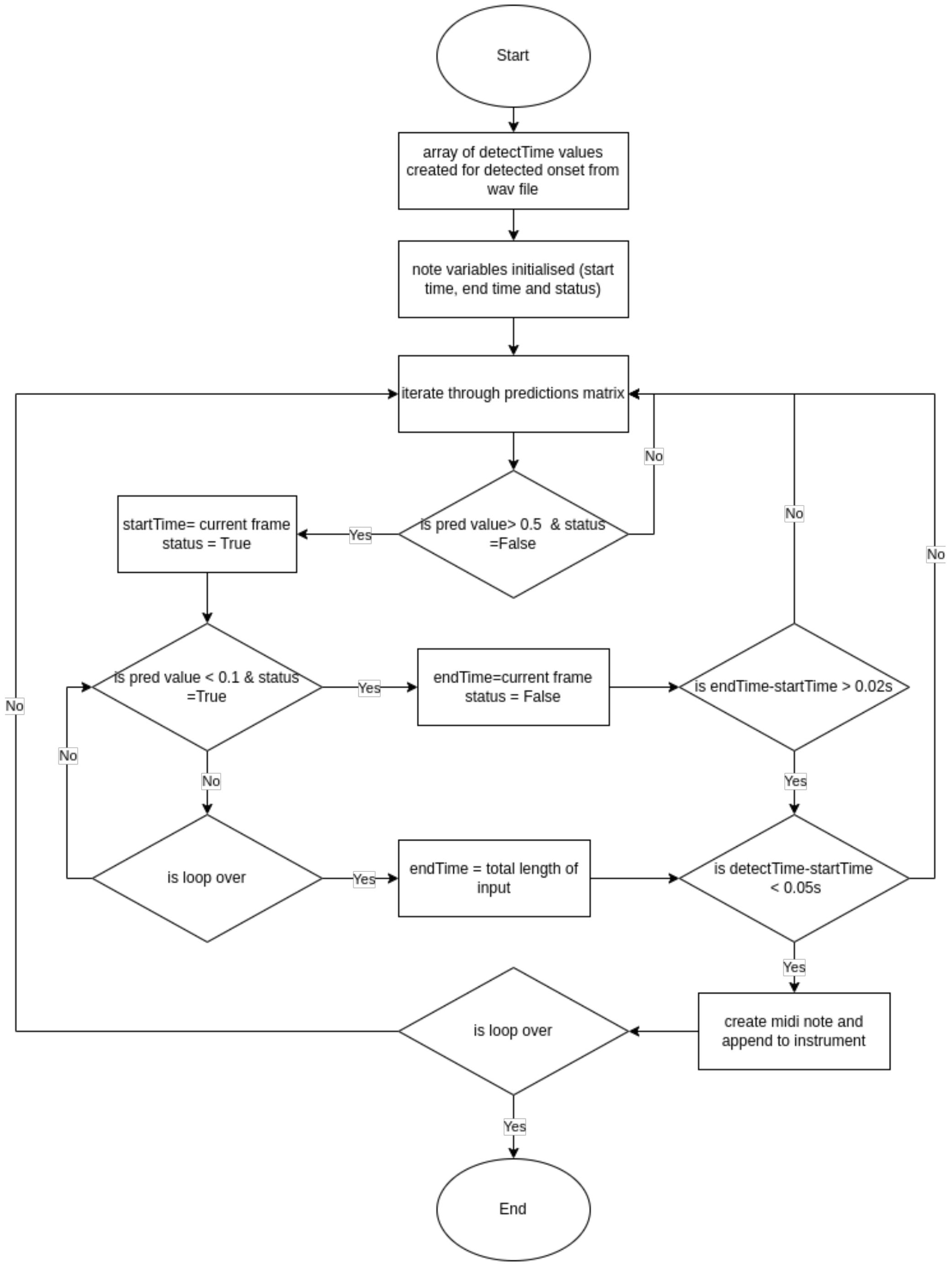
Start
array of detectTime values created for detected onset from wav file
note variables initialised (start time, end time and status)
iterate through predictions matrix
is pred value> 0.5 & status =False
Yes
No
startTime= current frame status = True
is pred value < 0.1 & status =True
Yes
No
endTime=current frame status = False
is endTime-startTime > 0.02s
No
Yes
is loop over
Yes
No
endTime = total length of input
is detectTime-startTime < 0.05s
No
Yes
create midi note and append to instrument
is loop over
No
Yes
End

Figure 4.1: Post-processing algorithm flowchart

Chapter 5

Results

Having developed a fully realised system it is now important to determine how it performs, therefore this chapter will begin by outlining the methods of testing utilised, thereafter The results of the tests will be presented and discussed before finally cross-checking the results against the requirements that were outlined in Chapter 3.2 to confirm that they have been met.

5.1 Test Procedure

The models will each be tested on the testing data that was set aside as mentioned in Chapter 4.1 the performance will be recorded for both the initial predictions and the predictions after post-processing. The testing dataset consists of 54 songs that provide a variety of recording conditions from clean electronically generated WAV files to Live recordings of songs. It also provides a variety of song complexities and tempos as such the results from these initial tests should provide a decent benchmark for how the model will perform in general. Following this the models will be tested on one of the six songs listed below to determine how well they are able to generalise. This process will be used to narrow down the models to the overall best performing model.

Once the best performing model has been determined, it will be used to conduct the tests using the remaining songs. As previously noted in this report, while the traditional metrics used to analyse AMT systems are informative, they are not sufficient enough to be able to fully determine how good a transcription is or how suitable a model may be for AMT. Therefore after the best model has been confirmed qualitative metrics such as spectrogram and musical score comparison will be used to further determine the quality of the model's outputs.

Table 5.1: Ground Truth data for for songs

Input	**Tempo (BPM)**	**Number of notes**
Silent Night - Franz Xaver Gruber	100	74
Hedwig's Theme - John Williams	190	87
Fur Elise - Ludwig Van Beethoven	120	131
Golden Hour - JVKE	189	384
From The Start - Laufey	150	542
Stay With Me - Sam Smith	84	374

Each of the songs used will have a MIDI file that will act as the ground truth. The MIDI files also provide a means of generating clean WAV files that will act as input to the models. ”Fur Elise” and ”Hedwig's Theme” were selected in order to test the model under the conditions it was optimised for, which in this case is monophonic piano music. ”Golden Hour” was chosen as an input to test how well the system transcribes at a tempo above 150BPM. ”From The Start” was chosen to test how well the system can handle a polyphonic piano song (two melodies on a single instrument in this case). ”Stay With Me” was chosen to test how well it can transcribe a song consisting of mainly chords with minor polyphony. Songs 4 to 6 are meant to be a form of stress testing for the model to determine how well it can transcribe under conditions it was not specifically built for. All MIDI files were sourced from

When selecting metrics by which the success of the system it is important to note that in AMT there are no standardised metrics that are capable of capturing the musically relevant aspects of a transcription [35]. With this in mind the metrics used are the Precision, Recall and F1-score as suggested by [36] as well as several other authors. These metrics are dependent on the concepts of true positives(TP), false positives(FP), false negatives(FN) and true negatives(TN) [11] as these summarise the performance of a classification thereby making it easier to calculate the chosen metrics, their structure can be summarised as shown in table 5.2

Table 5.2: Classification Matrix

	Predicted Positive	**Predicted Negative**
Actual Positive	TP	FN
Actual Negative	FP	TN

In the context of this classification model, true positives are the frames where both the predicted note and the actual note are active. False positives are the frames where the system predicted a note, but there was no actual note and false negatives are the frames where there was an actual note, but the system failed to predict it. true negatives are the frames where both the system and the ground truth indicated silence.

