*i*Journals: International Journal of Software & Hardware Research in Engineering (IJSHRE)
ISSN-2347-4890
Volume 9 Issue 11 November 2021

Autonomous Assistance in Guitar Tutoring for Uninterrupted Self Learning and Practice

Avni Prateek Mehta
School : RN Podar School, Mumbai, Maharashtra, India,
avnimehta.p@gmail.com

Reetu Jain
Chief Mentor ON MY OWN TECHNOLOGY
Mumbai, Maharashtra, reetu.jain@onmyowntechnology.com

Abstract – COVID Pandemic necessitated "assisted learning" and gave me an idea to experiment if my Guitar Tutor - a software, could guide me if I am playing the right guitar notes. Practicing and playing the right music without waiting for the availability of the coach is a huge boost in taking the interest forward. Guitar Tutor compares the real tune with the notes played and exactly points out the notes that are played differently. Three main parameters to determine and compare frequency are - Thickness of the string, frequency, and hand-held placement. I had to create our own database of chords. In our database, I gathered audio files (recorded in the WAV format, sampled at fs = 44100 Hz, and quantized with 16 bits), and the corresponding precomputed PCP vectors. The PCP vectors are computed on windows comprising each 16384 samples, which correspond to 0,37 seconds. The window size was chosen experimentally. I noticed that windows containing only 4096 samples produce correct results, however, the best results for our application. My GuitarTutor software uses Feed Forward Neural Network to derive a model for extraction and classification of notes. When the learner plays the note to my GuitarTutor, it detects an audio event, filters the noise using LibROSA and PyAudioAnalysis programming modules. Now the NN model analyses the note to extract specific patterns of frequency which then evaluates, predicts the note played and compares to the base-dataset with an accuracy of 85%.

***Keywords* – PyAudio Analysis, Neural Networks, Librosa, Machine Learning.**

I. INTRODUCTION

A hobby like playing musical instruments is one hobby that teaches patience and perseverance. One of them is a musical guitar instrument. With the many variations of the chord in playing the guitar, to be proficient or just to sing harmonious melodies while being clear needs perseverance and patience. Therefore, teachers are needed for teaching materials that can facilitate the learning of guitar musical instruments. Covid has been a very rough ride for the music industry as the traditional method of teaching music, face to face, has been replaced with online teaching of music which is very tough. In fact it will be a while for this industry to bounce back. This transition to online digital delivery of music lessons has not always been a smooth ride for many. [1] Teaching music online requires a completely different set of skills, compared to being in the same room with the learners. Therefore, it is important to carefully assess the teaching approach and methodology of independent music teachers if you are planning to learn music remotely from an individual. Many might offer trial lessons for learners to experience the process. However, there is more to learning music online and remotely than just attending Zoom or Skype lessons. Institutes like The True School of Music have invested heavily in online remote learning infrastructure and teacher enablement to ensure a truly immersive and social learning experience. One of the aspects that suffered the most during the transition to online music education was the ability to experience ensemble playing, and the faculty at True School have come up with innovative approaches to collaborative music playing in a remote learning environment.

So, this project tries to help the music industry develop self-analysis of guitar learning by using machine learning. The first compulsory step of a retrieval system which is able to process music is the characterization of music. Several techniques are available but the probably best known to musicians is that of chords [1]. A chord, in its most basic music theory definition, is three or more notes played at one time. However, anyone who has messed around strumming random guitar notes knows that most combinations of three or more notes are not very pleasant chords. The most basic form of a chord is a Major chord based on the

Major scale, which is the typical, upbeat musical scale. Every Major chord is played with three notes that correspond to the first, third, and fifth notes of the Major scale of the note it is based upon. A C scale, for example, goes C, D, E, F, G, A, and B.

Minor chords work similarly but are played on the minor scale instead of the Major scale. For example, the A minor scale is A, B, C, D, E, F, G. Note that the A minor scale has the same notes as C major because A is the relative minor of C. The only difference between these two scales is the order in which the notes are played.

