Tampereen yliopisto

Inkariina Simola

TABLATURE NOTATION FROM MONOPHONIC GUITAR AUDIO USING CNN

Faculty of Information Technology and Communication Sciences
M.Sc. Thesis
June 2023

TIIVISTELMÄ

Inkariina Simola: Tablature notation from monophonic guitar audio using CNN
Pro gradu -tutkielma
Tampereen yliopisto
Tietojenkäsittelytieteiden tutkinto-ohjelma
Kesäkuu 2023
Tarkastajat: Martti Juhola, Henry Joutsijoki

Otelaudalla varustetuilla kieli-instrumenteilla tuotetun musiikin automaattinen nuotintaminen synnyttää joko perinteistä nuottikirjoitusta tai tabulatuuria. Perinteisestä nuottikirjoituksesta poiketen tabulatuuri tarjoaa yksikäsitteisen vastaavuuden nuottimerkinnän ja sen otelaudalla olevan sijainnin välille, ja on siksi suosittu nuotintamistapa kitaralle. Yksittäisen nuotin soittamiseen käytetyn kielen ja nauhan yhdistelmän tarkka tunnistaminen äänitteestä edellyttää sekä sävelkorkeuden että kielen tunnistamista, tyypillisesti tässä järjestyksessä. Tämä pro gradu -tutkielma tarkastelee käytetyn kielen tunnistamista sähkökitaralla soitetuista yksittäisistä nuoteista konvoluutioneuroverkon avulla.

Opinnäytettä varten kerättiin yli 10000 sähkökitaralla soitettua nuottia, joiden perustaajuus on tunnistettavissa ja jotka ovat peräisin kolmesta eri kitarayksilöstä. Jokaiselle nuotille laskettiin spektrogrammi, Mel-spektrogrammi ja CQT (constant-Q transform). Konvoluutioneuroverkko koulutettiin tunnistamaan kyseisten piirteiden perusteella ääninäytteen tuottamiseen käytetty kitaran kieli. Neuroverkkomallit arvioitiin 6-kertaisella ristiinvalidoinnilla. Paras tarkkuus 0.932 saavutettiin mallilla joka koulutettiin CQT:n avulla.

Avainsanat: Konvoluutioneuroverkko, automaattinen kitaratranskriptio, signaalinkäsittely, tabulatuuri, sähkökitara

ABSTRACT

Inkariina Simola : Tablature notation from monophonic guitar audio using CNN
M.Sc. Thesis
Tampere University
Degree Programme in Computer Science
June 2023
Supervisors: Martti Juhola, Henry Joutsijoki

Automatic Music Transcription for instruments with fretboards, such as the guitar, involves transcribing audio into either standard notation or tablature notation. Tablature notation provides a one-to-one mapping between the symbol for a note and the string-fret combination used to produce it, and is often preferred over standard notation for this reason. Detecting the string-fret combination used to produce a note involves pitch detection and string detection, which are usually performed in this order in existing approaches. This Master's Thesis focuses on electric guitar string detection from monophonic samples using a convolutional neural network (CNN).

A dataset containing over 10000 guitar notes with a detectable fundamental frequency was collected from three electric guitars and feature engineered to extract spectrogram, Mel-spectrogram and constant-Q transform per sample. Three convolutional neural networks were trained, one on each feature, to detect the guitar string from which each original sample had originated. The models were subjected to 6-fold stratified cross-validation. A string detection accuracy of 0.932 was achieved with the model trained on the constant-Q transform data.

Keywords: Convolutional Neural Network, Automatic Guitar Transcription, Tablature, Electric Guitar

CONTENTS

1 INTRODUCTION **1**

1.1 Motivation 1

1.2. Domain-specific concepts and background 2

1.2.1 Scientific pitch notation 2

1.2.2 Frequencies produced by a vibrating string 2

1.2.3 Equal temperament tuning system 6

1.2.4 Guitar structure 7

1.3. Scope 10

1.3.1 Step 1: Instrument isolation 10

1.3.2 Step 2: Note onset detection 10

1.3.3 Step 3: Fundamental frequency (f_0) detection 11

1.3.4 Step 4: String detection 11

1.3.5 Step 5: Tablature visualization 12

2 RELATED WORK **12**

2.1 Previous work involving constrained tablature generation 12

2.1.1 Constraints based on transition probability 12

2.1.2 Anatomy-based constraints 12

2.2 Previous work involving tablature transcription 13

2.2.1 Previously used datasets 13

2.2.2 Previously used methods 13

3 DATASET **14**

3.1 Data collection procedure 15

3.2 Data cleaning 15

4. METHODS **17**

4.1 Feature engineering 17

4.1.1 Discrete Fourier transform 17

4.1.2 Feature 1: Spectrogram 18

4.1.3 Feature 2: Mel-spectrogram 22

4.1.4 Feature 3: Constant-Q transform 23

4.1.5 Feature engineering algorithm 25

4.2 Convolutional neural network 25

4.2.1 Input layer 27

4.2.2 Convolutional layers 27

4.2.3 MaxPooling 2D layer 29

4.2.4 Dropout layer 29

4.2.5 Dense layer 30

4.2.6 Output 30

4.3 Training and evaluation 30

4.4. Prediction 31

5. RESULTS **33**

6. DISCUSSION AND CONCLUSION **34**

6.1 Discussion and future work: Dataset 34

6.2. Discussion and future work: Methods 35

6.3 Conclusion 36

REFERENCES **37**

1 INTRODUCTION

1.1 Motivation

While standard music notation is perfect for instruments with an inbuilt one-to-one mapping between pitch and fingering position, such as the piano, it is ambiguous when used with fretted instruments such as the guitar. Stringed instruments with fretboards offer several ways to produce most pitches in their range, but standard notation fails to address this issue, which explains the popularity of tablature notation among guitar players. Tablature offers a one-to-one mapping between notes and the fretboard positions used to produce them, by denoting strings with lines and frets by numbers placed on those lines, as shown in Figure 1.

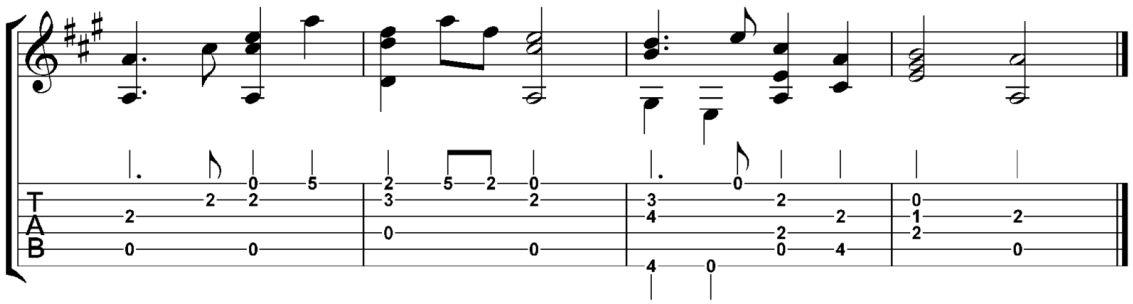


Figure 1: Standard notation (top) and tablature notation for the same part (bottom) ["Guitar Tablature 2005"].

Creating tablature without an automatic transcription system is laborious manual work. As of the writing of this Master's Thesis, there exists no commercial solution capable of performing accurate string-fret detection from audio recordings produced without special equipment (such as hexaphonic pickups or purpose-built MIDI guitars). Much previous academic work on the subject likewise resorts to using plausibility filters to generate probable or playable tablature, instead of concentrating on ground truth string-fret combination detection.

Previous work achieving state of the art results in string-fret detection has relied, with the exception of work done by Dittmar *et al.* [2013], on either a limited dataset of samples produced by a single physical instrument [Kim *et al.* 2022; Wiggins and Kim 2019] or on prior knowledge of an instrument's physical characteristics [Barbanhco *et al.* 2012; Hjerrild *et al.* 2019a]. It also appears that string-fret detection results for the electric guitar fall short of those achieved with acoustic guitars when both scores are reported separately [Hjerrild *et al.* 2019b; Barbanhco *et al.* 2012].

1.2. Domain-specific concepts and background

This section introduces domain-related concepts such as scientific pitch notation, electric guitar structure and the equal temperament tuning system.

1.2.1 Scientific pitch notation

Scientific pitch notation (SPM) specifies musical pitch by assigning an octave index to each musical note within the range of human hearing [International Organization for Standardization (ISO, 1975)]. Figure 2 shows the pitch names for ten different C notes appearing at octave intervals, ranging from C_0 (16.35 Hz) to C_9 at 17739 Hz. The pitch range of a guitar with 24 frets ranges from E_2 to E_6, for which the fundamental frequencies are 82.4 Hz and 1318 Hz respectively.

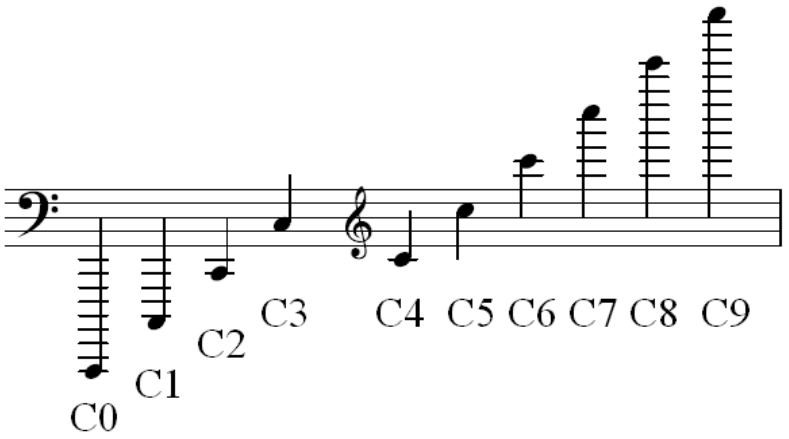


Figure 2: Scientific pitch notation for the musical note C across multiple octaves is shown below corresponding standard notation symbols ["Scientific pitch notation", 2023].

1.2.2 Frequencies produced by a vibrating string

When a string having length L is attached at both ends, tightened and set in motion, it vibrates with several modes of vibration, or standing waves, as shown in Figure 3. The longest vibrating segment has only two nodes, corresponds to a wavelength of $\lambda_0 = 2L$ and produces what is called the lowest partial, or *fundamental frequency* of the vibrating string. Higher partials of a vibrating string are called harmonics, the first harmonic f_1 being produced by segments having length $L/2$ and each subsequent harmonic f_n corresponding likewise to a vibrating length of L/n. The fundamental frequency f_0 of a vibrating string having length L, mass m and tension T is shown in Equation 1.

$$f_0 = \frac{\sqrt{\frac{T}{m/L}}}{2L} \tag{1}$$

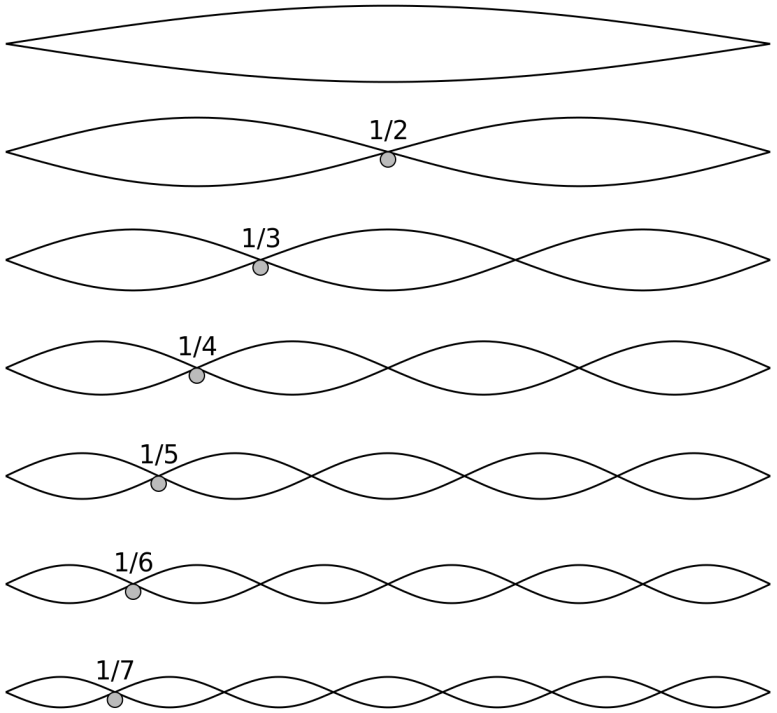


Figure 3: Vibration and standing waves in a string. The fundamental frequency is shown on top, followed by the first six overtones [“Fundamental frequency", 2012].

On a guitar, the fundamental frequency is produced by the segment of a string that is plucked and free to vibrate. When the string rings open, the nodes are located at the bridge and the nut, as shown in Figure 4 (bottom image). If the string is being fretted, the nodes are at the bridge and the fret against which the string is being pressed, as shown in Figure 4 (top image).

In order to observe the frequency spectrum of a given musical note, its discrete Fourier transform (DFT) can be calculated, as described in Section 4.1.1. A series of peaks can then be observed at f_0 and subsequent harmonics by plotting the spectral magnitudes of successive frequency bins against an axis representing frequency. Figure 5a shows the waveform of a note played on the open low E string of an electric guitar and Figure 5b shows the note's frequency spectrum. The fundamental frequency of the note is seen as a spike at 82 Hz and the partials can be seen to its right. Note that the fundamental frequency corresponds to the partial with the lowest frequency, not the one with the largest magnitude.

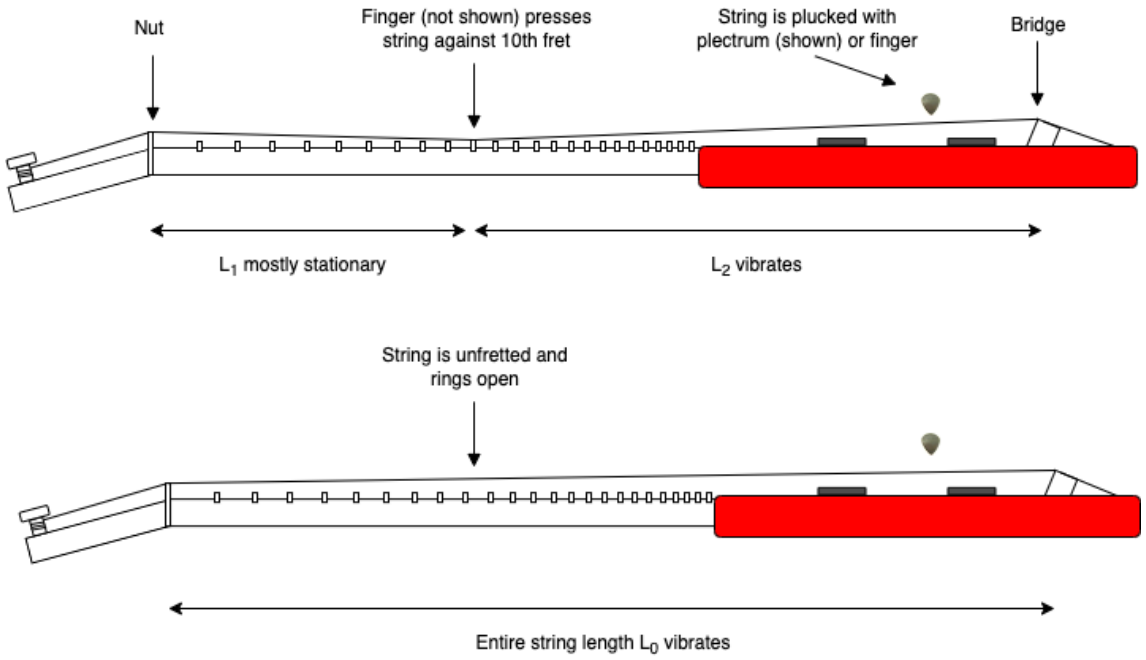


Figure 4: String on an electric guitar viewed from the side (dimensions not to scale). Top: String segment L_2 vibrates between contact points at the 10th fret and bridge. Bottom: Entire string length vibrates, i.e. rings open, between contact points at nut and bridge.