Precision acts as a measure of the proportion of correctly predicted notes among all of the predicted notes, thus indicating the system's ability to avoid false positives.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{5.1}$$

Recall acts as a measure of correctly predicted positive notes out of all the notes predicted as positive, meaning high recall indicates a low rate of false negatives.

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{5.2}$$

F1-score acts as the harmonic mean of the precision and recall, therefore providing a measure of the overall accuracy of the system

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5.3}$$

All the results will be presented in tables in Section 5.2, After which the system performace will be discussed. Finally the acceptance criteria given Chapter 3.2 will be checked against the results to determine how well the system has performed in comparison to the outlined goals.

5.2 Experimentation Results

The AMT system developed has been designed to transcribe monophonic piano music by detecting onsets, offsets, and by predicting pitches. The system converts audio in the form of WAV files into musical scores. To facilitate this the audio is preprocessed and then uploaded into the model to produce an output. Excerpts in the form of musical scores from these outputs will be available in Appendix A. It should be noted that all scores shown are initial results from them model with no post processing applied unless indicated.

5.2.1 Benchmarking

Following the testing procedure from Section 5.1, the test set of data is prepared. All the songs contained in the set are preprocessed to produce a numpy array for the data and a corresponding numpy array containing all ground truth data for it. Table 5.3 is a collection of the relevant scores obtained by each model tested using this dataset after training was completed.

Table 5.3: Benchmark Performance

Model	**Precision(%)**	**Recall(%)**	**F1-Score(%)**
CQT DNN	86.46	65.33	74.43
CQT CNN	84.84	55.24	66.91
CQT LSTM	82.86	61.66	70.71
MFCC DNN	30.54	1.00	1.80
MFCC CNN	31.32	0.14	0.28
MFCC LSTM	31.83	1.33	2.55
Mel-Spectrogram DNN	6.87	4.66	5.55
Mel-Spectrogram CNN	79.26	48.43	60.12
Mel-Spectrogram LSTM	84.65	66.43	74.44

5.2.2 Initial Song Test

As each model was designed for the purposes of transcribing monophonic piano the selected piece is ”Silent Night”. This song was used because it is a relatively simple song with one core melody that should be able to effectively gauge how well these models perform in what should be an ideal situation for them. The relevant scores for their transcriptions are shown in table 5.4.

5.2.3 CQT DNN Monophonic

The results of the CQT DNN for all six test songs will be summarised in the tables below. In addition to the metrics from the previous subsections the tempo

Table 5.4: Real World Performance

Model	**Precision(%)**	**Recall(%)**	**F1-Score(%)**
CQT DNN	52.11	94.35	67.14
CQT CNN	39.89	65.47	49.57
CQT LSTM	37.51	88.12	52.62
MFCC DNN	30.54	1.00	1.80
MFCC CNN	31.32	0.14	0.28
MFCC LSTM	31.83	1.33	2.55
Mel-Spectrogram DNN	2.45	3.05	2.72
Mel-Spectrogram CNN	5.86	6.89	6.31
Mel-Spectrogram LSTM	13.89	9.91	11.56

and note count will be added to the metrics.

Table 5.5: CQT DNN Silent Night Performance

Metric	**Before Post-processing**	**After Post-processing**
Precision (%)	52.11	88.20
Recall (%)	94.35	82.59
F1-Score (%)	67.14	85.30
Tempo (BPM)	190	190
Note Count	244	77

Table 5.6: CQT DNN Hedwig's Theme Performance

Metric	**Before Post-processing**	**After Post-processing**
Precision (%)	34.87	75.42
Recall (%)	90.92	79.29
F1-Score (%)	50.41	77.31
Tempo (BPM)	120	120
Note Count	687	168

Table 5.7: CQT DNN Fur Elise Performance

Metric	**Before Post-processing**	**After Post-processing**
Precision (%)	4.42	5.28
Recall (%)	16.14	10.03
F1-Score (%)	6.94	6.92
Tempo (BPM)	190	190
Note Count	287	118

5.2.4 CQT DNN Stress Tests

While the was specifically built for monophonic music and will be critcally analysed on its monophonic music transcription performance, it could be valuable to know how the model performs in conditions it wasn't specifically built for. The results from the stress tests are as follows.

Table 5.8: CQT DNN Golden Hour Performance

Metric	**Before Post-processing**	**After Post-processing**
Precision (%)	23.26	26.37
Recall (%)	98.19	93.34
F1-Score (%)	37.61	41.13
Tempo (BPM)	190	190
Note Count	1037	854

Table 5.9: CQT DNN From The Start Performance

Metric	**Before Post-processing**	**After Post-processing**
Precision (%)	26.27	34.65
Recall (%)	77.80	70.79
F1-Score (%)	39.28	46.53
Tempo (BPM)	150	150
Note Count	2815	1581

Table 5.10: CQT DNN Stay With Me Performance

Metric	**Before Post-processing**	**After Post-processing**
Precision (%)	23.84	36.99
Recall (%)	82.66	73.09
F1-Score (%)	37.01	49.12
Tempo (BPM)	168	168
Note Count	1927	749

5.3 Discussion of Results

Prior to discussing the results it should be noted that besides the testing dataset, which contained a mix of live audio recordings and MIDI generated recordings, all individual songs used to obtain results are MIDI generated. This distinction is important because MIDI generated audio contains no noise and the onsets and offsets are more clearly defined. This means MIDI generated audio acts as essentially an ideal input that is prepared with no human error or noise to filter out. Therefore all result are likely to be stronger than

5.3.1 Benchmarking

When comparing the benchmarking results of the AMT models built across the different feature extraction techniques. The CQT based models overall had the best average performance. The MFCC based models produced the lowest overall average. Interestingly the performance of the Mel-spectrogram based model improved in order of DNN, CNN and finally the LSTM.

The results from the CQT models are interesting because in theory the best performing model would've been the LSTM followed by the CNN then the DNN. However the results show the exact opposite trend. The most likely reason for this is that the models simply did not train long enough for the LSTM and CNN to fully understand the relationships between what they received as input which could've resulted in lower performance. This could be further supported by the fact that CQTs contain a plethora of musically relevant information [12] thus the DNN could excel at feature extraction without specialised techniques [13] that could complicate the task such as those of the CNN and LSTM.

The results obtained from the MFCC based models are all underwhelming which indicates that the this method may not be suitable for feature extraction in AMT systems. This could be due to the compactness of MFCC's as noted in Section 2.3.4. To obtain the smaller file sizes it is most likely that the data from the audio was compressed until it was no longer suitable for feature extraction. Therefore resulting in weak performance as there is not enough information for the neural networks to adequately train on.