Chords on the guitar work the same way musically as on a piano or other instrument but are a little more confusing because of the way notes are laid out on the neck of the guitar. This is the reason guitar through online teaching has been difficult for new beginners, so using machine learning and chords as the main parameters I have come up with a system that will detect the frequency of the chord and help to learn the basics of the chord perfectly. This means, which are notes that when played together have a proper balance to sound pleasing for the human ear.

II. LITERATURE WORK

I have always been into music. It was one thing that made my mood better, covid life made it very hard for learning and pursuing my passion of music. I have seen various teachers and schools for online teaching, it was always difficult to teach online. An observation that I made was the way I place fingers on the string makes a major difference. As the teachers viewed through the camera it was always difficult for the teachers or even for me to get the finger placement correct, and the problem was compounded by the internet and camera quality, hardware components like mic, speaker etc are also other the reasons that affect online teaching and learning process. To overcome these issues, I have noticed that machine learning can be used and it will help to solve these issues, by taking the sound of the guitar and training the audio, so that it corrects me to play perfectly. This system is a start, but it will solve many problems and help to learn music with fun.

III. METHODOLOGY

The most used descriptor for chord identification has been the Pitch Class Profile (PCP). [1]A chord is composed of a set of tones regardless of their heights, and therefore a PCP vector seems to be an ideal feature to represent a musical chord. There are some variations to obtain a 12-bin PCP, but its computation usually follows the same steps. First the algorithm transforms a fragment of the input sound to a discrete Fourier transform (DFT) spectrum X(.). Then the algorithm derives the PCP from X(.)

Let P CP∗ (p) be a vector defined for p = 0, 1, . . . , 11 as

Equation 1

$$PCP^*(p) = \sum_{l} ||X(l)||^2 \, \delta(M(l), p)$$

where δ (·, ·) denotes Kronecker's delta. M(l) is defined as

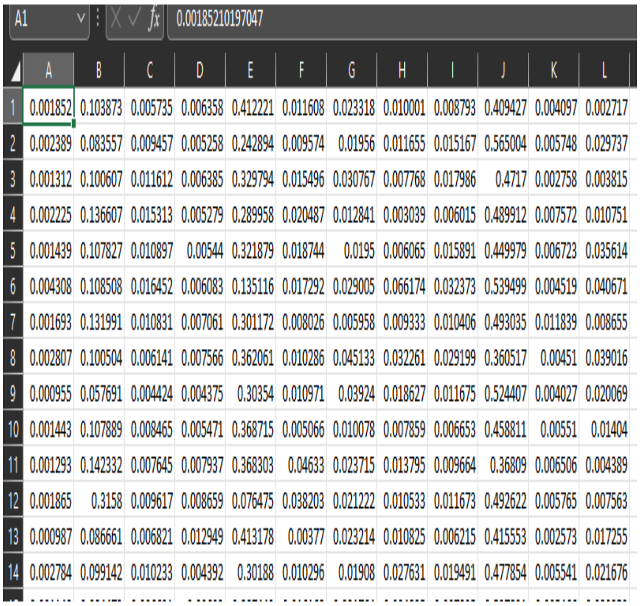
$$M(l) = \begin{cases} -1 & l = 0 \\ round(12 \log_2((f_s \cdot \frac{l}{N})/f_{ref})) \bmod 12 & l = 1, \ldots, \frac{N}{2} - 1 \end{cases}$$

where is the reference frequency falling into P CP∗ (0), N the number of bins in the DFT of the input signal, and fs is the sampling frequency. For example, for a standard scale starting with a C, the reference frequency is 130.80 Hz.