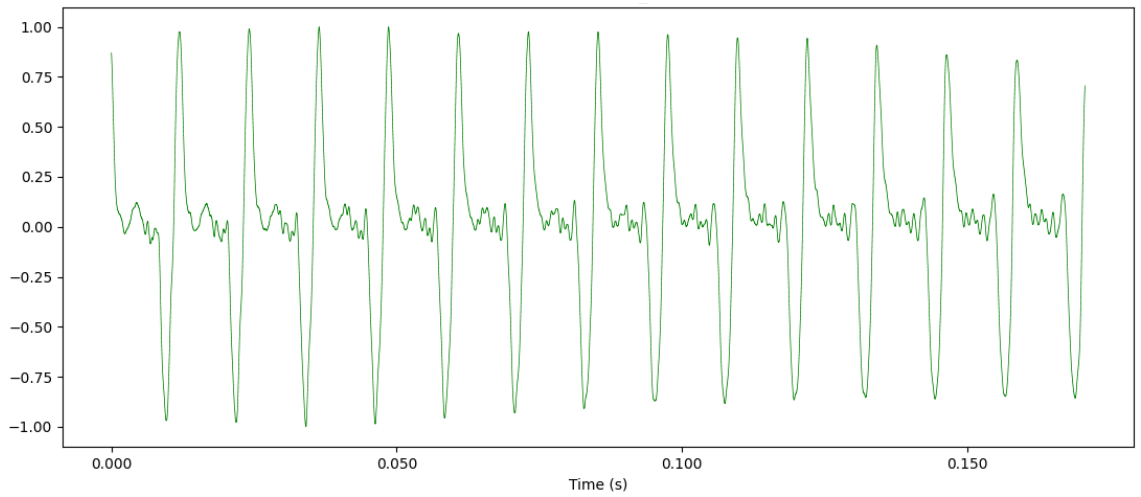


Figure 5a: Waveform of the pitch E_2 played on the open low E string.

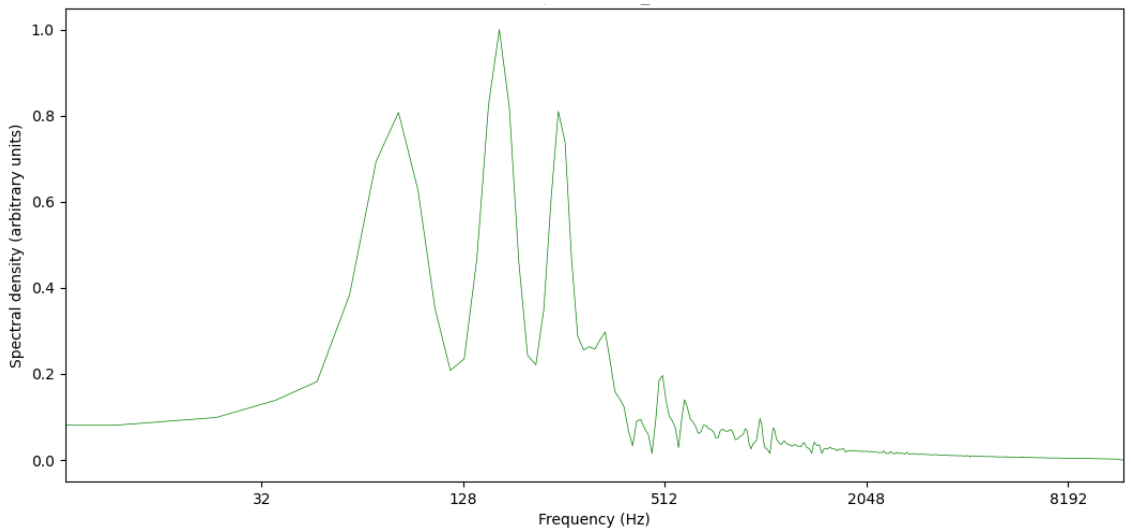


Figure 5b: DFT magnitudes of the waveform shown in Figure 5a. The fundamental frequency is visible as a peak at 82 Hz, with the first and second harmonics showing as peaks at 164 Hz and 328 Hz respectively.

For comparison, Figure 6a shows the waveform of the pitch E_6 played on the 24th fret of the high E string. Less harmonics are present, with only the first harmonic being visible at 2636 Hz. Figures 5b and 6b illustrate the tonal differences at the extremes of an electric guitar's range. At the low end, the thick open E_2 string with a maximally long vibrating segment produces more harmonics than the thin E_6 string fretted at the 24th fret and vibrating with a much shorter segment that contains barely any harmonics. Less pronounced but similar differences apply to samples that originate from different strings but contain the same pitch.

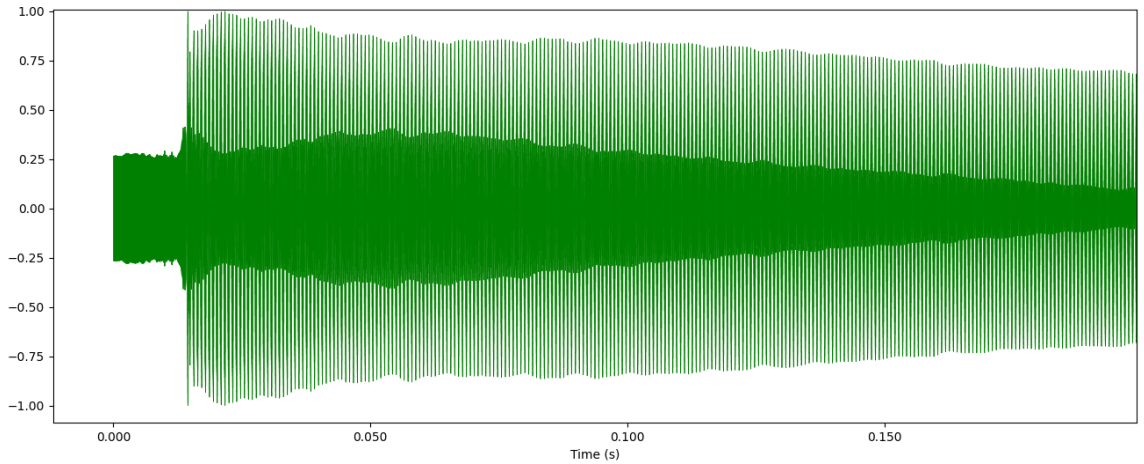


Figure 6a: Audio segment containing pitch E_6 played at the 24th fret of the high E_4 string.

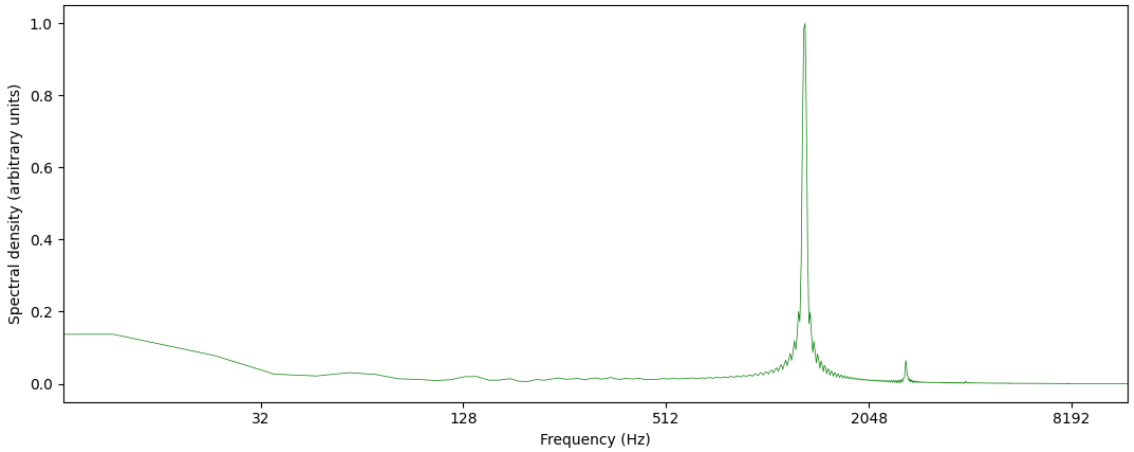


Figure 6b: DFT magnitudes of the waveform shown in Figure 6a. The fundamental frequency is visible as a peak at 1318 Hz.

1.2.3 Equal temperament tuning system

Within the twelve-tone equal temperament tuning system, the frequency ratio between any adjacent pair of notes is the same ["Equal temperament", 2023]. An octave is defined as the distance between a frequency f and its double $2f$, and each octave is divided into twelve semitones. Figure 7 shows the musical pitches playable by most six-stringed guitars as impulses on an x-axis representing frequency.

The twelve semitones within an octave are spaced in such a way that a human ear accustomed to Western music perceives them to be equally spaced, just as it perceives successive octaves to be equally spaced. Each semitone can therefore be expected to have a frequency rf_s, where f_s is the frequency of the preceding (lower-pitched) semitone, r is a coefficient and r^2f_s is the frequency of the following (higher-pitched) semitone. When this definition is combined with the definition of an octave, Equation 2 can be constructed to find the semitone coefficient r.

$$2 = r^{12} \Leftrightarrow r = \sqrt[12]{2} \Leftrightarrow r \approx 1.05946 \quad (2)$$

When detecting fundamental frequency f_0, the concept of a quarter tone becomes relevant. String instruments tend to go out of tune, and pitch detection involves matching detected f_0 within a quarter tone's distance of a given musical pitch. A quarter tone having frequency qf_s lies between two successive semitones having frequencies f_s and q^2f_s. Again, f_s is the frequency of the preceding (lower-pitched) semitone, q^2f_s is the frequency of the following (higher-pitched) semitone, and q is a coefficient. As one octave is divided into 24 quarter tones, Equation 3 can be constructed to calculate the quarter tone coefficient q:

$$2 = q^{24} \Leftrightarrow q = \sqrt[24]{2} \Leftrightarrow q \approx 1.02930 \quad (3)$$

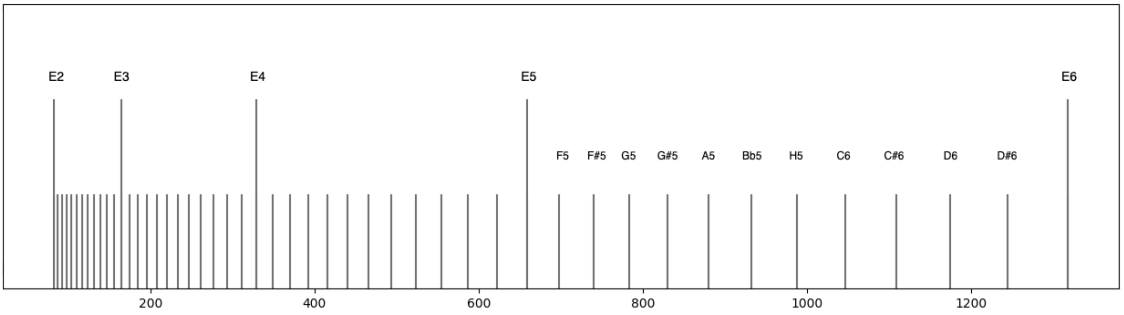


Figure 7: Octave intervals E_2-E_6 and semitone intervals E_5-E_6 shown on an x-axis representing frequency (Hz). Note the increasing distance in Hz between successive semitones and octaves.

Because distances within musical tuning systems are based on ratios, a reference point from which semitones and octaves are calculated using the constant *r* has to be selected. A reference point referred to as *A440*, *Stuttgart pitch*, or A_4 was pinned at 440 Hz by the International Organization for Standardization in 1975 [ISO 1975]. Figure 8 shows this reference pitch as a yellow piano key above a blue piano key showing middle C i.e. C_4.

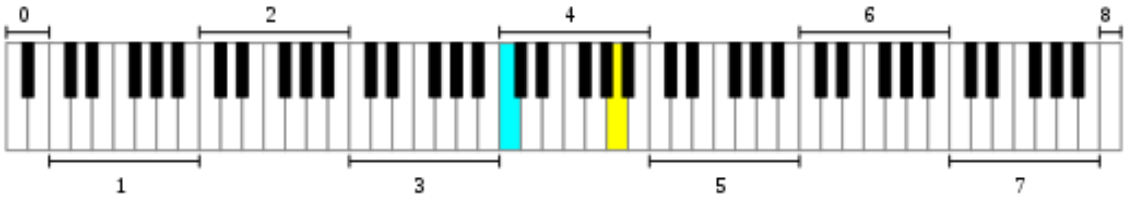
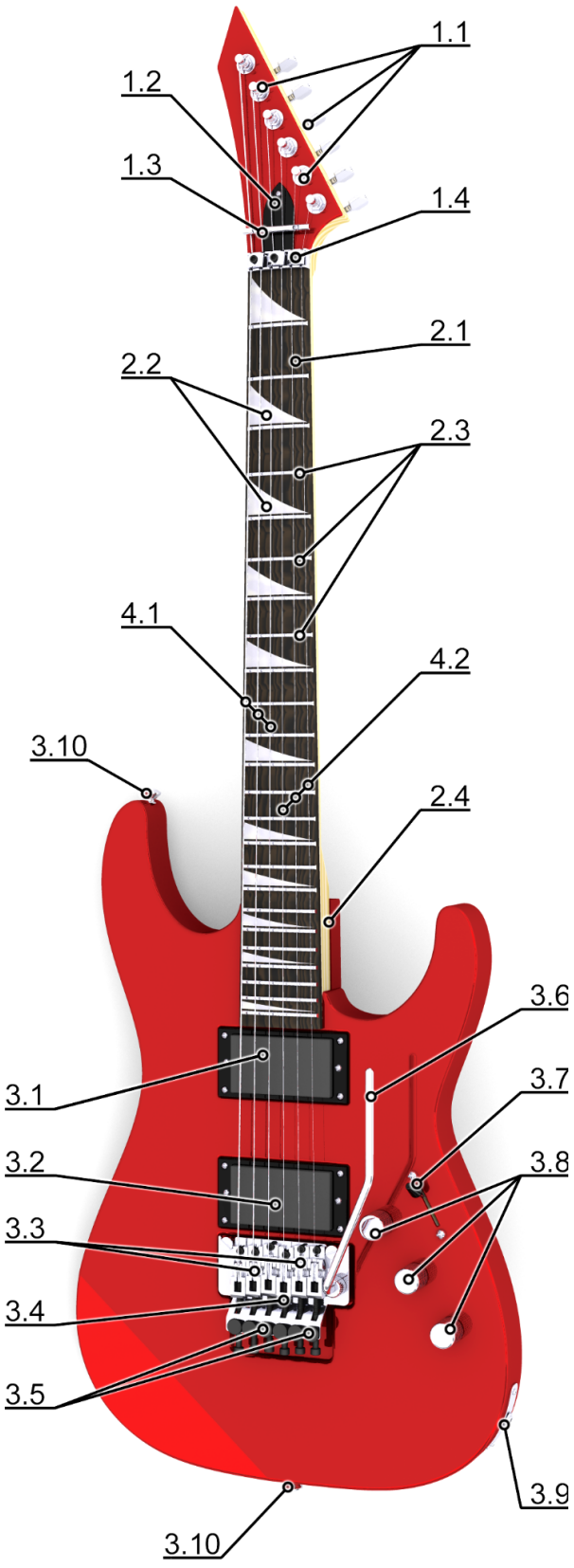


Figure 8: Reference pitch A440 shown as a yellow piano key above middle C, which is shown as a blue piano key [“Piano frequencies", 2012].

Twelve-tone equal temperament is the most commonly used tuning system in the Western world, and will be used in this Master's Thesis.

1.2.4 Guitar structure

The structure and parts of a typical electric guitar are shown in Figure 9. All guitars consist of a *body* (3), a *fingerboard* i.e. fretboard (2.1) supported by the *neck* (2) and at least six *strings* (4). The strings are attached to the *bridge* (3.4) at one end and *machine heads* i.e. tuning pegs (1.1) at the other. In the case of electric guitars, the body is solid and includes one or more *microphones* (3.1 and 3.2 in Figure 9) for picking up string vibrations. The strings are numbered 1 to 6, starting with the high E string, which is the most lightweight, and ending with the low E string, which is the heaviest string and produces the lowest pitches. In Figure 9, strings E_2, A_2 and D_3 are referred to as *bass strings* (4.1) and strings G_3, B_3 and E_4 are referred to as *treble strings* (4.2). When playing the instrument, strings are plucked with fingers or a plectrum near the microphones, as shown in Figure 4.



1. Headstock
   - 1.1 Machine heads
   - 1.2 Truss rod cover
   - 1.3 String guide
   - 1.4 Nut

2. Neck
   - 2.1 Fingerboard
   - 2.2 Inlay fret markers
   - 2.3 Frets
   - 2.4 Neck joint

3. Body
   - 3.1 "Neck" pickup
   - 3.2 "Bridge" pickup
   - 3.3 Saddles
   - 3.4 Bridge
   - 3.5 Fine tuners
   - 3.6 Tremolo arm
   - 3.7 Pickup selector switch
   - 3.8 Volume and tone control knobs
   - 3.9 Output connector
   - 3.10 Strap buttons

4. Strings
   - 4.1 Bass strings
   - 4.2 Treble strings

Figure 9: Parts of an electric guitar ["Electric guitar", 2023].