The mel-spectrogram based models produce results that align with the trends proposed in theory. The results indicate that LSTMs in combination with the mel-spectrograms could potentially produce the best performance. This is likely due to the more compressed nature of mel-spectrograms as compared to CQTs therefore allowing the model to learn temporal relationships of music faster than its CQT based alternative.

5.3.2 Initial Song Test

With a firm understanding of the possible reasons behind the performance in the benchmarking stage, Each model performed a test on an unseen song. Across all models there was a decline in performance. However there was a particularly large drop in performance for the mel-spectrogram based models.

While the test data used for benchmarking was separate form the rest of the data it should be noted that the MAPS dataset does contain some songs that appear more than once but under different recording conditions. Thus it is possible that if any repeating songs were present in the test set then an overfit model could potentially produce higher scores and perform poorly on unseen data. Another possibility is a lack of variability in the types of music used in the training stage as such it may be overfit for a particular type of music and would therefore severely underperform when presented music of another genre. On the basis of these conclusions the best solutions would be to provide the model with more training data with a variety of different musical styles present.

Upon review of the models' performance on a song considered ideal for them (monophonic piano song with few notes) the best model is noted and used for all further testing

5.3.3 CQT DNN Monophonic Performance

This section of the discussion reviews the performance of the songs that were tested on the finalised model. From the results obtained from these tests it it immediately apparent that the model overall performs below the expectations set by the benchmark test. However once the post-processing algorithm is applied to the predictions the performance of the model significantly increases to the point that it even outperforms the benchmark results on monophonic piano music i.e "Silent Night" and "Hedwig's Theme".

It however should be noted that even after post processing is complete the outputs still have more notes than the original midi file. This is a clear indicator that the model is producing a significant amount of false positives which in turn are affecting the precision score. It is also of note that after post-processing the recall score of the model is falling meaning that it is finding the correct notes in the wrong places as well. These false positives are most likely harmonics that were detected as onsets.

Overall the model has strong performance on monophonic music especially after post-processing is applied to remove false positives. However the model does perform poorly on "Fur Elise" notably worse than on any other song, including the ones used to test the limits of the system. Therefore it is imperative to determine the cause of this abysmal result.

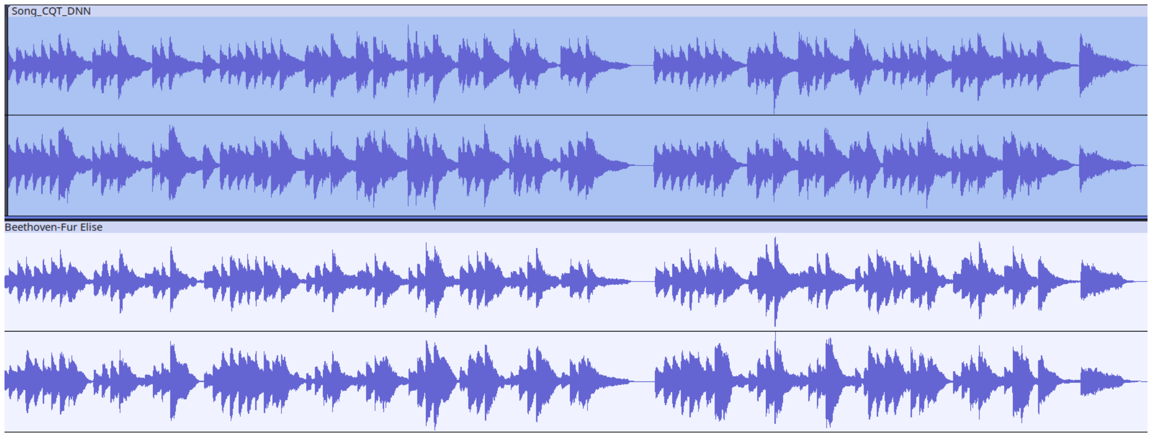

Figure 5.1: Fur Elise transcription wave (top) and original wave (bottom)

The model's performance on "Fur Elise" does not necessarily mean that the transcription of the song is objectively unimpressive. As noted in Chapter 3.2 there are no specialised metrics that can measure the perceptual quality. This is particularly apparent in Figure 5.1 where it clearly shown that the waveforms for the transcription and the original audio are quite similar, especially the placement of their peaks and silences. However it is of note that while this is a good indicator that the timing of the notes played is similar it does not show that the pitches predicted are correct.

Figure 5.2 shows the CQT of the "Fur Elise" transcription audio and original. It is immediately apparent that the two CQTs are very similar visually with all their high energy areas being aligned, therefore showing that the peaks in their wav audio are of the same or at worst minutely different frequencies. This proves that the timing and notes played back are immensely similar if not the same. Moreover the standard metrics do not adequately measure perceptual quality of a transcription which is arguably the most important aspect of the output from an AMT system. The standard metrics focus on the similarity between the outputted MIDI file and

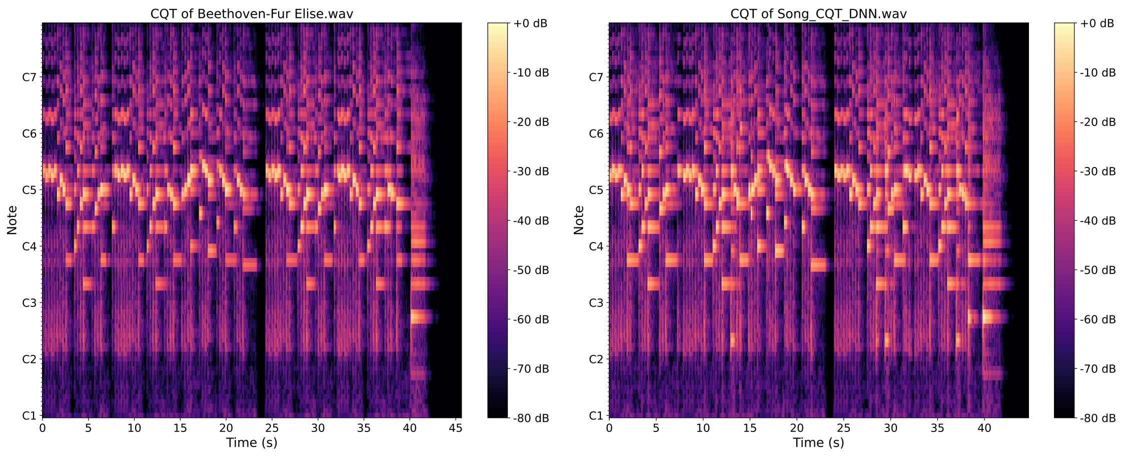

Figure 5.2: Fur Elise transcription CQT (right) and original CQT (Left)

the ground truth MIDI file.