Equation 2

$$PCP(p) = \frac{PCP^*(p)}{\sum_{j=0}^{11} PCP^*(j)}$$

Part of the difficulties in machine learning techniques originate from the elaboration of a database of samples, also called dataset, and chord classification is no exception. The primary requirement for a chord recognizer using machine learning techniques is that the dataset contains enough data to build a model. For that reason, I had to create our own database of chords. In our database, I gathered audio files (recorded in the WAV format, sampled at fs = 44100 Hz, and quantized with 16 bits), and the corresponding precomputed PCP vectors. The PCP vectors are computed on windows comprising each 16384 samples, which correspond to 0,37 seconds. The window size was chosen experimentally. I noticed that windows containing only 4096 samples produce correct results, however, best results for our application are achieved using a bigger window size. Since our final goal is not to recognize all the existing chords, but to develop a music recognition system, I can limit chords to the most frequent ones. Therefore, I chose a subset of 10 chords: A, Am, Bm, C, D, Dm, E, Em, F, G. In our database, these chords are represented by an identifier ranging respectively from 0 to 9.



A1 | 0.00185210197047

	A	B	C	D	E	F	G	H	I	J	K	L
1	0.001852	0.103873	0.005735	0.006358	0.412221	0.011608	0.023318	0.010001	0.008793	0.409427	0.004097	0.002717
2	0.002389	0.083557	0.009457	0.005258	0.242894	0.009574	0.01956	0.011655	0.015167	0.565004	0.005748	0.029737
3	0.001312	0.100607	0.011612	0.006385	0.329794	0.015496	0.030767	0.007768	0.017986	0.4717	0.002758	0.003815
4	0.002225	0.136607	0.015313	0.005279	0.289958	0.020487	0.012841	0.003039	0.006015	0.489912	0.007572	0.010751
5	0.001439	0.107827	0.010897	0.00544	0.321879	0.018744	0.0195	0.006065	0.015891	0.449979	0.006723	0.035614
6	0.004308	0.108508	0.016452	0.006083	0.135116	0.017292	0.029005	0.066174	0.032373	0.539499	0.004519	0.040671
7	0.001693	0.131991	0.010831	0.007061	0.301172	0.008026	0.005958	0.009333	0.010406	0.493035	0.011839	0.008655
8	0.002807	0.100504	0.006141	0.007566	0.362061	0.010286	0.045133	0.032261	0.029199	0.360517	0.00451	0.039016
9	0.000955	0.057691	0.004424	0.004375	0.30354	0.010971	0.03924	0.018627	0.011675	0.524407	0.004027	0.020069
10	0.001443	0.107889	0.008465	0.005471	0.368715	0.005066	0.010078	0.007859	0.006653	0.458811	0.00551	0.01404
11	0.001293	0.142332	0.007645	0.007937	0.368303	0.04633	0.023715	0.013795	0.009664	0.36809	0.006506	0.004389
12	0.001865	0.3158	0.009617	0.008659	0.076475	0.038203	0.021222	0.010533	0.011673	0.492622	0.005765	0.007563
13	0.000987	0.086661	0.006821	0.012949	0.413178	0.00377	0.023214	0.010825	0.006215	0.415553	0.002573	0.017255
14	0.002784	0.099142	0.010233	0.004392	0.30188	0.010296	0.01908	0.027631	0.019491	0.477854	0.005541	0.021676

Figure 1: "The dataset of A chord"

I have used Librosa and audio analysis to understand the frequency and amplitude of each chord. Using this algorithm, I have plotted down with respect to frequency and time taken in seconds as you can see in the below graph. [1]

```
import librosa
import IPython.display as ipd
✓ 0.7s
```

```
x, sr = librosa.load(r'C:\Users\Radhakrishna\Documents\Guit
print(type(x))
print('x length: {}'.format(len(x)))
print(type(sr))
print('sr = {}'.format(sr))
✓ 0.3s
<class 'numpy.ndarray'>
x length: 39130
<class 'int'>
sr = 22050
```

```
print('Sound clip is {}.'.format(len(x)/sr))
✓ 0.1s
Sound clip is 1.7746031746031745.
```