Each string on a guitar is attached to a tuning peg and tightened so that it is tightly pressed against two contact points, the *saddle* (3.3) and the *nut* (1.4). The distance between the saddle and the nut - also referred to as scale length - together with string tension and mass determines the fundamental frequency produced by a string when it rings open. This vibrating segment can be shortened by pressing the string against a fret, as shown in Figure 4. The art of guitar playing essentially consists of selective string plucking, combined with maneuvers that change the vibrating lengths of strings.

A *fret* (2.3 in Figure 9) is a thin metal bar that crosses the neck transversely at a point corresponding to a vibrating length required to produce a musical pitch when the string in question is tightened to its intended tension. The fret spacing is anchored on the 12th fret; pressing the string against it halves the length of the vibrating segment and thus produces a pitch that is one octave higher than the string's unfretted pitch. The remaining frets are placed at semitone intervals on either side of the 12th fret.

The strings on a given guitar differ from each other in thickness and density, and sets of guitar strings come in many *gauges*. Some typical gauge ranges are shown in Table 1 (see Section 3). The fundamental frequency produced by a given vibrating segment of a string is determined by Equation 1. Tuning a guitar involves adjusting string tension with the tuning peg (see 1.1 in Figure 9) until the unfretted string rings at its intended pitch, which for a guitar in standard tuning is either E_2, A_2, D_3, G_3, B_3 or E_4, depending on the string in question. The fundamental frequencies of the open strings range between 82 Hz on the low E_2 string and 329 Hz on the high E_4 string.

The structural property most relevant to this Master's Thesis is the fretboard layout, which allows notes of identical pitch to be played from several positions, and thus creates ambiguity with respect to the origin of a recorded note. Figure 10 shows a matrix containing all of the string-fret combinations and their associated musical pitches on a fretboard of a 24-fret six-string guitar in standard EADGBE tuning. The pitch with the most fingering options is E_4, highlighted in green. Only nine musical pitches have a single unambiguous point of origin on this fretboard: C_6, D_6 and E_6 on the first string and pitches E_2-$G\#_2$ on the sixth string.

	← Headstock																				Guitar body →				
String 1	E4	F4	F#4	G4	G#4	A4	Bb4	B4	C5	C#5	D5	D#5	E5	F5	F#5	G5	G#5	A5	Bb5	B5	C6	C#6	D6	D#6	E6
String 2	B3	C4	C#4	D4	D#4	E4	F4	F#4	G4	G#4	A4	Bb4	B4	C5	C#5	D5	D#5	E5	F5	F#5	G5	G#5	A5	Bb5	B5
String 3	G3	G#3	A3	Bb3	B3	C4	C#4	D4	D#4	E4	F4	F#4	G4	G#4	A4	Bb4	B4	C5	C#5	D5	D#5	E5	F5	F#5	G5
String 4	D3	D#3	E3	F3	F#3	G3	G#3	A3	Bb3	B3	C4	C#4	D4	D#4	E4	F4	F#4	G4	G#4	A4	Bb4	B4	C5	C#5	D5
String 5	A2	Bb2	B2	C3	C#3	D3	D#3	E3	F3	F#3	G3	G#3	A3	Bb3	B3	C4	C#4	D4	D#4	E4	F4	F#4	G4	G#4	A4
String 6	E2	F2	F#2	G2	G#2	A2	Bb2	B2	C3	C#3	D3	D#3	E3	F3	F#3	G3	G#3	A3	Bb3	B3	C4	C#4	D4	D#4	E4
Fret	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

Figure 10: Matrix showing the fretboard of a guitar in standard EADGBE tuning seen from above, showing strings 1-6 as horizontal rows and frets 0-24 as columns. Fret number 0 refers to an open string. String-fret combinations that produce the E_4 note are highlighted in green.

1.3. Scope

This section outlines the scope of this Master's Thesis and gives a brief overview of the context.

An end-to-end Automatic Guitar Transcription (AGT) system has to solve some or all of the following problems:

1. Instrument isolation from surroundings
2. Note onset detection
3. Polyphonic fundamental frequency (f_0) detection
4. String detection
5. Tablature visualization

String-fret combination detection is achieved by combining the results of fundamental frequency detection with the results of string detection. This masters' thesis will be limited to monophonic audio samples and focus only on the step involving string detection. The other steps are briefly touched on below to introduce context around the issue of string detection.

1.3.1 Step 1: Instrument isolation

All existing research on guitar tablature creation that the author was able to find during the writing of this Master's Thesis (see Section 7) involves a setup using isolated, unprocessed recordings containing the sound of a single acoustic, electric, classical or bass guitar. In the real world, this setup corresponds to transcribing a new piece, exercise or a pre-recorded solo guitar recording with minimal post-processing.

It could be argued that transcribing isolated single-guitar recordings represents a small subset of actual scenarios where tablature notation is called for. A real-world guitarist might wish to learn guitar parts embedded in popular music pieces instead, and this would require methods that are able to extract tablature from multi-instrument compositions. Unfortunately no previous research exists on tablature transcription from audio containing multiple instruments, perhaps because of the increased technical complexity involved.

This Master's Thesis follows the approach of existing research on tablature transcription and concerns itself with isolated guitar recordings only.

1.3.2 Step 2: Note onset detection

Note onset detection is required to separate individual notes from preceding and subsequent notes in order to limit the scope of string-fret detection to a single musical note or chord at a time. For this Master's Thesis, onset detection is performed with **librosa** [McFee *et al.* 2023] during dataset construction stage and prediction stage. See Figures 11 and 21 in Sections 3.2 and 4.4.

1.3.3 Step 3: Fundamental frequency (f_0) detection

Approaches used for pitch detection in existing literature include the openSMILE toolkit [Eyben and Schuller 2010], pYIN [Mauch and Dixon 2014] and custom methods.

In previous work on tablature transcription, fundamental frequency detection is performed prior to or separately from string detection. Detecting f_0 first provides the string detection component with some awareness of which strings to expect, as each string can only produce a subset of the musical pitches within a guitar's range, and the detected pitch narrows down string options. However, detecting f_0 prior to detecting string can also subject the string detection stage to errors that happen at f_0 detection stage. Wiggins and Kim [2019] note that in multipitch estimation systems, a pitch can easily be mistaken for the overtones of a coexisting pitch, and while this is not an issue in monophonic f_0 detection, it is possible that string detection done prior to f_0 detection could inform multipitch detection in future work.

The majority of existing work on tablature transcription uses datasets containing both monophonic and polyphonic samples. Because of time constraints, the dataset used in this Master's Thesis contains only monophonic samples. Pitch detection is used during dataset construction stage to discard samples without a detectable pitch, including accidental thumps, plectrum scrapes and similar non-harmonic artifacts. The algorithm used for f_0 detection is the **librosa.pyin** implementation of the PYIN f_0 estimator [Mauch and Dixon 2014].

1.3.4 Step 4: String detection

Existing research on tablature transcription approaches string detection with a range of different *plausibility filters*. At one end of the range, the aim is to generate playable or plausible tablature using a set of constraints based on transition probabilities and assumed biomechanical constraints. This approach will be referred to as *tablature generation* in later sections. At the other end, the focus is on detecting ground truth string-fret combinations, without the influence of prior musicological or biological domain knowledge or dataset bias. This latter approach will be referred to as *tablature transcription* in later sections.

This Master's Thesis focuses on tablature transcription, not tablature generation. The aim is to detect string-fret position accurately, regardless of how musically unlikely or unplayable the resulting tablature turns out to be. The only constraint applied is a minimalistic plausibility filter that rules out absolutely impossible notes, i.e. pitches that do not exist on the fretboard on a given string in standard tuning.

As a result of these decisions, the section on previous work focuses heavily on research that has a purely detection-oriented angle and mentions constraint-informed research only in passing. Research which fails to report, or establish a performance metric for, string-fret detection results is also omitted [Barbancho *et al.* 2009; Traube and Smith 2001].

1.3.5 Step 5: Tablature visualization

In order for a tablature transcription system to produce usable results, the results have to be presented visually to the user. There are several ways to generate tablature; it is possible to omit note duration information, for example, and this approach is widely used on popular tablature sharing sites, such as Ultimate Guitar ["Ultimate Guitar"]. This Master's Thesis is concerned with string detection only, and will therefore not concern itself with tablature visualization.

2 RELATED WORK

Previous approaches to string-fret combination detection are given an overview in this section.

As mentioned in Section 1.3.4, research on transcribing string-fret combinations focuses either on detecting string-fret-combination ground truth, referred to here as tablature transcription, or on attempts to generate playable or probable tablature, referred to here as tablature generation. Although previous research on tablature generation is given an overview in Section 2.1.1 of this Master's Thesis, the main focus is on tablature transcription.

2.1 Previous work involving constrained tablature generation

A selection of different constraints comes up in previous research on guitar tablature generation. It is possible, for example, to extract transition likelihoods from training data [Boloyos *et al.* 2021; Ryynanen and Klapuri 2007] or to attempt to quantify the difficulty of transitions or hand stretches from an anatomical perspective [Dittmar *et al.* 2013].

2.1.1 Constraints based on transition probability

Boloyos *et al.* [2021] used recurrent neural networks (specifically LSTMs) and Hidden Markov Models on GuitarSet [Xi *et al.* 2018] samples in an attempt to explore the effect of adding more temporal context to the act of transcription. Transition likelihoods learned from any given dataset are subject to bias however, i.e. they may successfully predict tablature for pieces resembling the training data, but fail when applied to different music genres. Strings of different thicknesses also exhibit different sonic characteristics, and detecting string-fret combination ground truth is important in order to preserve tonal decisions made by the composer of a piece. Constraints based on transition probabilities learned from training data are therefore not considered in this Master's Thesis.

2.1.2 Anatomy-based constraints

Several previous works attempt to establish definitions for anatomically impossible fingerings and transitions on the fretboard [Dittmar *et al.* 2013; Barbancho *et al.* 2012]. A maximum stretch distance of 6 frets was imposed by Dittmar *et al.* [2013] on the tablature generation algorithm, the assumption being that chords spread out over large fret distances would be impossible for anyone to play. A decision like this may approximate reality for some combinations of player and guitar, but fails to account for differences in hand size, proficiency, guitar neck dimensions and the effect of playing position. As fret spacing varies along the

fretboard, fret distances that would be impossible for a given player to cover near the fretboard nut may be easily managed near the body of the guitar (see Figures 4 and 9 in Sections 1.2.2 and 1.2.4). Defining what constitutes an impossible transition depends also on the tempo of the piece in question. Furthermore, some playing techniques involve using not one but two hands on the fretboard, which allows for humanly impossible distances between simultaneously and consecutively fingered frets.

The desire to create tablature that is accessible to the widest possible audience may be a motivating factor behind the use of anatomic constraints in tablature generation. There exists, however, no scientifically sound method of drawing a line between playable and unplayable tablature. For this reason, anatomical constraints are left entirely out of consideration in this Master's Thesis.

2.2 Previous work involving tablature transcription

This section describes the datasets and methods that have been used in previous research on tablature transcription, i.e. string-fret combination ground truth detection.

2.2.1 Previously used datasets

In existing research, the most commonly used dataset is GuitarSet [Xi *et al.* 2018], a collection of acoustic guitar samples created and annotated with the help of a hexaphonic pickup attached to the data collection guitar. The dataset was used exclusively by Kim *et al.* [2022], Maaiveld [2021], Wiggins and Kim [2019] and Cwitkowitz *et al.* [2023], and augmented with the 10-chord Montefiore dataset[1] [Osmalskyj *et al.* 2012] by Jadhav *et al.* [2022]. The Real World Corpus (RWC) dataset [Goto *et al.* 2002], containing three and four acoustic and electric guitars respectively, was used by Barbancho *et al.* [2012] and Michelson *et al.* [2018]. Samples from two electric guitars in the IDMT-SMT-Audio-Effects dataset [Stein *et al.* 2010] were used by Abeßer [2013].

In addition to the aforementioned datasets, a variety of unpublished bespoke sample collections was used alone or in combination with a published dataset. Geib *et al.* [2017] constructed a special dataset to complement their detection method which is based on what the authors refer to as string-inverse frequencies, i.e. the vibrations of string segment L_1 in Figure 4 (top image). Monophonic samples from an electric guitar, the most relevant instrument with respect to this Master's Thesis, were included in the datasets used by Abeßer [2013], Barbancho *et al.* [2012], Dittmar *et al.* [2013], Geib *et al.* [2017], Hjerrild and Christensen [2019], Hjerrild *et al.* [2019] and Michelson *et al.* [2018].

2.2.2 Previously used methods

Although most research on ground truth string-fret combination detection was based on audio exclusively, some attempts relied on additional channels of information. Six-channel audio transmitted by a special hexaphonic pickup was assumed as input in the methods proposed by O'Grady and Rickard [2009] and Reboursière and Dupont [2013], while Paleari *et al.* [2008] and Perez-Carrillo *et al.* [2016] relied on audiovisual input. The inharmonicity coefficients

[1]This dataset is no longer available.

used for string detection by Barbancho *et al.* [2012] yielded a near-perfect accuracy of 0.997, but had to be estimated beforehand per string. When using averaged coefficients from a heterogeneous selection of 13 different guitars, the results were inferior to pre-estimated ones with an accuracy of 0.75. The string-inverse frequency method used in [Geib *et al.* 2017] assumed input data recorded in a way that captures both parts of a fretted, vibrating string (i.e. segments L_1 and L_2 in Figure 4, top image) and is not usable with ordinary guitar recordings.

The parametric pitch estimation approach developed by Hjerrild and Christensen [2019] yielded excellent results using the Maximum a priori, with an average absolute error of 3% for string-fret detection on the electric guitar used; however it required a prior training sample from each string of the guitar it was to be used on. (In a subsequent paper by Hjerrild *et al.* [2019], the authors attempted to substitute the training phase with prior knowledge on the physical properties of strings, but the results deteriorated.)

A support vector machine -based approach combined with Inertia Ratio Maximization with Feature Space Projection (IRMFSP) was utilized in two studies to yield a precision of 0.93 [Abeßer 2013] and mean accuracy of 0.92 [Dittmar *et al.* 2013]. The dataset used by Abeßer [2013] consisted of monophonic samples from two electric guitars taken from the IDMT-SMT-Audio-Effects dataset [Stein *et al.* 2010], while a bespoke dataset of polyphonic samples created with three electric guitars formed the dataset in [Dittmar *et al.* 2013].

Several authors [Jadhav *et al.* 2022; Kim *et al.* 2022; Wiggins and Kim 2019; Maaiveld 2021; Cwitkowitz *et al.* 2023] used a convolutional neural network to detect string-fret combination, achieving accuracies in the range of 0.82-0.92. While only the results of [Jadhav *et al.* 2022] were explicitly reported using the accuracy metric, all results essentially refer to the same success rate. The precision metric used by Maaiveld [2021], and Cwitkowitz *et al.* [2023] refers to the amount of correct string-fret detections within the pool of detected notes, and the *Tablature Disambiguation Rate* (TDR) defined by Wiggins and Kim [2019] and used by Kim *et al.* [2022] represents what in this Master's Thesis is defined as the string-fret combination detection accuracy. All five papers, except for the one by Jadhav *et al.* [2022], which also included chords from the Montefiore dataset, utilized the GuitarSet dataset exclusively.

3 DATASET

The dataset created for this study consists of 10099 monophonic audio samples collected from three different electric guitars. The three instruments used for data collection come from two different manufacturers and differ in weight, scale length, fret spacing, pickup type, string *action* (distance from fretboard) and string set gauge, as shown in Table 1.

Guitar model	Scale length	Frets	String gauge	Pickup	Plectrum position
Ibanez 440S with Floyd Rose bridge	648 mm	22	010" - 050"	Middle SC	Middle SC
ESP LTD EC-200QM	648 mm	24	010" - 046"	Neck HB	Between neck HB and bridge HB
Ibanez PGMM21 Paul Gilbert MGN	564 mm	24	009" - 042"	Bridge HB	Between neck HB and bridge HB

Table 1: Properties of the electric guitars used for dataset creation. SC denotes single coil pickup, HB denotes humbucker pickup.

3.1 Data collection procedure

Raw audio was recorded directly from the output jack of each instrument into an audio editor using 48kHz sample rate and 16-bit depth. Multiple samples of differing durations were collected from each string and fret, such that an entire string's worth of samples was contained in a single recorded file. Each audio file was manually named after the instrument and string that produced it.

Playing technique was limited to downstrokes and upstrokes performed with a plectrum, and the picking position (see Table 1) chosen for each instrument remained constant throughout the recordings. All three guitars were in standard EADGBE tuning.