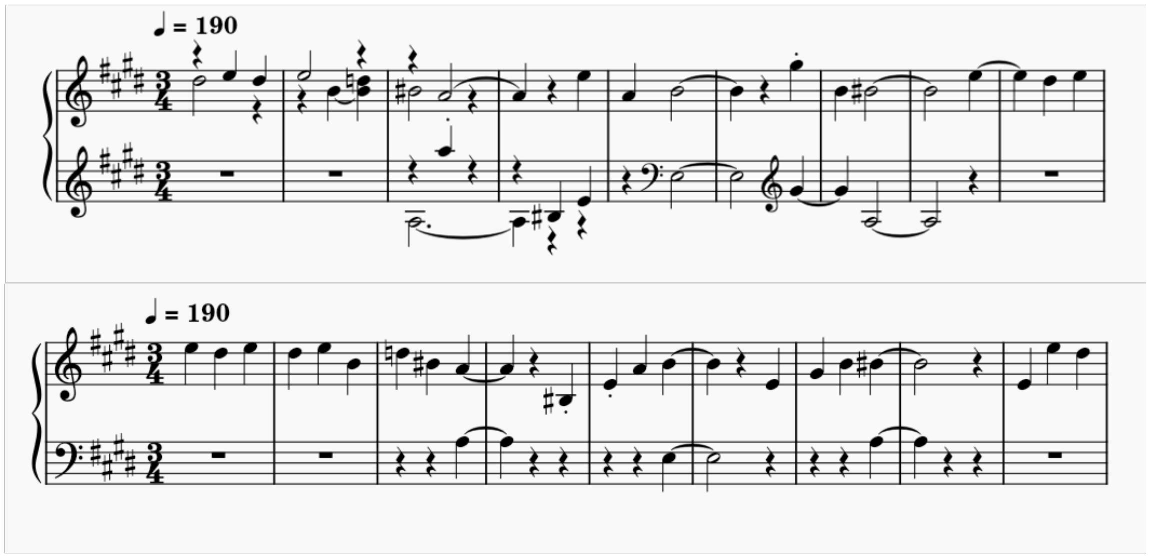

Figure 5.3: Fur Elise transcription (top) and original (bottom)

The transcription quality is checked against the original midi. It is evident from Figure 5.3 that there are quite a few differences between the original MIDI and the transcribed version. Therefore, even though perceptual metrics show that they are in fact highly similar the scores obtained record the performance of the model on the song as poor.

5.3.4 CQT DNN Stress Tests

At first glance the performance of the model on more sophisticated songs is particularly weak with the model failing to obtain an F1-score of 50% or more however this does not necessarily mean that a transcription is subpar.

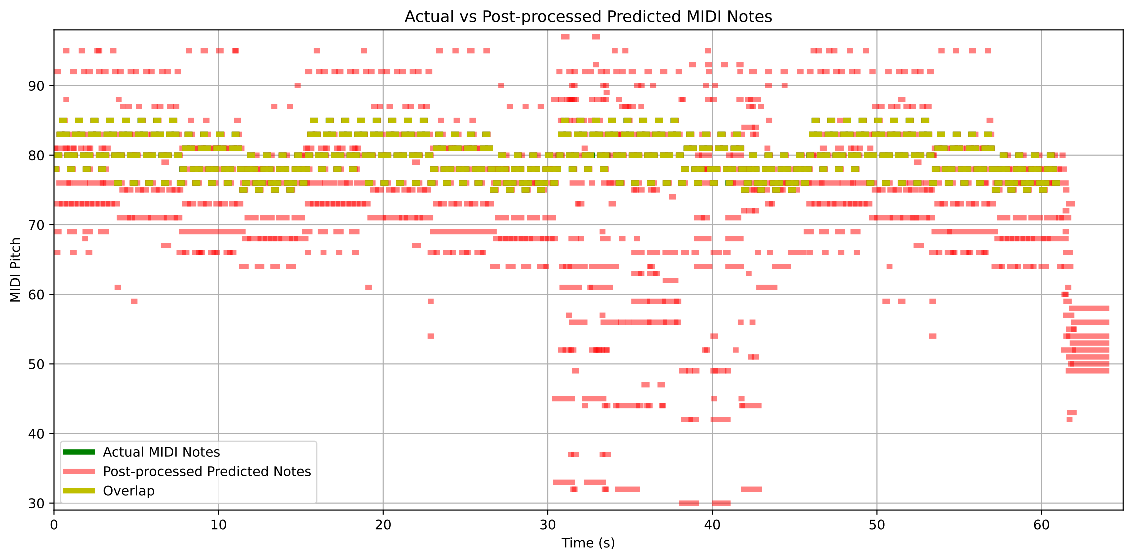

Figure 5.4: Golden Hour Actual Notes vs Final Prediction

”Golden Hour” was chosen as input to test how well the model could transcribe a song that has a rapid succession of notes at a high tempo. Figure 5.4 clearly shows that the model identified almost every single note perfectly. However despite the model’s success in identifying the notes correctly the abundance of false positives significantly lowered the F1-score and precision resulting in the scores recorded.

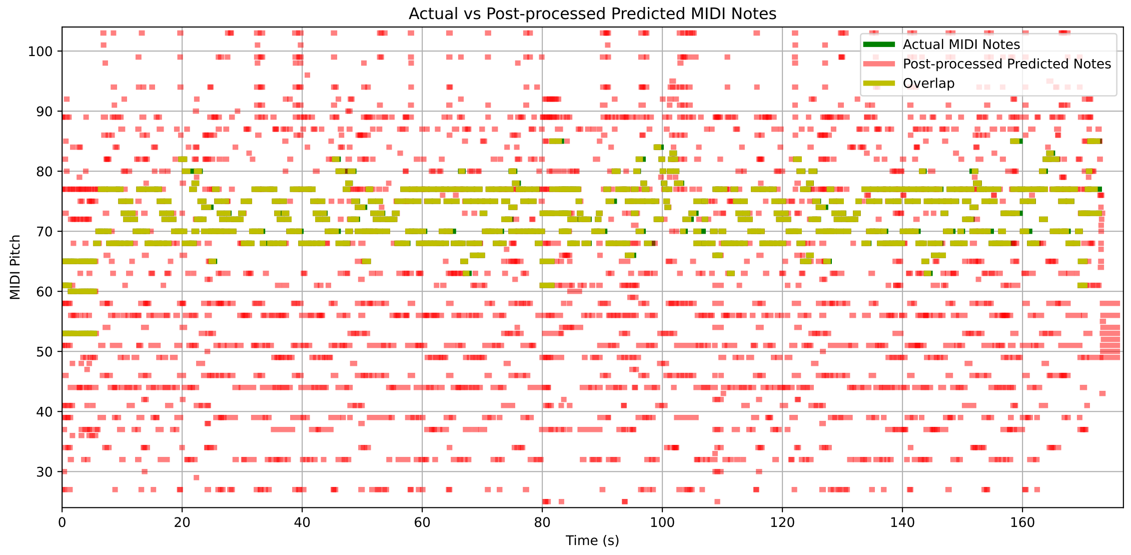

Figure 5.5: From The Start Actual Notes vs Final Prediction

”From The Start” was chosen as an input to determine how well the model can perform on a polyphonic piano song. Despite the model not being trained with the intention of transcribing piano polyphony the model managed to correctly transcribe the vast majority of the notes. This shows that the model is capable of handling piano polyphony well. Despite its accurate predictions for the true notes of the song it still suffers from an abundance of false positive which in turn will lower the quantitative scores for this transcription

Finally ”Stay With Me” was chosen as an input to determine how well the model performs on a song mainly consisting of chords with minor polyphonic elements. While this transcription has more missed parts than the others it is due to the sustain as opposed to it missing a note entirely. This means that the offset detection while strong could still use some improvements. This transcription like the others suffers from a plethora of false positives that lower its quantitative scores.

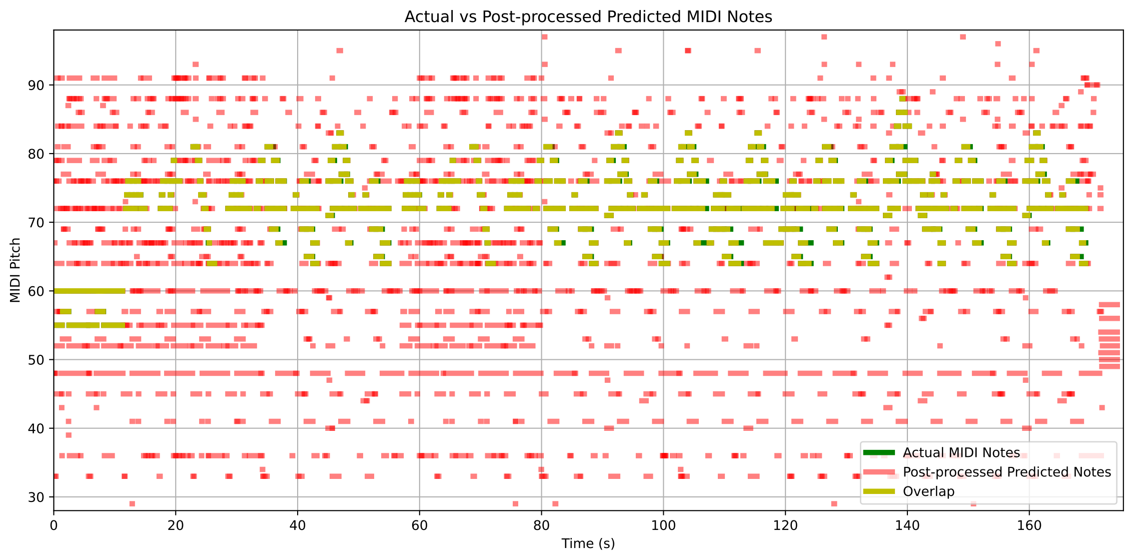

Figure 5.6: Stay With Me Actual Notes vs Final Prediction

5.4 Acceptance Test Procedures

Throughout this paper a variety of techniques and methods have been employed that have culminated into the performance seen in Section5.3. Therefore with the completion of the system and its testing it is imperative to determine if it achieves the acceptance criteria that was outlined for it in Chapter3.3

A-ID	**Piano Note Classification Specifications**	**Status**
A-01	Inputted WAV files must be used to create output MIDI file	Pass
A-02	Numpy array for ground truth must be produced from annotations	Pass
A-03	WAV file must have features extracted into numpy array	Pass
A-04	Min-Max scaling correctly applied on extracted features	Pass
A-05	Large inputs of data must be split to save space	Pass
A-06	Build and train DNN,CNN and LSTM models	Pass
A-07	Code is optimised to extract his data from provided annotations	Pass
A-08	Obtain an F1-score that is better than in initial testing phase	Pass
A-09	Plot graphs on prediction vs actual with overlap	Pass
A-10	Tempo must be double, half or within 5BPM of true tempo	Pass
A-11	Minimum recall score of 60%	Pass
A-12	Final model must be capable of detecting onsets offsets and pitch	Pass
A-13	A 3 minute song must be generated within a 60 seconds	Pass
A-14	Post-processing improves results of transcription	Pass

Table 5.11: AMT System Acceptance Test Procedures

Chapter 6

Conclusions

6.1 Conclusions

This project was started with the aim of designing and implementing an AMT system with a focus on monophonic music using machine learning techniques that should take WAV audio as an input and outputs a MIDI file. This file may be used for further processing or to produce sheet music. The integration of the machine learning, plotting and digital signal processing libraries contributed to the building and comparing of different amalgamations of feature extraction techniques and Neural network models. Upon thorough comparison of the built models the best one was taken and tested across a variety of songs. After initial testing of the model was complete post-processing was employed and it successfully increased the performance of the model. To test its capability it was given not only the monophonic music within the scope of the project, but also polyphonic songs.

The system performed beyond the initial expectations on monophonic music after post-processing was applied. Testing on polyphonic music found that while the system can successfully identify the vast majority of notes in a polyphonic song, he number of false positives it generates hinders its overall performance. Even after post-processing was applied there was an immense number of false positives present significantly lowered the F1-score of the model. Beyond just the metrics of the model it was found that the model is able consistently estimate the tempo correctly and was able to identify the correct key for the majority of the songs tested thus further proving how well the model works.

Overall the project is considered to be a success as it passed all outlined acceptance test procedures. This however does not mean that the model is perfect t still has plenty of room for improvement, it does however present a viable basis on which future work can be built upon.

6.2 Future Work

Given the shortcomings of the system, that were identified in Chapter 5.3 the first problem to tackle would be the development of more perceptual metrics. It was seen in Chapter 5.3 that there are times where the standard metrics could give a false sense of how well the model is actually performing.

The model in its current state while being able to confidently identify true onsets, does have its performance suffer due to how many false positives it generates especially in polyphonic music. It has shown that even with polyphonic music it can identify the correct notes and timings for the melodies played. Therefore by implementing a more robust fix to this problem it could potentially make the model immediately suitable for polyphonic piano music transcription.

The model that was ultimately constructed for this project was developed through many compromises. One major compromise was the use of a small section of a dataset for training. It would be ideal to train the model on a larger dataset with greater variability. Another improvement would be the post-processing which was limited to using manual methods such as peak picking to implement it due to the large amount of time it takes to train a hidden markov model (HMM) thus it may serve the project well to develop a hidden markov model to further improve onset detection. Another consideration for improving the model would be the filtering of notes that do not fall within the detected key of a piece. This could significantly reduce false positives due to harmonics.

Ultimately the best outcome for this work is for it to be used as the basis for a model that would be able to transcribe polyphonic music. Therefore all future efforts should be towards that goal. While this model was intended for monophonic piano music, improving it such that it performs well on polyphonic piano music will only further improve its already good monophonic performance.

Bibliography

[1] GDJ, "Download musical notes, wave, music. royalty-free vector graphic," Pixabay.com, 07 2023. [Online]. Available: https://pixabay.com/vectors/musical-notes-wave-music-abstract-8135225/

[2] F. Simonetta, S. Ntalampiras, and F. Avanzini, "Audio-to-score alignment using deep automatic music transcription," in *2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP)*, 2021, pp. 1–6.

[3] B. Parker, *Good Vibrations: The Physics of Music*, ser. Good Vibrations. Johns Hopkins University Press, 2009. [Online]. Available: https://books.google.co.za/books?id=YP6EDIcj8dwC

[4] S. R. , "Properties of longitudinal waves – hsc physics," Science Ready. [Online]. Available: https://scienceready.com.au/pages/properties-of-longitudinal-waves

[5] W. Bennett, A. Morrison, and C. Holland, *The Science of Musical Sound: Volume 1: Stringed Instruments, Pipe Organs, and the Human Voice*. Springer International Publishing, 2018. [Online]. Available: https://books.google.co.za/books?id=eB1tDwAAQBAJ

[6] N. Carter, *Music Theory: From Beginner To Expert - The Ultimate Step-By-Step Guide to Understanding and Learning Music Theory Effortlessly*.