3.2 Data cleaning

The Python packages **librosa** [McFee *et al.* 2023] and **numpy** [Harris *et al.* 2020] were used at the data cleaning stage for a number of tasks. For each recorded audio file, DC offset was removed and silence was trimmed from file beginning and end. The trimmed recordings were then sliced into single-note segments using the librosa.onset module and segments shorter than 1024 samples were discarded to ensure a DFT window length 512 for audio downsampled to 24 kHz (see definition of window length in Section 4.1.1). The remaining segments were normalized between [-1, 1]. In order to remove segments that had no detectable pitch, fundamental frequency (f_0) detection was performed on each segment using the **librosa.pyin** implementation of the PYIN f_0 estimator [Mauch and Dixon 2014]. Segments with no detectable f_0 were discarded and the remaining segments were saved to disk as files that contained a single note each and carried instrument and string information in the filename. See Figure 11 for an overview of the data cleaning process.

Possible misdetections were identified by the following method: each sample with a detected fundamental frequency f was matched with the musical pitch whose frequency was within a quarter-note of f (the formula for calculating the quarter-note ratio is presented in Section 1.2.2). Samples were then grouped by musical pitch into bins containing only a single pitch, e.g. E_2. Each bin was then verified to contain the expected frequency only, by playing the samples back-to-back to a human validator (i.e. the author) as a monotonous pitch sequence, from which any deviating pitches or noisy samples were easy to identify and remove by hand.

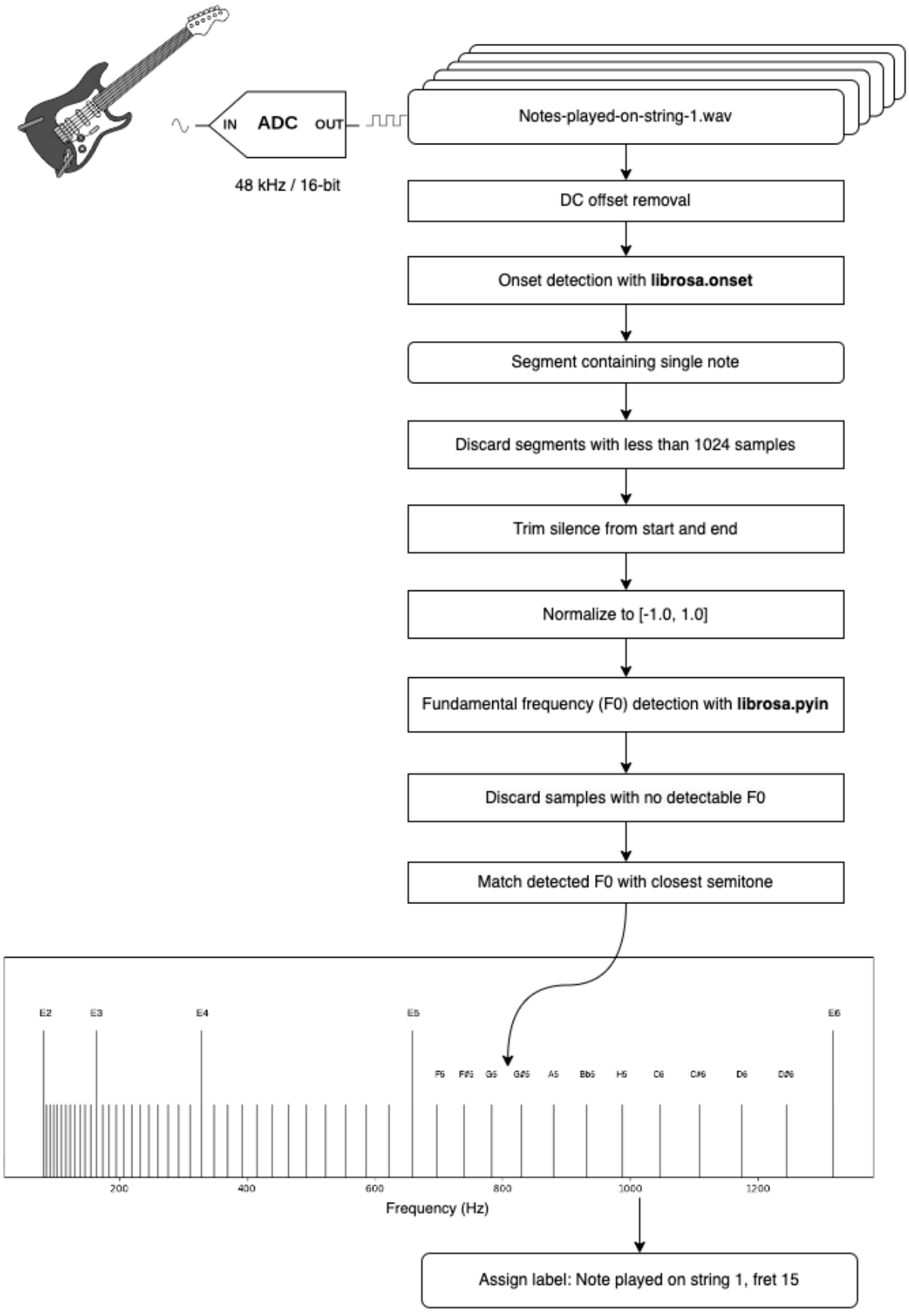


Figure 11: Data collection and labeling procedure

At this point, each audio sample in the dataset contained a single note produced by a known string and instrument, but lacked fret information. To generate fret labels, a **pandas** [Reback *et al.* 2021] DataFrame representation of a 24-fret guitar fretboard was constructed and populated with the musical pitches expected from a guitar in standard EADGBE tuning. Each sample originating from string S was then matched with the fret whose pitch matched the sample's detected pitch.

The resulting 10099 samples span the musical pitches E_2-E_6 for the 24-fret guitars and pitches E_2-D_6 for the Ibanez 440S, which had only 22 frets.

4 METHODS

Audio samples in the dataset were feature engineered into two-dimensional format using three different transforms: a spectrogram, the Mel-spectrogram and the constant-Q transform. One convolutional neural network was trained for each transform to detect which of the guitar strings (EADGBE) was used to produce each original sample.

4.1 Feature engineering

4.1.1 Discrete Fourier transform

All of the features used in this Master's Thesis are based on the discrete Fourier transform (DFT) [Allen 1977], which is an algorithm for calculating the frequency domain representation of a signal.

Natural sounds can be thought of as consisting of multiple cosine waves superimposed on each other, and the Fourier transform [Osgood, 2009] offers a way to decompose a sound into its frequency components, yielding a *frequency spectrum*. See Figures 5b and 6b for examples of frequency spectra. From a musical perspective, a Fourier transform yields the relative intensities of different pitches and their harmonics, which is useful in string detection, considering the differences in tension and thickness of the strings on which a given musical note can be played.

In the context of digitally sampled signals, the discrete Fourier transform (see Equation 4) is used instead of the original Fourier transform. In Equation 4, N represents the length of the signal in samples and ω refers to a component frequency or "frequency bin" under observation. $X(\omega)$ is calculated for every position in the signal, and the output of each calculation is a complex number. The resulting complex-valued vector of size N constitutes the digital Fourier transform. The magnitude of each complex number represents the magnitude of the frequency bin under observation, while angle represents phase. For the purposes of this Master's Thesis, only the DFT magnitude is used.

$$X(\omega) = \sum_{n=0}^{N-1} x(n) e^{-i 2\pi \omega n / N} \tag{4}$$

The frequency resolution of DFT is dictated by the length and sample rate of the signal under analysis. If a signal is too short for a low frequency to complete its cycle, said frequency will not appear in the DFT, while high frequencies will be subject to *aliasing* [Robinson and Clark 1991]. Aliasing occurs when an analog-to-digital (A/D) converter with a sampling rate of *f* attempts to sample a signal having frequency 0.5*f* or higher. While a longer wavelength is sampled several times during its cycle, higher frequencies approaching 0.5*f*, also known as the *Nyquist frequency*, have short wavelengths that are sampled only a few times per cycle. The results of lowering the sample rate and thus lowering the Nyquist frequency can be seen in Figure 12, which shows four different frequencies sampled at 10 kHz on the left and 100 Hz on the right. Longer wavelengths retain their characteristics when sampled at a lower rate, while higher frequencies suffer.

Since the audio data used in this Master's Thesis is downsampled during preprocessing to 24 kHz, the Nyquist frequency is 12 kHz. The highest harmonics of an electric guitar reach the 6-8 kHz region, so this is more than enough for our purposes. The lowest frequency of a guitar in standard tuning is produced by the open E_2 string, which has a frequency of roughly 82 Hz. However, when considering the problem of string detection, frequencies with an unvarying origin point on the fretboard are not strictly required as part of the dataset. This is the case with pitches E_2-$G\#_2$ and C_6-E_6 , which appear only on the 6th and 1st strings of a guitar in standard tuning. For segments containing these pitches, fundamental frequency detection alone will yield the string-fret combination, as there is only one string that could have produced them. The effective range of interesting frequencies starts therefore from 110 Hz (A2) instead of 82 Hz.

The time that it takes for a periodic 110 Hz signal to complete one cycle is ~9.1 milliseconds, which corresponds to ~436 samples when using a 48 kHz sampling frequency and even less for a signal sampled at 24 kHz. A DFT window length of 512 is therefore sufficient for both sampling frequencies. The Short-time Fourier transform (STFT) [Durak and Arikan 2003] is an effective algorithm for calculating the DFT, and its **librosa.stft** implementation was used in this Master's Thesis.

4.1.2 Feature 1: Spectrogram

The discrete Fourier transform reports the frequency and phase content of an entire signal but offers no temporal dimension. The DFT of a sample played forward and backward is the same, because the frequency spectrum is unchanged. However, the frequency content of most sounds produced by physical objects tends to evolve over time, and this is also true for sounds produced by guitar strings. For the purposes of many sound detection tasks, including the one attempted in this Master's Thesis, a spectrogram is therefore often more useful than a mere DFT.

Figure 13 shows a three-dimensional spectrogram of a sound played on a French horn evolving through time. Only the lower frequency bins are activated at the attack and decay portions of the note, with higher frequency bins active in the middle portion. This corresponds to the characteristic sound of brass instruments in general.

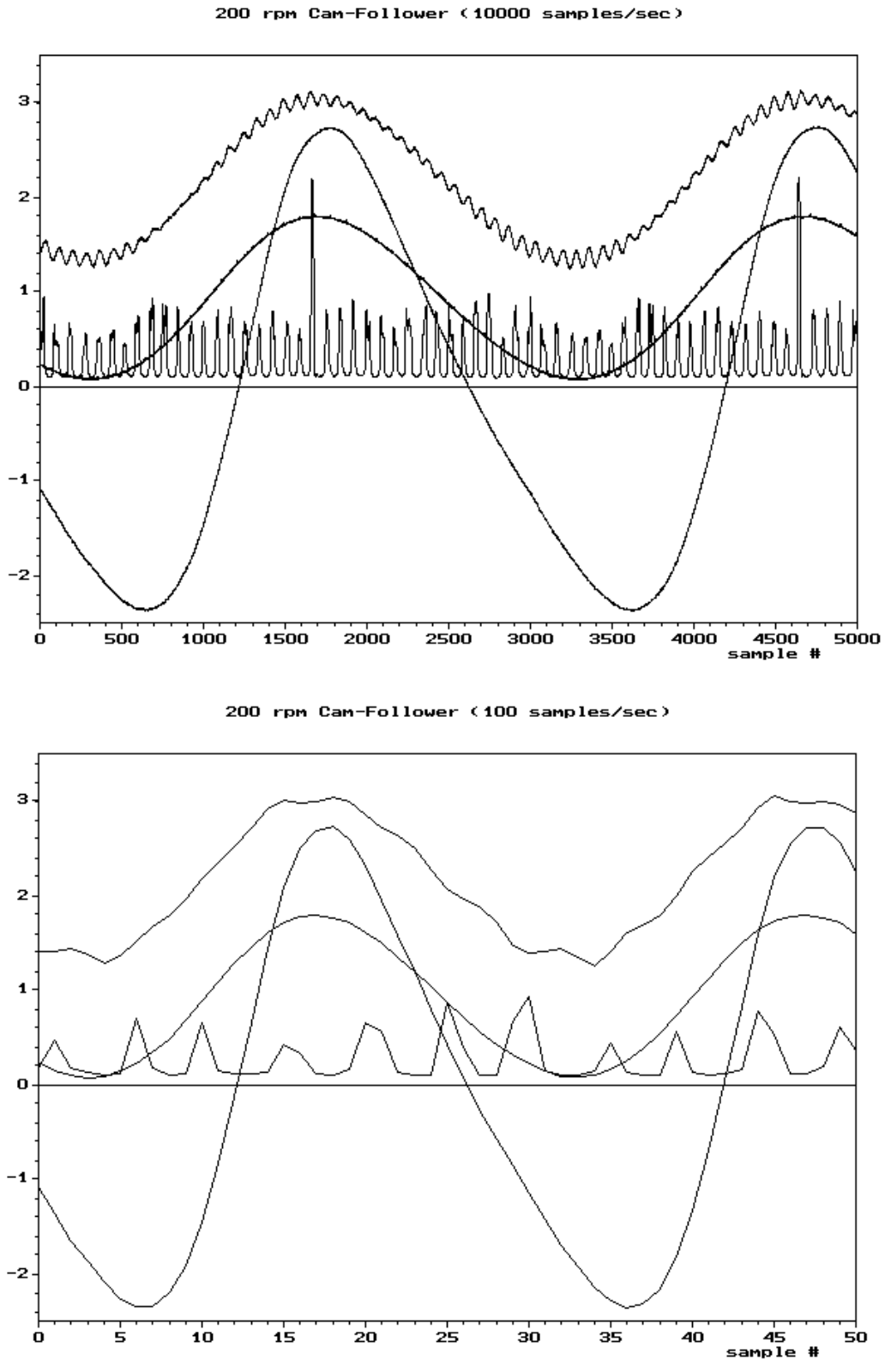


Figure 12: Four waveforms reconstructed from samples taken two sampling rates, 10kHz (top) and 100 Hz (bottom) ["Aliasing", 2023]. The 100 Hz sampling rate is not sufficient to capture the shape of the higher frequencies, while the lower frequencies remain intact.

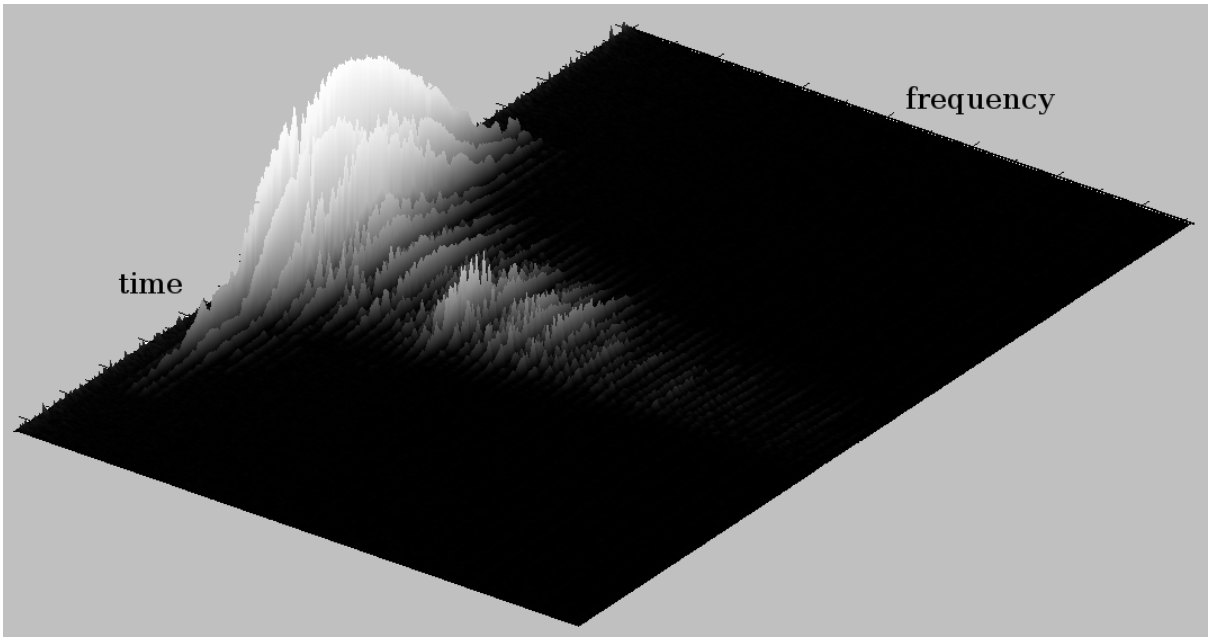


Figure 13: Three-dimensional spectrogram of a sound played on a French horn whose volume first increases and then decreases. The elevation of the diagram at a particular time displays the current amplitude ["3D-Spectrogram French horn", 2008].