CreateSpace Independent Publishing Platform, 2016. [Online]. Available: https://books.google.co.za/books?id=_h9GvgAACAAJ

[7] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription: Challenges and future directions," *Journal of Intelligent Information Systems*, vol. 41, 12 2013.

[8] Havik, "Automatic transcription of monophonic music using digital signal processing," *University Of Cape Town*, p. 69, 10 2023.

[9] K. Fathurahman and D. P. Lestari, "Support vector machine-based automatic music transcription for transcribing polyphonic music into musicxml," 08 2015. [Online]. Available: https://ieeexplore.ieee.org/document/7352558

[10] K. R. Chithra and M. S. Sinith, "A comprehensive study of time-frequency analysis of musical signals," 05 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10170306

[11] A. Elbir, H. O. İlhan, G. Serbes, and N. Aydın, "Short time fourier transform based music genre classification," in *2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)*, 2018, pp. 1–4.

[12] R. A. Dobre and C. Negrescu, "Automatic music transcription software based on constant q transform," in *2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 2016, pp. 1–4.

[13] S. Sigtia, E. Benetos, and S. Dixon, "An end-to-end neural network for polyphonic piano music transcription," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 5, pp. 927–939, 2016.

[14] M. Subramanian, L. S. S, and R. V R, "Deep learning approaches for melody generation: An evaluation using lstm, bilstm and gru models," in *2023 14th*

International Conference on Computing Communication and Networking Technologies (ICCCNT), 2023, pp. 1–6.

[15] M. Scarpiniti, E. Sigismondi, D. Comminiello, and A. Uncini, "A u-net based architecture for automatic music transcription," in *2023 IEEE 33rd International Workshop on Machine Learning for Signal Processing (MLSP)*, 2023, pp. 1–6.

[16] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "A comparison of audio signal preprocessing methods for deep neural networks on music tagging," in *2018 26th European Signal Processing Conference (EUSIPCO)*, 2018, pp. 1870–1874.

[17] B. S. Gowrishankar and N. U. Bhajantri, "An exhaustive review of automatic music transcription techniques: Survey of music transcription techniques," in *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, 2016, pp. 140–152.

[18] S. T. , "Convolutional neural networks (cnn): Summary," www.superdatascience.com, 08 2018. [Online]. Available: https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-summary

[19] F. Cong, S. Liu, L. Guo, and G. A. Wiggins, "A parallel fusion approach to piano music transcription based on convolutional neural network," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 391–395.

[20] A. Ashfaq, "Rnns (recurrent neural networks) ever wondered how we get to understand things just by looking at them. we don't need to think from scratch to learn something new, rather we build upon the foundations we've already acquired." Linkedin.com, 05 2024. [Online]. Available: https://www.linkedin.com/pulse/understanding-lstm-networks-aqsa-ashfaq-l9bif

[21] B. S. Gowrishankar and N. U. Bhajantri, "Deep learning long short-term memory based automatic music transcription system for carnatic music," in *2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, 2022, pp. 1–6.

[22] A. Braz, C. M. F. Rubira, and M. Vieira, "Development of complex software with agile method," in *2015 Agile Conference*, 2015, pp. 97–101.

[23] M. Technologies and G. E. M, "Agile development with musato technologies," Musato Technologies, 01 2021. [Online]. Available: https://musatotech.co.za/agile-development/

[24] N. Funakoshi and T. Tanaka, "Accurate estimation of tone duration in automatic music transcription," in *Proceedings of SICE Annual Conference 2010*, 2010, pp. 588–590.

[25] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, "Onsets and frames: Dual-objective piano transcription," 2018. [Online]. Available: https://arxiv.org/abs/1710.11153

[26] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic music transcription: An overview," ResearchGate, 2019. [Online]. Available: https://www.researchgate.net/publication/330068609_Automatic_Music_Transcription_An_Overview

[27] J. Nam, J. Ngiam, H. Lee, and M. Slaney, "A classification-based polyphonic piano transcription approach using learned feature representations." 01 2011, pp. 175–180.

[28] L. Iben Nasr, A. Masmoudi, and L. Hadrich Belguith, "Natural tunisian speech preprocessing for features extraction," in *2023 IEEE/ACIS 23rd International Conference on Computer and Information Science (ICIS)*, 2023, pp. 73–78.

[29] R. Prieto and D. Bravo, “Machine learning and audio signal processing for predictive maintenance: A review,” in *2023 IEEE 6th Colombian Conference on Automatic Control (CCAC)*, 2023, pp. 1–6.

[30] Q. Kong, B. Li, X. Song, Y. Wan, and Y. Wang, “High-resolution piano transcription with pedals by regressing onset and offset times,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3707–3717, 2021.

[31] R. G. C. Carvalho and P. Smaragdis, “Towards end-to-end polyphonic music transcription: Transforming music audio directly to a score,” in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2017, pp. 151–155.

[32] S. Masood and A. Abbas, “Neural networks and deep learning: A comprehensive overview of modern techniques and applications,” 01 2024.

[33] Y.-T. Wu, B. Chen, and L. Su, “Automatic music transcription leveraging generalized cepstral features and deep learning,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 401–405.

[34] B. McFee, C. Raffel, D. Liang, D. Ellis, M. Mcvicar, E. Battenberg, and O. Nieto, “librosa: Audio and music signal analysis in python,” 01 2015, pp. 18–24.

[35] A. Ycart and E. Benetos, “Learning and evaluation methodologies for polyphonic music sequence prediction with lstms,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1328–1341, 2020.

[36] Y.-T. Wu, B. Chen, and L. Su, “Polyphonic music transcription with semantic segmentation,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 166–170.

[37] W. Yiyan, "Equalization filter algorithm of music signal based on time-frequency domain analysis," in *2019 IEEE 2nd International Conference on Electronics and Communication Engineering (ICECE)*, 2019, pp. 75–78.

[38] G. Schuller, "Low latency time domain multichannel speech and music source separation," in *2021 55th Asilomar Conference on Signals, Systems, and Computers*, 2021, pp. 549–553.

[39] W. Wang, "Music chord sequence recognition method based on audio feature extraction algorithm," in *2023 IEEE International Conference on Control, Electronics and Computer Technology (ICCECT)*, 2023, pp. 1286–1289.

[40] J. Bello, L. Daudet, and M. Sandler, "Automatic piano transcription using frequency and time-domain information," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, pp. 2242 – 2251, 12 2006.

[41] *Sound - Resonance, Frequency, Wavelength — Britannica*, 2024. [Online]. Available: https://www.britannica.com/science/sound-physics/Open-tubes

[42] J. A. Moorer, "On the transcription of musical sound by computer," *Computer Music Journal*, vol. 1, no. 4, pp. 32–38, 1977. [Online]. Available: http://www.jstor.org/stable/40731298

[43] L. Fitria, Y. K. Suprapto, and M. H. Purnomo, "Music transcription of javanese gamelan using short time fourier transform (stft)," in *2015 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 2015, pp. 279–284.