In order to show a spectrogram in two dimensions, a heatmap is more convenient. Figure 14a shows the spectrogram of a note played on the open low E string of an electric guitar with the heat map values representing spectral density normalized between [-1, 1]. The image does not contain much visual information. In Figure 14b, the heat map has been converted to Decibel scale (dB) before normalization and more detail is visible. The normalized Decibel scale version of the spectrogram is used for training the CNN in this Master's Thesis.

In order to construct a spectrogram, a time window of constant length (also referred to as a frame) is moved across the signal from beginning to end. Along the way, DFTs of the windowed portion of the signal are calculated and added to the spectrogram as columns. The length of the moving window is referred to as window size, and it determines both the lowest frequency that can be detected from an audio signal via DFT and the number of resulting frequency bins (see Section 4.1.1). Hop size refers to the number of samples between the starting points of each successive window, with a value of one indicating a sliding window.

The resulting spectrogram is a two-dimensional array, where each row contains the DFT magnitudes of one frequency bin. In order to prepare the spectrograms for use as CNN input, the values were converted to Decibel scale and normalized between [-1, 1].

Before constructing the spectrograms, each audio sample was resampled at 24 kHz and truncated to a length of one second, i.e. 24000 samples. Spectrograms were then calculated with **librosa.stft** using a window length of 512 and hop length of 32. The resulting spectrograms having shape 257 x 50 were converted to Decibel scale and normalized between [-1, 1]. See Figure 17.

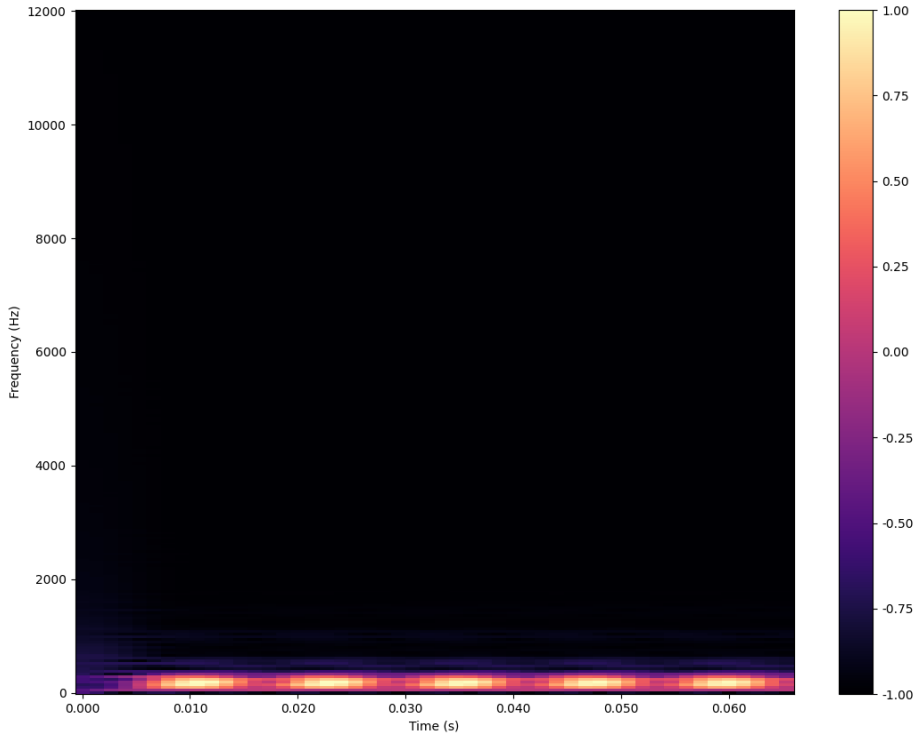


Figure 14a: Two-dimensional spectrogram of the E_2 pitch (82 Hz) played on the open low E string of an electric guitar. The heat map values represent spectral density.

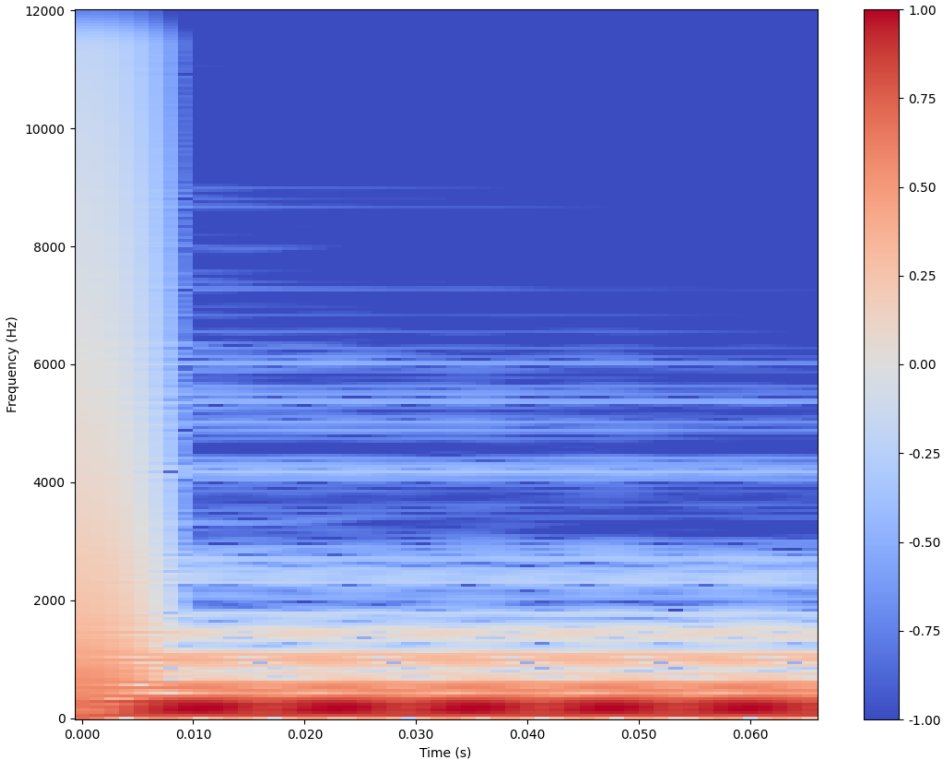


Figure 14b: Spectrogram shown in Figure 14a converted to Decibel scale before normalizing.

4.1.3 Feature 2: Mel-spectrogram

On a *Mel spectrogram*, frequency bins representing consecutive musical pitches (or octaves) have equal spacing on the frequency axis [Dörfler *et al.* 2017]. This is not the case with a regular spectrogram with a linear scale y-axis. As shown in Figure 7, the octave jump between e.g. E_2 (82 Hz) and E_3 (164 Hz) is only 82 Hz, while the octave jump between E_5 (659 Hz) and E_6 (1318 Hz) is a much larger 659 Hz. This leads to cramped spacing of musical pitches at the lower end of the frequency spectrum, visible in the spectrograms shown in Figures 14a and 14b, where the E_2 pitch and its first and second harmonics appear lumped together.

The Mel scale (as in 'melody') is a scale of frequencies that corresponds to human perception of pitch in such a way that successive pitches on the Mel scale sound equally distant from each other. As explained in Section 1.2.3, human perception regarding musical pitch is not linear but logarithmic, and interprets the distance between any frequencies f and $2f$ to be equal. Measuring pitch distances in Hz does not support this mode of interpretation, because a difference of 100 Hz in between two low frequencies sounds considerably larger than the same 100 Hz difference between two high frequencies. A Mel spectrogram is computed by mapping a regular spectrogram S onto Mel scale by calculating M.dot(S), where M is a linear transformation matrix used to project DFT bins onto Mel-frequency bins. The transformation matrix, also referred to as a *Mel filter bank*, is determined by the sampling rate, the Nyquist frequency and number of frequency bins required.

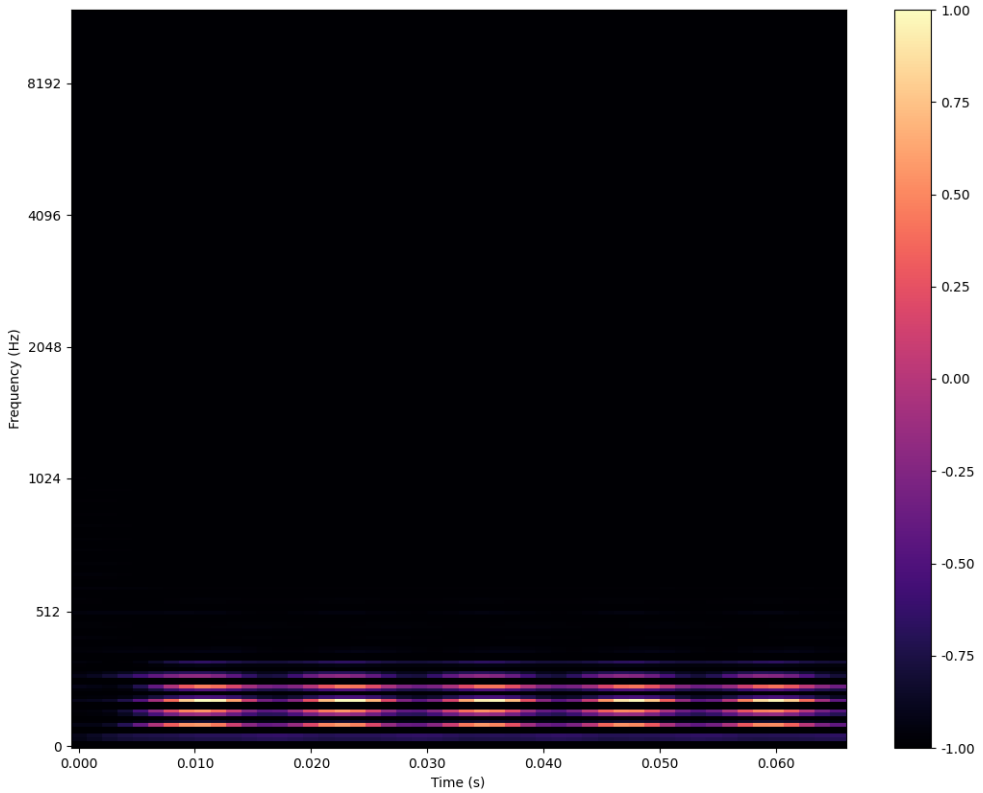


Figure 15a: Mel-spectrogram of the E2 pitch (82 Hz) played on the open E string of an electric guitar with the heat map values representing spectral density.

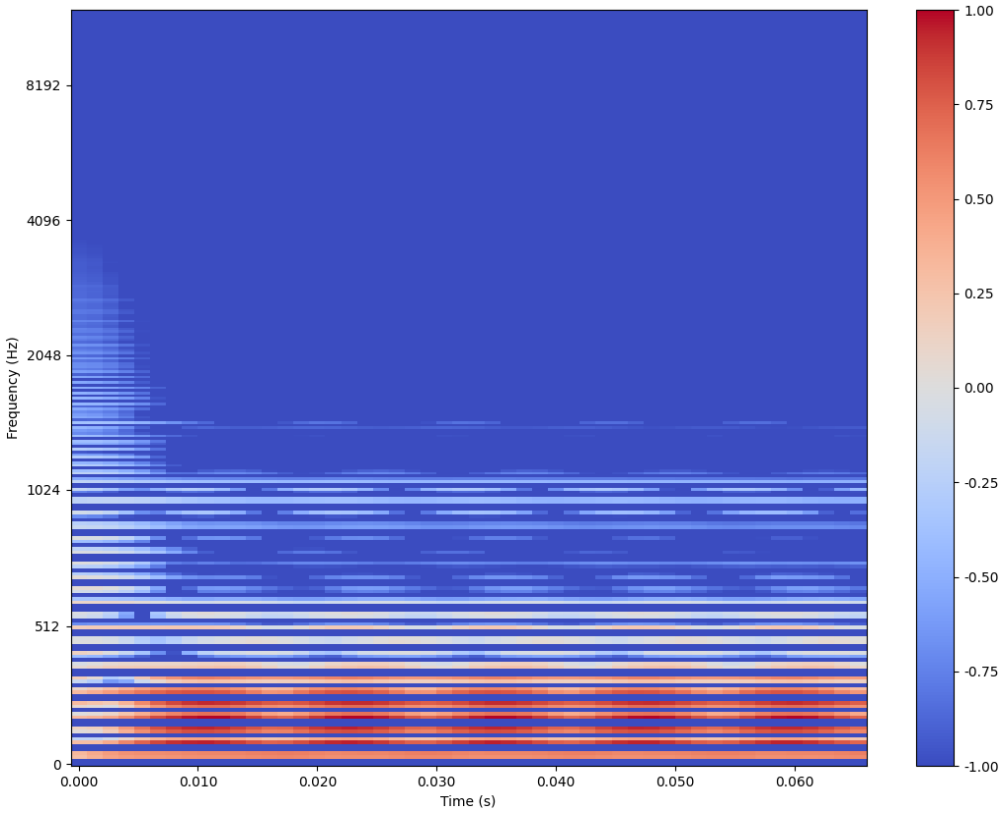


Figure 15b: The Mel-spectrogram shown in Figure 15a converted to Decibel scale.

Figure 15a shows the spectrogram shown in Figure 14a converted to Mel scale and normalized to [-1,1]. Notice the increased resolution of the fundamental frequency and its harmonics when compared to Figure 14a. As with the regular spectrogram, the spectral density values of the Mel-spectrogram were converted to Decibel scale (dB) and normalized between [-1,1] (see Figure 16b) before using the feature as a CNN input.

4.1.4 Feature 3: Constant-Q transform

The *constant-Q transform*, like the Mel-spectrogram, is the output of a bank of filters with geometrically spaced center frequencies to account for the non-linear spacing of successive musical pitches [Schörkhuber and Klapuri 2010]. What differentiates the constant-Q transform from a Mel-spectrogram is the *Q-factor*, defined as the ratio of a filter's center frequency to its bandwidth, which is constant for all frequency bins. This is not the case with Mel-spectrograms or spectrograms, as both use a fixed window that covers few (if any) complete cycles for the lowest frequencies and multiple cycles for higher frequencies.

When using the constant-Q transform, each filter has a width that is a multiple of the previous filter's width, as shown in Equation 5, where δf_k denotes the bandwidth of the kth filter, f_{min} denotes the central frequency of the lowest filter, and n is the number of filters per octave.

Figure 16a shows the constant-Q transform of the E_2 pitch (82 Hz), normalized to [-1-1].

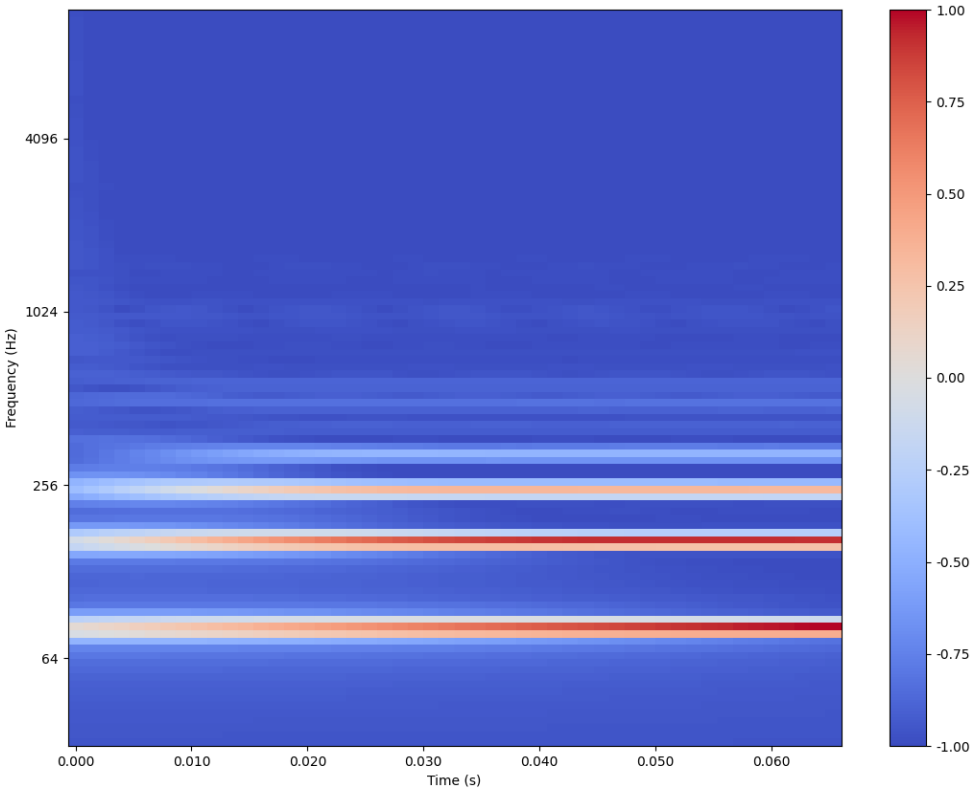


Figure 16a: Constant-Q transformation of the E2 pitch (82 Hz), played on the open E string of an electric guitar with the heat map values representing spectral density.

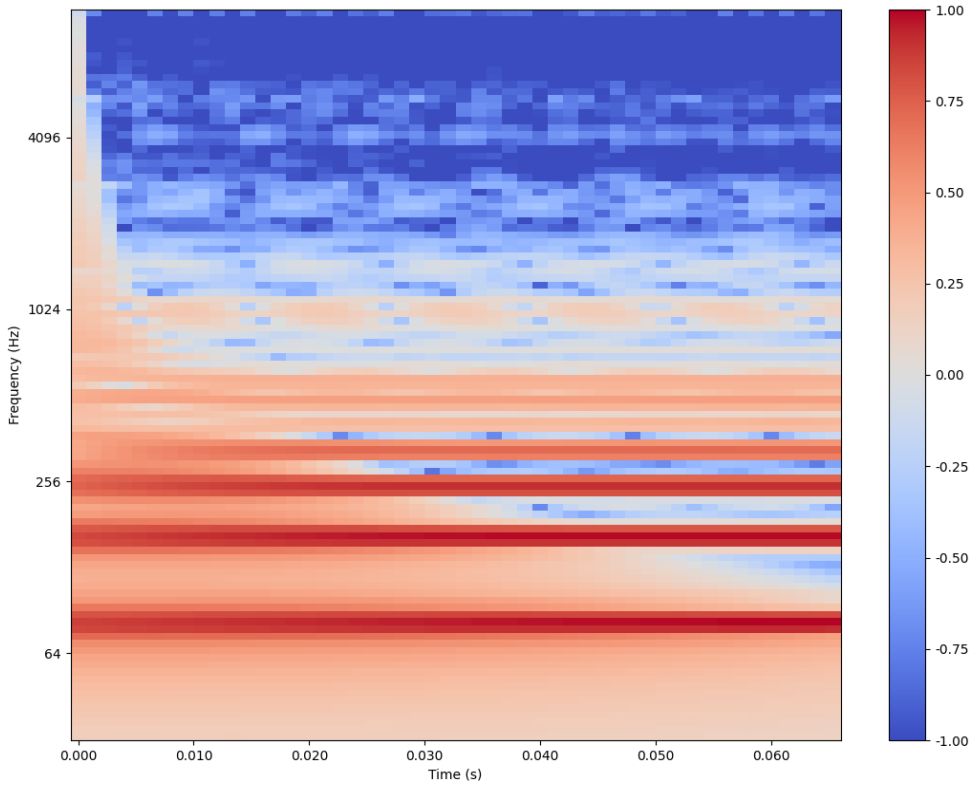


Figure 16b: The constant-Q transformation shown in Figure 16a converted to Decibel scale.Notice the even larger spacing between f_0 and the harmonics, when compared to the Mel spectrogram and spectrogram of the same audio file, shown in Figures 15a and 14a respectively.

$$\delta f_k = 2^{1/n}\delta f_{k-1} = \left(2^{1/n}\right)^k \delta f_{min} \quad (5)$$

As with the previous features, the spectral density values of the constant-Q transform were converted to Decibel scale (dB) and normalized between [-1,1] (see Figure 16b) before using the feature as a CNN input.

4.1.5 Feature engineering algorithm

Because the upper range of an electric guitar note harmonics is in the 6kHz region, a sample rate of 24 kHz, was considered sufficient and the 48 kHz audio segments were resampled at 24 kHz. The resulting Nyquist frequency was 12 kHz, yielding ample room for any high harmonics that could possibly occur in the data. The audio segments were truncated to a length of 24000 samples after downsampling.

As the dataset consisted of audio samples, i.e. time series data, it had to be transformed into a format usable as CNN input. Three different transformations were calculated using the librosa library: a spectrogram, Mel-spectrogram and the Constant-Q transform. All features were built using the **librosa** library and the non-default settings for each are shown in Table 2. Figure 17 shows the individual steps of the feature engineering pipeline.

Because of variation in the lengths of the original audio samples, the representations had varying widths, which had to be truncated or zero-padded to a constant width in order to conform to CNN input shape requirements. After brief experimentation a feature width of 50 was settled on per feature. As a hop length of 32 was used in feature construction, the remaining 50 windows represent the first 50 x 32 = 1600 samples of audio, which equates to the first 67 ms of each audio file when using data downsampled to 24 kHz. This length roughly corresponds to the length of a 16th-note played at a tempo of 240 beats per minute (i.e. a very short note) and represents a minimum signal length obtainable for the majority of played notes in a transcription setting. Using a larger feature width could enhance string prediction accuracy for notes of longer duration, but would correspondingly increase the number of trainable parameters in the CNN.

4.2 Convolutional neural network

A simple convolutional neural network (CNN) was constructed using **tensorflow.keras** [Abadi et al. 2015] and used to train three models. One model was trained on the spectrograms of the audio samples, another on the Mel-spectrograms and a third one on the constant-Q transforms. All models were evaluated using 6-fold cross-validation [Zeng and Martinez 2000] and the average accuracy over all six folds was reported per model in Table 3 (see Section 5).

The CNN architecture is shown in Figure 18 and descriptions of the layers follow in this section.

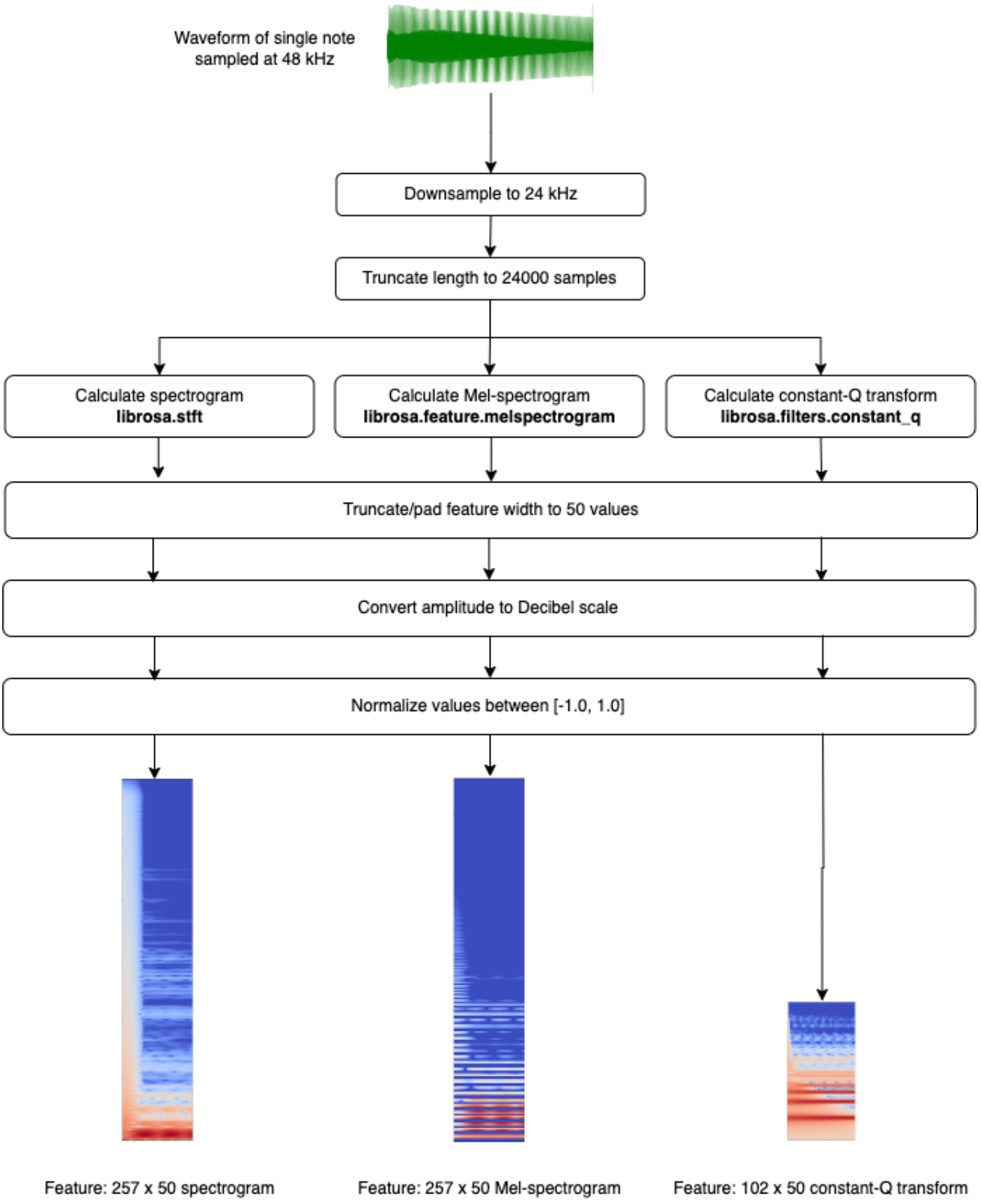


Figure 17: Feature engineering pipeline

Feature	Implementation	Input samples	Non-default parameters	Height	Width
Spectrogram	librosa.stft	24000	n_fft=512, hop_length=32	257	50
Mel-spectrogram	librosa.feature.melspectrogram	24000	n_fft=512, hop_length=32, n_mels=257	257	50
Constant-Q transform	librosa.filters.constant_q	24000	n_bins=102, hop_length=32	102	50

Table 2: Feature specifications. Source audio was truncated to the first 24000 samples before feature construction and the width of the two-dimensional features was truncated to 50 values.

4.2.1 Input layer

The input layer of a CNN consists of image data where each image has a height, a width and a number of channels. All of the three features constructed were two-dimensional grayscale images representing some variation of an audio spectrogram. (The colors shown in Figures 14-16 are the result of mapping grayscale value ranges to color ranges for perceptual reasons, not representations of a 3-channel RGB image). The width and height of the images varied per feature type, as shown in Table 2.

Input layer image height depends on the feature calculation method, e.g. the window size used to perform DFT during feature construction, and truncating or downsampling it would affect resolution. Image width corresponds to the number of time windows used, and represents the duration of the original sample that was covered during feature construction. Height was therefore left untouched, and a width of 50 samples was considered to be sufficient for all features, as described in Section 4.2.

4.2.2 Convolutional layers

A convolutional layer consists of one or more *filters* that move along the input image, calculating dot products between the filter and the image at multiple locations on the image and storing the results in a new matrix a.k.a *feature map* [Li *et al.* 2022]. On a 2D convolution layer, each filter is a three-dimensional matrix of shape *(h, w, c)* where *h* denotes matrix height, *w* denotes matrix width and *c* denotes the number of channels in the image.

Filters consist of one or more *kernels*. When using 2D convolution, a kernel is a two-dimensional matrix that corresponds to a channel in the input image and contains weights that are multiplied with the values of that channel as the filter moves along. For RGB images, filter shape is *(h, w, 3)*, meaning that a kernel of shape *(h, w)* exists for each of the channels R, G and B. When using grayscale images, as in this Master's Thesis, the filter shape is *(h, w, 1)* and, because there is only one channel, the kernel is practically equivalent to the filter. The values of a kernel, like the nodes of a regular feedforward neural network, are learnable parameters and each kernel of each filter is initialized independently from each other.

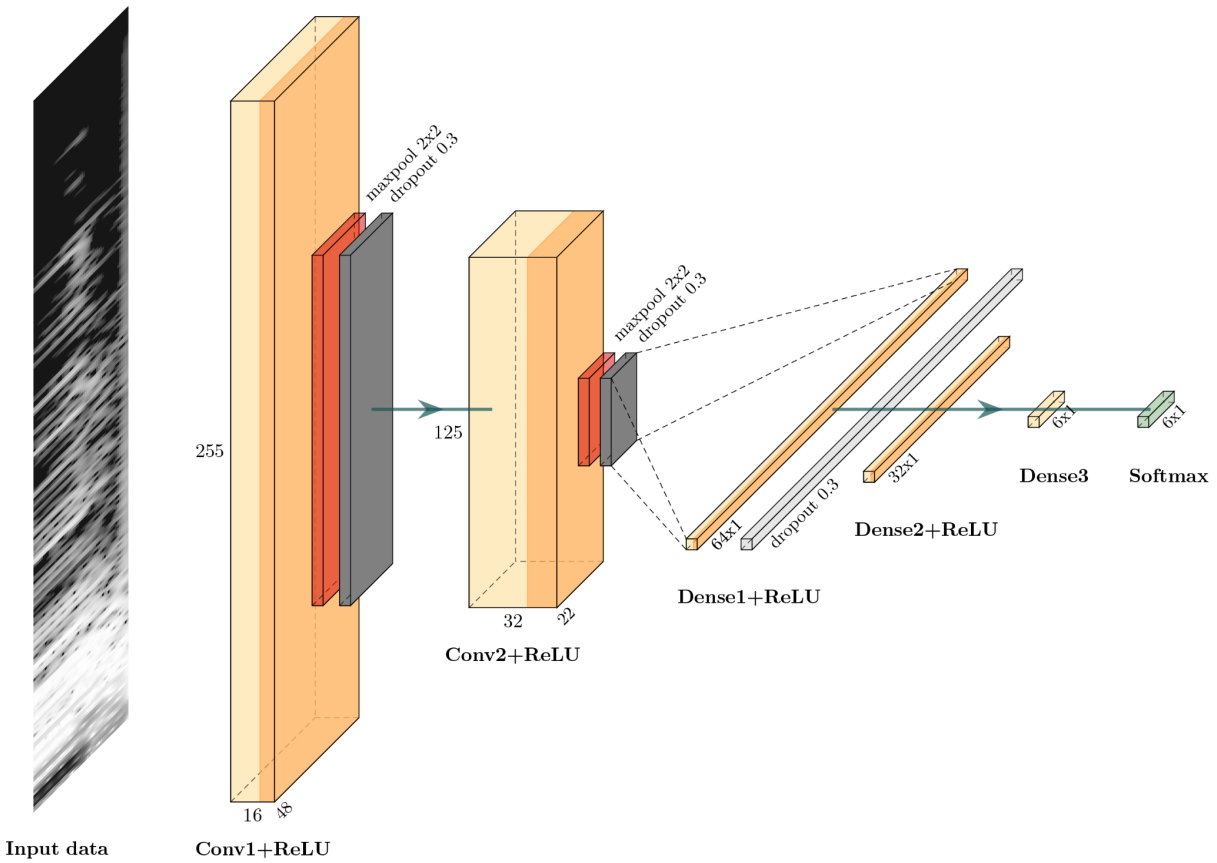


Figure 18: Convolutional neural network architecture. Convolutional layer dimensions shown apply to the models trained on spectrogram and Mel-spectrogram data. For the model trained on constant-Q transform data, Conv1 and Conv2 dimensions were 100x48 and 48x22 respectively. A kernel size of 3x3, a dropout rate of 0.3 and max pooling size of 2x2 were used throughout.

In convolution, each filter starts at the top left corner of an image where its values are multiplied with the values of the corresponding part of the image, referred to as the receptive field, using the dot product. The filter then slides to the right onto the next *receptive field*, the location of which is determined by the *stride* parameter, which indicates the number of matrix columns to move at each step. In 2D convolution, stride is defined for height and width separately.

The dimensions of a filter, together with its stride values, define the dimensions of the filter's output matrix. Figure 19 shows an example filter with dimensions 3x3 traversing a 5x5 input image. Using a vertical stride of 1 and a horizontal stride of 2, the filter yields a 3x2 output matrix after convolution is complete. However, as can be seen from Figure 19, the values towards the edges of the input image are included in a receptive field less often than values towards the center. It follows that the dimensions of the filter's output matrix are also smaller than the original image. In order to include edge values better and to preserve dimensionality, the input image can be extended by padding the edges of the image symmetrically with zero values.