[44] H. Takeda, T. Nishimoto, and S. Sagayama, "Rhythm and tempo analysis toward automatic music transcription," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, vol. 4, 2007, pp. IV–1317–IV–1320.

[45] F. J. John Joseph, S. Nonsiri, and A. Monsakul, *Keras and TensorFlow: A Hands-On Experience*, 07 2021, pp. 85–111.

[46] N. Degara, A. Pena, M. E. P. Davies, and M. D. Plumbley, “Note onset detection using rhythmic structure,” in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 5526–5529.

[47] Free, “Free piano tutorials with pdf scores and midi download,” Freepianotutorials.net, 2024. [Online]. Available: https://www.freepianotutorials.net/

Appendix A

Supporting Data

All Essential project files are linked below

- Link to youtube channel with audio of transcribed songs
- Github for Piano Note Classification Thesis

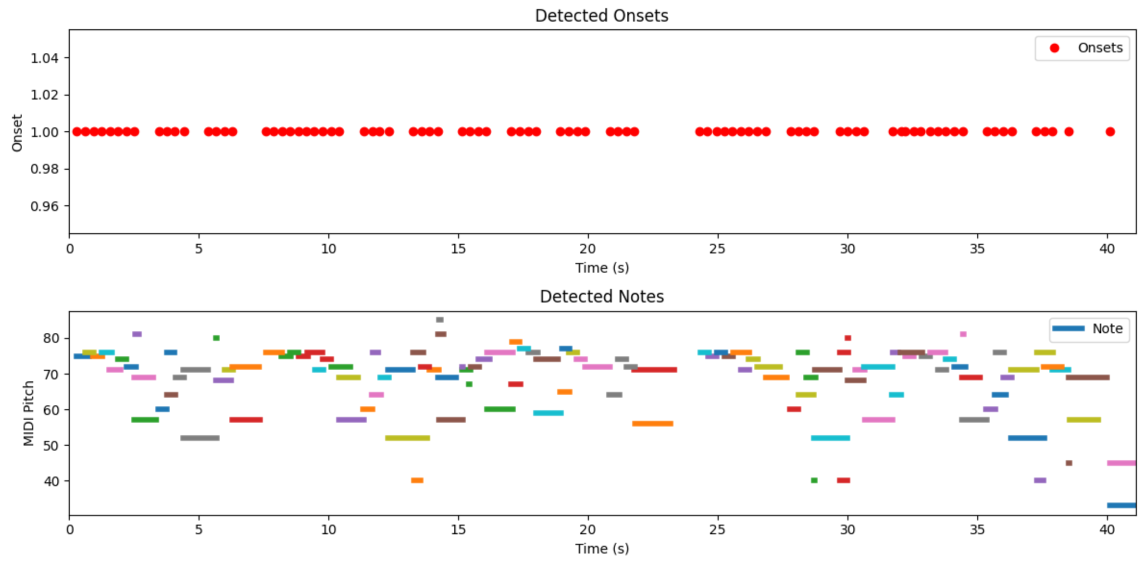

Figure A.1: Example onset check for post processing

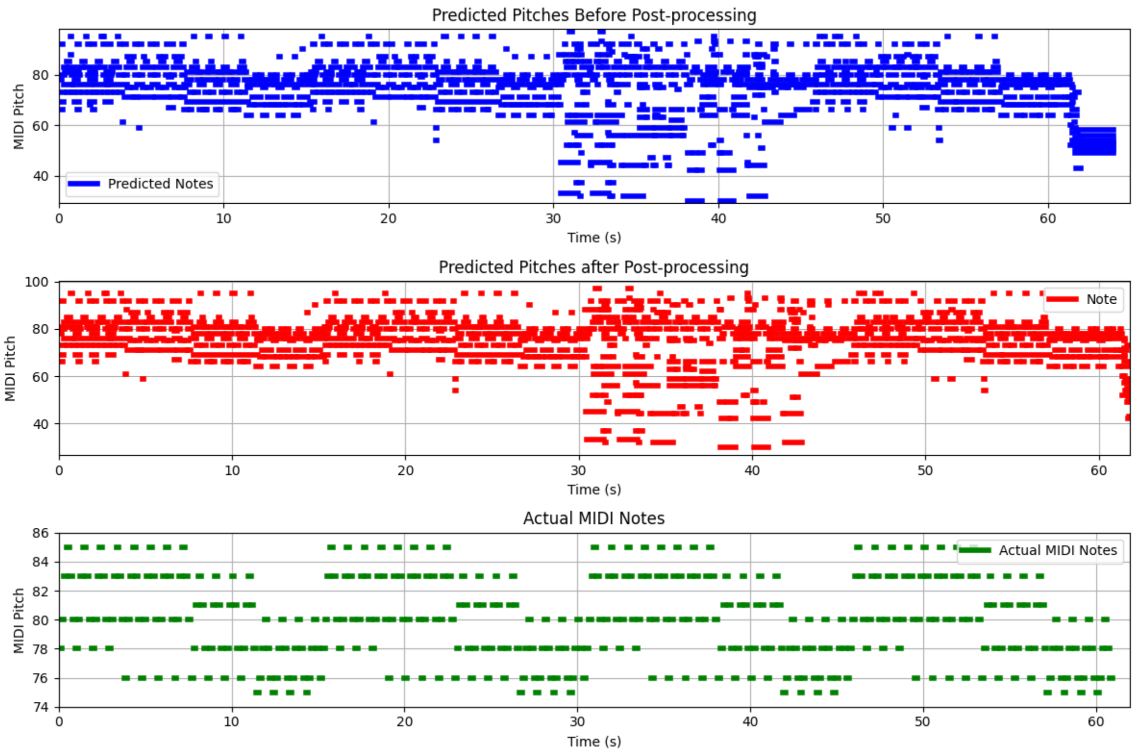

Figure A.2: comparison of notes for Golden Hour

Graduate Attribute	**Response**

GA 1: Problem Solving	The Piano note classification converting an audio recording of music into a symbolic representation, such as musical notation. This is a complex task that involves identifying the pitch, onset, offset, and duration of each note played, as well as other musical elements like tempo. It is a useful technology that should allow a musician to transcribe pieces that they play or hear so that they can recreate them. Machine learning is being used to identify the different musical features present in a spectrogram such as onset, offset, pitch and velocity to identify the notes and their duration in a piece of music. As such different models were used to test identification of each of these characteristics individually before being combined into one system.
GA 4: Investigations, Experiments, and Data Analysis	In order to create the proposed system a comprehensive investigation into machine learning techniques, different preprocessing methods and large amounts of data analysis were conducted for the different models built and how the nature of a model affects the results obtained. Furthermore experimentation using different ground truth labelling methods and frame sizes were conducted. The primary method of model analysis will be done by means of comparing the results of different machine learning architectures such as their accuracy, precision, f1 score and audible accuracy (if someone can identify the original piece based on the output of the models)

GA 5: Use of Engineering Tools	Throughout this project I have utilised visual studio, python and audacity with libraries catered to audio signal analysis and machine learning to optimise my data for training and testing a machine learning model. The use of google colab has also been implemented into my workflow to supplement for the lack of hardware powerful enough to fully train and test the models built.
GA 6: Professional and Technical Communication	Throughout the course of this project I have regularly been in contact with my supervisor to update them on the current state of my project and report in order to give a clear idea of how I am meeting my project milestones to ensure that all work is completed in good time. Along with these a comprehensive report on the entire process will be written, after which a presentation to showcase the project will be prepared and finally a poster to give an overview of the project as a whole will be designed.
GA 8: Individual Working	As I have researched this project I have single handedly reviewed papers, tested different techniques from the papers I read and implemented the different modules of the project. Through my own efforts have developed and modified models to improve the results of the machine learning model striving to achieve greater accuracy. The implementations based on all the testing have been implemented by me alone and my supervisor has provided me with guidance on how to best ensure that I meet milestones.