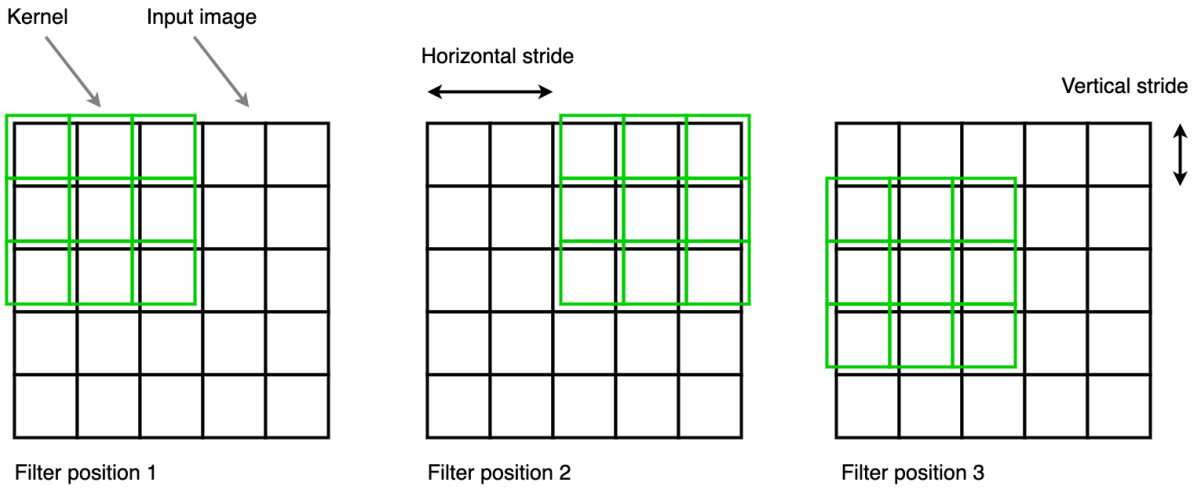


Figure 19: Movement of a 3x3x1 convolutional filter across a 5x5 single-channel image using vertical stride of 1 and horizontal stride of 2.

If a 2D convolution layer has more than one filter, each travels across the input image independently. Each filter's output values are passed through an activation function (ReLU in our case) and collected into a two-dimensional matrix. The matrices are then stacked on top of each other and passed on to the next layer in the CNN as an input.

The model used in this Master's Thesis contains two 2D convolutional layers with 16 and 32 filters respectively. A kernel size of 3x3, a horizontal and vertical stride of 1 and a ReLU activation function is used for both.

4.2.3 MaxPooling 2D layer

A pooling layer downsamples its input data by sliding a window, also referred to as *pool*, across the input and outputting a single value for each window position. The available functions are *average pooling*, which outputs the window average, and *max pooling*, which outputs the maximum value found in the window. Stride and padding parameters (see Section 4.2.2) are available as well.

A max pooling 2D layer with a pool size of 2x2 was used in the CNN after each Conv2D layer (see Figure 18).

4.2.4 Dropout layer

When using high-dimensional data, overfitting becomes an issue. This is also true for the spectrogram variants used in our case, which contained up to 257 x 50 = 12850 values per training data point. In an effort to make the CNN generalize better, dropout layers were added with a rate of 0.3 to stages shown in Figure 18, after experimenting with rates in the 0.2-0.4 range. Dropout layers randomly silence a fraction of the nodes (determined by *rate*), forcing

the network to generalize, as it cannot rely on all the nodes being consistently available [Baldi and Sadowski 2013].

4.2.5 Dense layer

Dense i.e. fully connected layers with a decreasing number of nodes were used towards the end of the neural network architecture before the final output layer. Dropout layers were again added between these layers, as seen in Figure 18.

4.2.6 Output

The final output layer of the neural network contained six nodes, one per guitar string. The output of the CNN was a 6x1 vector, representing the strings of a typical six-stringed electric guitar. The activation function used was softmax.

Because the experimental setup of this Master's Thesis involved a clean dataset and the emphasis was not on onset detection or f_0 detection, there was no need for a null class. In a realistic Automated Guitar Transcription system, different kinds of noise artifacts would probably get past the onset detection stage and the string recognition stage might have to learn a seventh label for samples with no string information present. Polyphony would have to be accounted for, and there would be a need for a multi-label classifier (as opposed to the multiclass classifier presented in this Master's Thesis).

4.3 Training and evaluation

Three models, one per feature type, were trained using sparse categorical cross entropy loss function and evaluated using stratified six-fold cross-validation on the CNN shown in Figure 18.

In order to establish an estimate of the models' accuracy, each was subjected to stratified k-fold cross-validation [Zeng and Martinez 2000] during training. The spectrogram data was split into six subsets of similar size, as shown in Figure 20, and data from all strings was included in each subset. One subset was kept aside as a test set and the five remaining subsets were merged to form a training set on which the model was trained for 100 epochs. All hyperparameters except for filter count, kernel shape and activation type were left at their default values because of time constraints.

After training for 100 epochs, the validation accuracy from the fold was stored. The test set was then swapped with another subset and the entire training-evaluation procedure was repeated, until all of the subsets had been held aside for testing purposes exactly once. The classification accuracies of all six training episodes were then averaged and reported in Table 3. The entire procedure was repeated for Mel-spectrograms and constant-Q transforms, resulting in three trained CNN models that had all been evaluated using cross-validation. See Sections 5 and 6 for results and discussion.

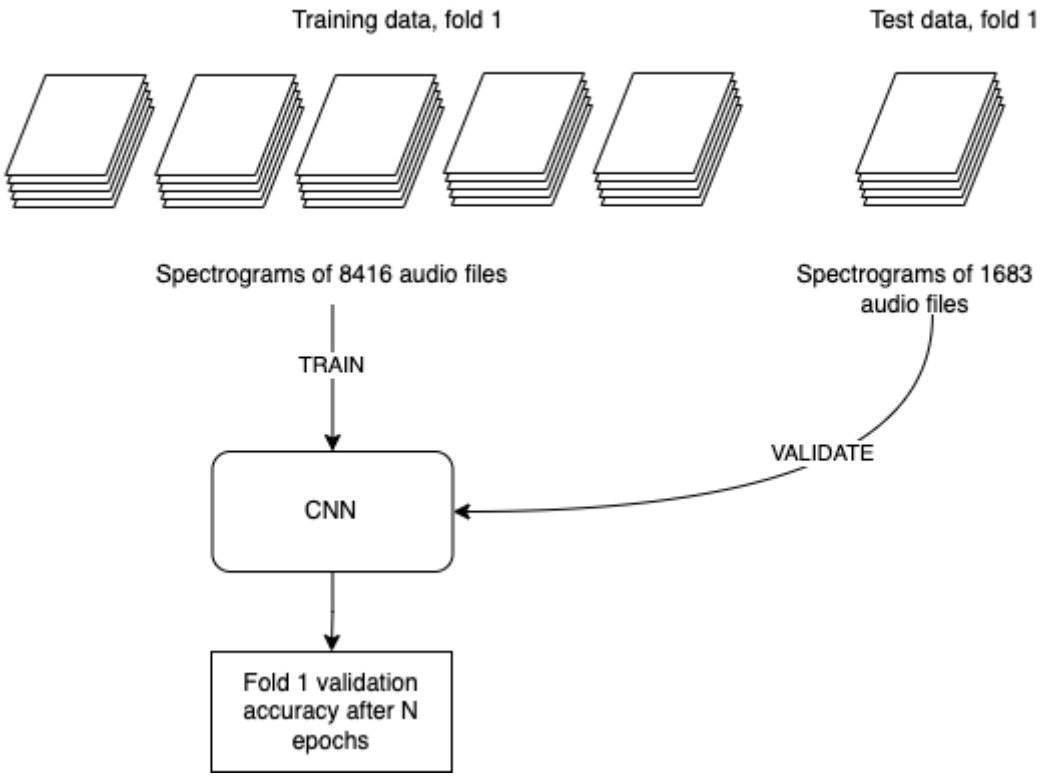


Figure 20: 6-fold cross-validation of model trained on spectrogram data, fold 1. For the next fold, the test data subset is swapped with one of the training data subsets until all have been used for validation and training.

4.4. Prediction

After training was completed, the weights of the final models were stored. The best-performing model (i.e. the one trained on CQT transforms, see Table 3) can be used to predict successive string-fret combinations from monophonic electric guitar recordings by cloning the repository [Simola, 2023], installing the requirements and following the instructions in the README.

This command runs the pipeline shown in Figure 21, which preprocesses the input file and predicts string-fret combination for each individual note using the model trained on the constant-Q transform. The resulting sequence of string-fret combinations is written to a text file in the format shown in Figure 22. This output is incomplete, as note duration information is missing entirely and the layout is not very user-friendly, but resembles tablature available on websites such as Ultimate Guitar - transcriptions from which were used by Burlet and Hindle [2017] for constructing a synthetic dataset from guitar samples. In order to produce professional quality tablature transcription, note duration detection should be incorporated into the algorithm and results should be presented in a more visually appealing format. See discussion on future work in Section 6.

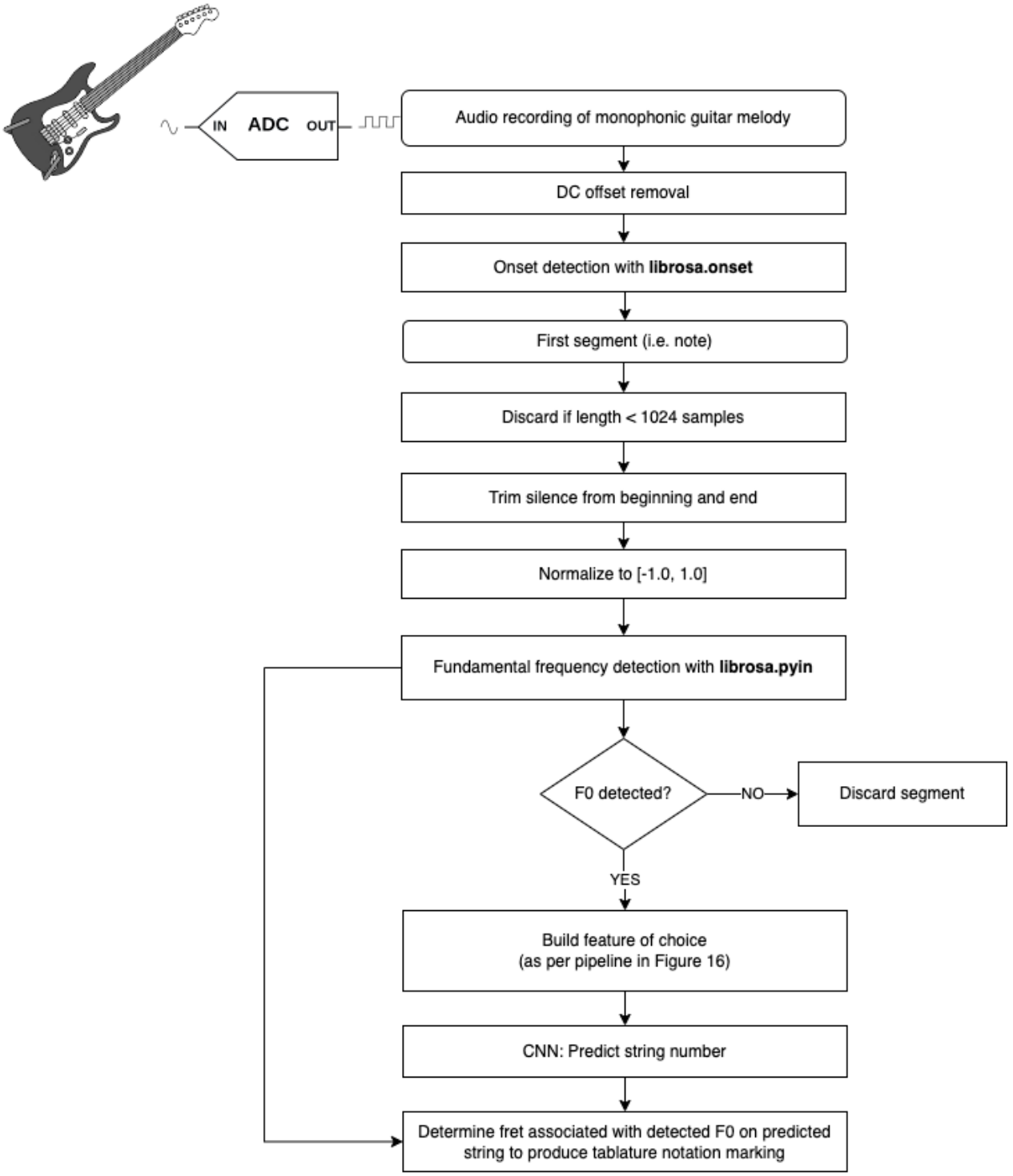


Figure 21: Prediction pipeline.

```
s1 ---------------------------------------------8----------
s2 -------------------------------------------------10-------
s3 -5--------------6-5-3-5-3---1-3-5-------9------------10-9-7-
s4 ---5-4-5-4-2---------------3---------4-5------10-------------
s5 -------------5-3-------------------3-------10---------------
s6 ------------------------------------------------------------
```

Figure 22: String-fret combinations of a monophonic melody detected by the algorithm. The output contains no duration information.

5 RESULTS

The average string detection accuracies achieved with the CNN models trained on each of the feature sets using stratified 6-fold cross-validation are presented in Table 3, along with the average accuracies reported by previous works.

It should be noted that previous studies have used a variety of datasets and metrics, which makes direct result comparisons problematic. For example, the GuitarSet dataset used by Kim *et al.* [2022]; Maaiveld [2021]; Wiggins and Kim [2019]; Cwitkowitz *et al.* [2023] and Jadhav *et al.* [2022] consists of samples from multiple guitarists using a range of playing techniques on a single acoustic guitar. In this Master's Thesis, a dataset collected by a single guitarist using a plectrum and three different electric guitars was used. It is also not necessarily sensible to compare a study with pure string detection scope to studies that attempt to implement an entire transcription pipeline, or transcribe polyphonic audio [Dittmar *et al.* 2013; Jadhav *et al.* 2022; Kim *et al.* 2022; Maaiveld 2021; Wiggins and Kim 2019]. Results from such studies are presented nevertheless for context when accuracy or a metric comparable to accuracy (see Section 2.2.2) is given.

Author	Dataset	Instruments	Polyphony	Approach	Accuracy
[Abeßer, 2013]	IDMT-SMT-Audio-Effects	2 x electric	poly	Custom features + SVM	0.93
[Barbancho et al. 2012]	Custom + RWC	13 x various	mono	Inharmonicity analysis	0.997*, 0.75
[Boloyous et al. 2021]	Custom	1 x acoustic	mono	Inharmonicity analysis	0.87
[Cwitkowitz et al. 2023]	GuitarSet	1 x acoustic	poly	CQT + CNN	0.805
[Dittmar & Abeßer 2013]	Custom	3 x electric	poly	Custom features + SVM	0.92
[Hjerrild & Christensen 2019]	Custom	1 x ac. 1 x el.	mono	Parametric pitch estimator + MAP	0.97*
[Jadhav et al. 2022]	GuitarSet + Montefiore	unknown	poly	CQT + CNN	0.887
[Kim et al. 2022]	GuitarSet	1 x acoustic	poly	CQT + CNN with attention mechanism	0.921
[Maaiveld 2021]	GuitarSet	1 x acoustic	poly	CQT + CNN	0.819
[Simola 2023]	Custom	3 x electric	mono	Spectrogram + CNN	0.923
[Simola 2023]	Custom	3 x electric	mono	Mel-spectrogram + CNN	0.832
[Simola 2023]	Custom	3 x electric	mono	CQT + CNN	0.932
[Wiggins & Kim 2019]	GuitarSet	1 x acoustic	poly	CQT + CNN	0.899

Table 3: Results comparison. Accuracies marked with an asterisk were achieved with methods requiring prior estimates or samples from the strings they were used on. Barbancho *et al.* [2012] reported results separately for monophonic and polyphonic data, but only monophonic results are shown here.

6 DISCUSSION AND CONCLUSION

6.1 Discussion and future work: Dataset

As described in Section 3, the custom dataset used in this Master's Thesis is strictly monophonic and was collected in a way that does not represent natural guitar playing. The notes were recorded from one string at a time in order to speed up the labeling task, and as a result there was practically no overlap between subsequent notes. Three different instruments with varying physical properties were used, which is an improvement compared to the datasets used in most previous works - namely GuitarSet, which was recorded using a single(acoustic) guitar fitted with a custom hexaphonic microphone for labeling purposes [Xi 2018]. On the other hand, among GuitarSet's many merits is the fact that several human players contributed their playing styles to it, which was not the case with the dataset used in this Master's Thesis, which involved only one human player (the author) and a restricted range of playing techniques.

When considering potential future work, the dataset built for this Master's Thesis has an upside however. Many of the existing datasets consist of notes extracted from actual played melodies, which follow familiar patterns and conventions, and all string-fret combinations are not equally represented. Notes appear in these datasets if they happen to be part of a melody or chord progression, and some key signatures, patterns and transitions are more prevalent in guitar music than others. As a result, datasets used in previous work rarely contain dictionaries of multiple notes from every single string-fret combination of several guitars, as our dataset does.