GA 9: Independent Learning Ability	Over the course of this project I have read extensively about the theory behind music and sound and have used this theory to carefully develop a machine learning model that is able to identify the different characteristics of sound in the context of music to obtain the relevant information necessary for the identification of piano notes. Throughout this project I alone have been responsible for ensuring that I understand the different approaches to the problem and have outlined my own strategy to implementing a fully realised system.

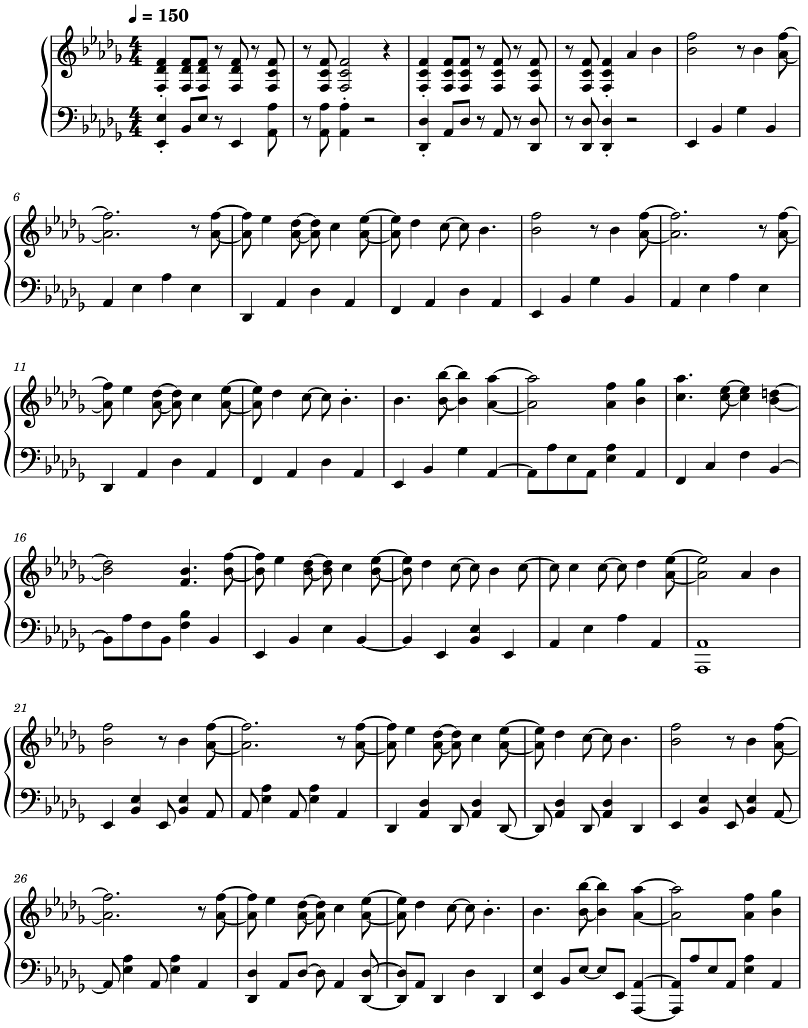

Figure A.3: From the Start Original MIDI

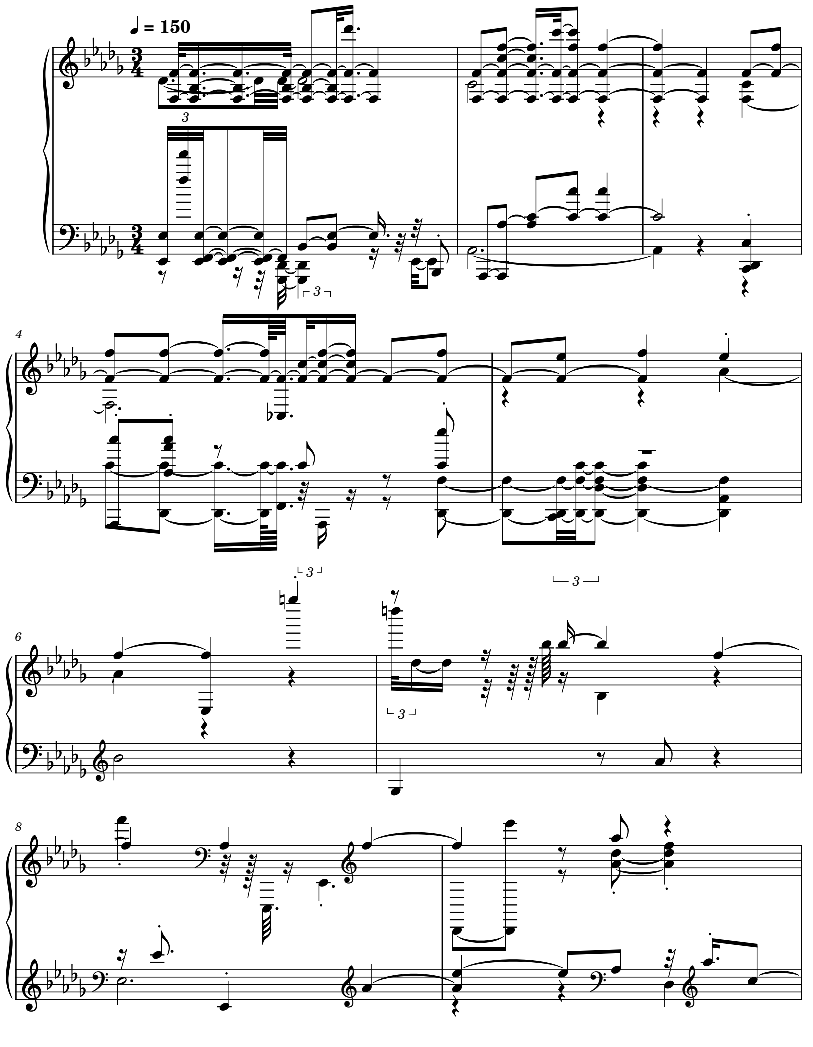

Figure A.4: From the Start Transcribed MIDI

Appendix B

AI use in Thesis

I used AI to check my grammar get definitions of words and utilised it to summarise key points in papers whilst conductiong research. I utilised ChatGPT for all my AI usage. to get the summaries I would upload the document and specify the type of information I'm looking for. Some examples of prompts I used are:

- Could you highlight key points in this paper
- Would you say that the word plethora is suitable for this sentence
- Does this paper contain any information on the theory behind their method