In future work, the dataset constructed for this Master's Thesis could be used as a set of building blocks for constructing a wide range of note combinations that are not limited to existing compositions, commonly used key signatures or the technical abilities of specific guitar players. These synthesized chords and progressions would of course lack natural transitions between notes and lack usefulness in cases where focus is on onset detection, but they could be useful for studying string-fret combination detection.

It is also worth considering the possible use cases for an automatic guitar transcription system. For most guitar players, a system that creates tablature from existing multi-instrument recordings would be in high demand. The complexity of the task is formidable however, and all previous research on AGT systems that the author was able to find focuses on the transcription of isolated guitar recordings instead.

When considering the remaining possible users of a system that is limited to transcribing tablature from isolated guitar recordings, the features expected from the system and the dataset may change. Teachers or composers dictating exercises and new compositions do not suffer from the limitations imposed by existing recordings, as they can produce the audio at will, adjusting tempo if required. They might not require a transcription system that is able to handle every possible sound variation and color that electric guitars, amplifiers and other sound processors can provide, but could be satisfied instead with a system that reliably transcribes carefully articulated and clean sounds into tablature form.

Lastly, as the pure sound of electric guitars differs greatly from that of acoustic and classical guitars, trying to build transcription systems that work for all guitar types may not be a feasible goal. The structural and acoustic differences between hollow-bodied (acoustic/classic) and solid-bodied (electric) guitars are considerable, and further exacerbated by differences in playing technique. Acoustic and classical guitars are often played with fingers, while electric guitars are mostly played with a plectrum, which is essentially a piece of plastic and has considerable effect on the sound color. String materials vary as well, with some acoustic and classical guitars using nylon strings instead of the nickel-plated steel strings commonly used in electric guitars. As mentioned in Section 1, string-fret detection seems to yield better results with acoustic guitars than electric guitars when both scores are reported separately [Hjerrild *et al.* 2019b; Barbanhco *et al.* 2012]. In light of these observations, the benefits and drawbacks of treating all types of guitars equally when designing an AGT system could perhaps be reconsidered.

6.2. Discussion and future work: Methods

The use of CNNs may be overkill for a task as simple as detecting string-fret combinations from monophonic electric guitar recordings. As reported in Sections 2.2.2 and 5, other methods have been used successfully and some previous studies [Barbancho *et al.* 2012; Hjerrild and Christensen 2019] achieved near-perfect accuracy on the string-fret detection task on monophonic data when physical characteristics of the guitar and the strings were known in advance. Considering that most guitar players use a manageable collection of guitars and can afford to provide a minimum of training material for an AGT system, this is not an unreasonable prerequisite. Yet when taking into account the challenges of polyphonic tablature transcription and the promising results achieved in the area using machine learning [Abeßer 2013; Dittmar *et al.* 2013; Kim *et al.* 2022; Wiggins and Kim 2019], CNNs and other machine learning methods are likely to keep playing a role in automatic guitar transcription.

The methods used in this Master's Thesis could have been improved with more systematic CNN architecture search and hyperparameter tuning, as barely any was attempted and the results should be considered a mere baseline. Using an autoencoder on the different spectrogram variants could have helped in dealing with the large feature space by providing latent representations with greatly reduced dimensionality.

The classification task could also have been fine-tuned by minimizing the effect of fundamental frequency on the detection task. As it stands, the CNN used in this Master's Thesis considers every string to be a possible source of the audio sample under observation, and spends part of its training time learning tasks that are already solved by fundamental frequency detection. The CNN should perhaps be allowed to focus on its main task, i.e. differentiating between the sources of identical fundamental frequencies that originate from different strings. Learning that energy in the low frequencies correlates negatively with high-pitched strings or vice versa corresponds to events in physical reality, but does not help the network focus on what exactly differentiates a given fundamental frequency played on one string from the same fundamental frequency played on other possible strings. The differentiating features may have much more to do with harmonics and/or inharmonic content

in high frequency ranges than fundamental frequency. Allowing the CNN to use f_0 (which is prominent and readily available for detection in all of the spectrogram variants) as a feature may divert its attention from the main task.

One possible way to disrupt the f_0-related associations made by the CNN could be the use of non-standard tunings, including the dropped-D or dropped C# tunings. A capo could be used to clamp all strings down on a chosen fret, effectively shortening the scale and giving a different sound character to strings that ring open. Incorporating samples that break up the standard tuning and normal associations between string-fret combination and musical pitch could possibly be used to mitigate the degree to which the CNN relies on fundamental frequency as a feature.

6.3 Conclusion

As described in Section 1.3, detecting string-fret combinations from monophonic guitar recordings is a subtask related to Automatic Guitar Transcription. The subject of this Master's Thesis was to validate the possibility of using a CNN to accomplish this subtask, not to generate a complete end-to-end AGT system. Judging by the results shown in Table 3, this modest goal was accomplished: the model trained on constant-Q transforms achieved an average cross-validated accuracy of 0.932, with minimal hyperparameter tuning and short training times on a laptop with 32 Gb of memory and no compatible GPU.

The obvious shortcomings of this Master's Thesis are related to the limitations imposed by the monophonic dataset, and the ensuing focus on monophonic string-fret detection only. In a real usage scenario, polyphonic audio is the norm and monophonic audio is the exception. Furthermore, in order to construct an actual AGT system, other tasks outlined in Section 1.3, such as onset detection and polyphonic pitch detection, would have to be addressed as well.

7 REFERENCES

Abeßer, J. 2013. Automatic string detection for bass guitar and electric guitar. In: Aramaki, M., Barthet, M., Kronland-Martinet, R. and Ystad, S. (eds.), *From Sounds to Music and Emotions. CMMR 2012. Lecture Notes in Computer Science 7900.* Springer, 333-352. https://doi.org/10.1007/978-3-642-41248-6_18

Abeßer, J. 2014. *Automatic transcription of bass guitar tracks applied for music genre classification and sound synthesis.* Ph. D. Dissertation, Department of Electrical Engineering and Information Technology, Ilmenau University of Technology.

Allen, J. 1977. Short term spectral analysis, synthesis, and modification by discrete Fourier transform. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 3, 235-238. doi: 10.1109/TASSP.1977.1162950.

Baldi, P. and Sadowski, P.J. 2013. Understanding dropout. In: Burges, C.J., Bottou, L., Welling, M., Ghahramani, Z.and Weinberger, K.Q. (eds.), *Advances in Neural Information Processing Systems 26 (NIPS 2013). https://proceedings.neurips.cc/paper_files/paper/2013*

Barbancho, I., Tardon, L. J., Barbancho A. M. and Sammartino, S. 2009. Pitch and Played String Estimation in Classic and Acoustic Guitars. *Audio Engineering Society,* Convention Paper 7701.

Barbancho, I., Tardon, L. J., Sammartino, S. and Barbancho, A. M. 2012. Inharmonicity-based method for the automatic generation of guitar tablature. *IEEE Transactions on Audio Speech and Language Processing* 20 (6), 1857-1868. doi: 10.1109/TASL.2012.2191281

Boloyos, M.K., Libunao, T. K., Masilungan, J., de Leon, F., Lucas, C. R. and Tolentino, C. T. 2021. Monophonic Audio-Based Automatic Acoustic Guitar Tablature Transcription System with Legato Identification. In: *ENCON 2021 - 2021 IEEE Region 10 Conference (TENCON)*, 516-521. doi: 10.1109/TENCON54134.2021.9707430.

Burlet G and Hindle A. 2017. Isolated guitar transcription using a deep belief network. *PeerJ Computer Science* 3 (109). https://doi.org/10.7717/peerj-cs.109

Cwitkowitz, F., Hirvonen, T. and Klapuri, A. 2023. FretNet - Continuous-Valued Pitch Contour Streaming for Polyphonic Guitar Tablature Transcription. In: *CASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1-5. doi: 10.1109/ICASSP49357.2023.10094825

Dittmar, C., Männchen, A. and Abeßer, J. 2013. Real-time guitar string detection for music education software. In: *14th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, 1-4. doi: 10.1109/WIAMIS.2013.6616120

Durak, L. and Arikan, O., 2003. Short-time Fourier transform: two fundamental properties and an optimal implementation. *IEEE Transactions on Signal Processing*, *51* (5), 1231-1242.

Dörfler, M., Bammer, R., Grill, T. 2017. Inside the spectrogram: Convolutional Neural Networks in audio processing. *2017 International Conference on Sampling Theory and Applications (SampTA)*, Tallinn, Estonia, 152-155. doi: 10.1109/SAMPTA.2017.8024472.

Geib, T., Schmitt, M.and Schuller, B., 2017. Automatic Guitar String Detection by String-Inverse Frequency Estimation. In: Eibl, M. and Gaedke, M. (eds.), *INFORMATIK 2017. Gesellschaft für Informatik,* 127-138. doi: 10.18420/in2017_08

Goto, M., Hashiguchi, H., Nishimura, T. and Oka, R., 2002. RWC Music Database: Popular, Classical and Jazz Music Databases. In: *ISMIR 3rd International Conference on Music Information Retrieval.* Retrieved from https://archives.ismir.net/ismir2002/paper/000049.pdf

Hjerrild, J. M. and Christensen, M. G. 2019. Estimation of Guitar String, Fret and Plucking Position Using Parametric Pitch Estimation. In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 151-155. doi: 10.1109/ICASSP.2019.8683408

Hjerrild, J.M., Willemsen, S. and Christensen, M. G. 2019. Physical Models for Fast Estimation of Guitar String, Fret and Plucking Position. In: *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 155-159. doi: 10.1109/WASPAA.2019.8937157

International Organization for Standardization [ISO] 1975. *Acoustics — Standard tuning frequency* (Standard musical pitch), ISO 16. Retrieved from https://www.iso.org/standard/3601.html

Jadhav,Y., Patel,A., Jhaveri, R. H. and Raut, R. 2022. Transfer Learning for Audio Waveform to Guitar Chord Spectrograms Using the Convolution Neural Network. *Hindawi Mobile Information Systems,* 1-11. https://doi.org/10.1155/2022/8544765

Kim,S., Hayashi, T. and Toda, T. 2022. Note-level Automatic Guitar Transcription Using Attention Mechanism. In: *30th European Signal Processing Conference (EUSIPCO)*, 229-233. doi: 10.23919/EUSIPCO55093.2022.9909659

Li, Z., Liu, F., Yang, W., Peng, S. and Zhou, J., 2022. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems, 33 (12),* 6999-7019.

Maaiveld, T. M. 2021. *Automatic Tablature Estimation with Convolutional Neural Networks - Approaches and Limitations.* A thesis presented for the degree of Master of Science. Department of Computer Science, Faculty of Sciences Vrije Universiteit Amsterdam Netherlands.

Mauch, M. and Dixon, S. 2014. PYIN: A fundamental frequency estimator using probabilistic threshold distributions. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 659-663. doi: 10.1109/ICASSP.2014.6853678

Michelson, J., Stern, R. and Sullivan, T. 2018. Automatic Guitar Tablature Transcription from Audio Using Inharmonicity Regression and Bayesian Classification. *Audio Engineering Society,* Convention Paper 10091.

O'Grady, P. and Rickard, S. 2009. Automatic hexaphonic guitar transcription using non-negative constraints. In: *IET Irish Signals and Systems Conference (ISSC),* 1-6. doi: 10.1049/cp.2009.1699

Osgood, B. 2009. The Fourier transform and its applications. *Lecture Notes for EE 261*.

Osmalskyj, J., Embrechts, J.-J., van Droogenbroeck, M. and Piérard, S. 2012. Neural Networks for Musical Chords Recognition. In: *Journées d'informatique musicale*, Mons, Belgium, 39-46. http://hdl.handle.net/2268/115963

Paleari, M., Huet, B, Schutz, A. and Slock, D. 2008. Audio-visual guitar transcription. In *Jamboree 2008: Workshop By and For KSpace PhD Students,* Paris, France.

Perez-Carrillo, A., Arcos, JL., Wanderley, M. (2016). Estimation of Guitar Fingering and Plucking Controls Based on Multimodal Analysis of Motion, Audio and Musical Score. In: Kronland-Martinet, R., Aramaki, M., and Ystad, S. (eds.), *Music, Mind, and Embodiment. CMMR 2015, Lecture Notes in Computer Science 9617.* Springer, 71-87. https://doi.org/10.1007/978-3-319-46282-0_5

Reboursière, L. and Dupont, S. 2013. EGT: Enriched Guitar Transcription. In: Mancas, M., d' Alessandro, N., Siebert, X., Gosselin, B., Valderrama, C. and Dutoit, T. (eds.), *Intelligent Technologies for Interactive Entertainment. INTETAIN 2013*, *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 124*. Springer, 163-168. https://doi.org/10.1007/978-3-319-03892-6_19

Robinson, E., and Clark, D. 1991. Sampling and the Nyquist frequency. *The Leading Edge*, *10* (3), 51-53.

Ryynanen, M. and Klapuri, A. 2007. Automatic Bass Line Transcription from Streaming Polyphonic Audio. In: *IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, IV-1437-IV-1440. doi: 10.1109/ICASSP.2007.367350

Schörkhuber, C. and Klapuri, A. 2010. Constant-Q transform toolbox for music processing. In: *7th Sound and Music Computing Conference*. Retrieved from https://zenodo.org/record/849741

Stein, M., Abeßer, J., Dittmar, C. and Schuller, G. 2010. Automatic detection of audio effects in guitar and bass recordings. *Audio Engineering Society,* Convention Paper 8013.

Traube, C. and Smith, J. O. 2001. Extracting the fingering and the plucking points on a guitar string from a recording. In: *Proceedings of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (Cat. No.01TH8575)*, 7-10. doi:10.1109/ASPAA.2001.969529

Wiggins, A. and Kim, Y. 2018. *Guitar tablature estimation with a convolutional neural network*. Department of Electrical and Computer Engineering, Drexel University.

Xi, Q., Bittner, R. M., Pauwels, J., Ye, X. and Bello, J. P. 2018. GuitarSet: A Dataset for guitar transcription. Center for Digital Music, Queen Mary University of London, UK.

Zeng, X. and Martinez, T.R., 2000. Distribution-balanced stratified cross-validation for accuracy estimation. *Journal of Experimental & Theoretical Artificial Intelligence*, *12* (1), 1-12.

Open source software:

Abadi, M. *et al.* 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. *Online*: https://www.tensorflow.org

Eyben, F., Wöllmer, M. and Schuller, B. 2010. openSMILE - The Munich Versatile and Fast Open-Source Audio Feature Extractor. *Proc. ACM Multimedia (MM), ACM,* Florence, Italy, 1459-1462.

Harris, C.R., Millman, K.J., van der Walt, S.J. *et al.* 2020. Array programming with NumPy. *Nature* 585, 357–362. doi: 10.1038/s41586-020-2649-2

McFee, B. *et al.* 2023. librosa/librosa: 0.10.0.post2 (0.10.0.post2). Zenodo. https://doi.org/10.5281/zenodo.7746972

Reback, J. *et al.* 2021. Pandas 1.3.5. *Online:* https://zenodo.org/record/5774815

Simola, I. K. 2023. audio-to-tab: 0.1.0, Github repository. *Online:* https://github.com/InkaSimola/audio-to-tab

Additional references:

3D-Spectrogram French horn, 2008. Retrieved from https://upload.wikimedia.org/wikipedia/commons/1/16/3D-Spectrogram_French_horn.png

Aliasing, 2023. Retrieved from https://upload.wikimedia.org/wikipedia/commons/2/20/Aliasing.gif

Electric guitar, 2023. Retrieved from https://en.wikipedia.org/w/index.php?title=Electric_guitar&oldid=1153614107#/media/File:Electric_Guitar_(Superstrat_based_on_ESP_KH_-_vertical)_-_with_hint_lines_and_numbers.png

Equal temperament, 2023. *Online:* https://en.wikipedia.org/w/index.php?title=Equal_temperament&oldid=1156371970

Fundamental frequency, 2012. Retrieved from https://en.wikipedia.org/w/index.php?title=Fundamental_frequency&oldid=1148631124#/media/File:Harmonic_partials_on_strings.svg

Guitar Tablature, 2005. Retrieved from https://commons.wikimedia.org/wiki/File:Guitar_Tabulature.png

Piano frequencies, 2012. Retrieved from https://en.wikipedia.org/wiki/File:Piano_Frequencies.svg

Scientific pitch notation, 2023. Retrieved from https://en.wikipedia.org/w/index.php?title=Scientific_pitch_notation&oldid=1146514356#/media/File:Scientific_pitch_notation_octaves_of_C.png

Ultimate Guitar. *Online*: https://www.ultimate-guitar.com