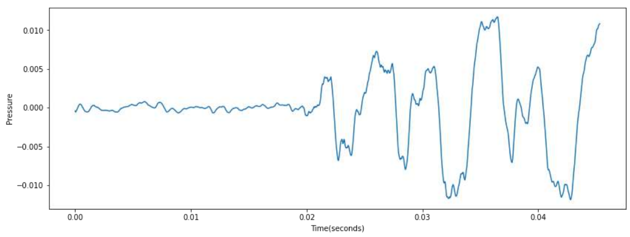


Figure 2:"The A chord frequency"

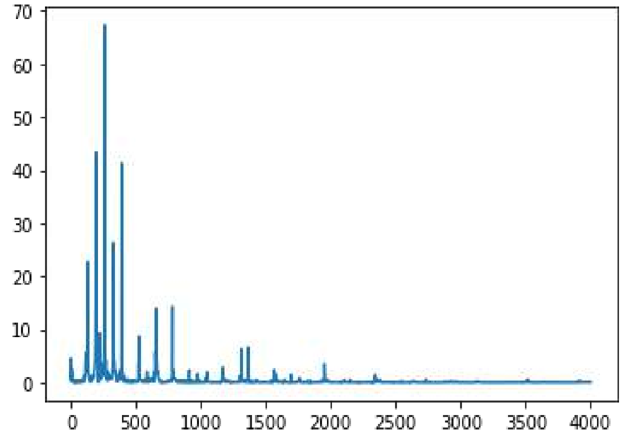


Figure 3: "FFT of A chord"

Convert the above frequency to FFT (A fast Fourier transform (FFT), which is an algorithm that computes the discrete Fourier transform (DFT) of a sequence, or its inverse (IDFT). Fourier analysis converts a signal from its original domain (often time or space) to a representation in the frequency domain and vice versa.

C String

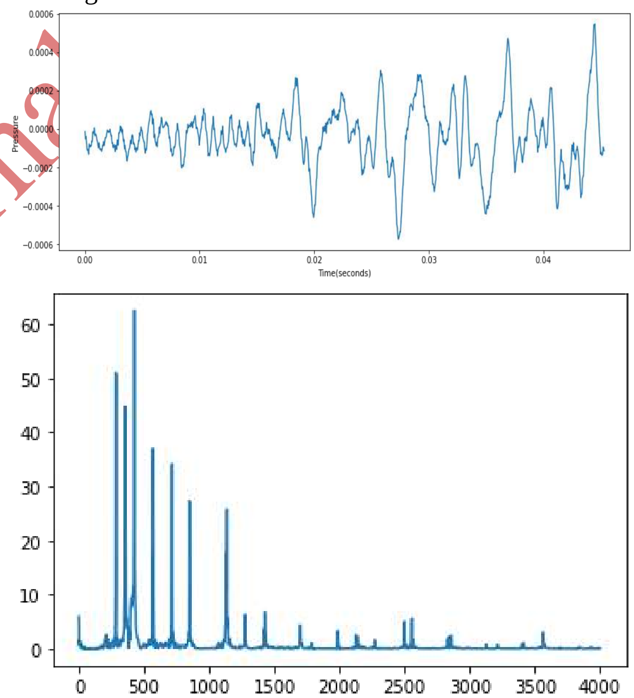
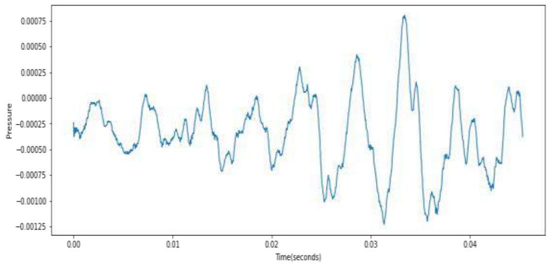


Figure 4: "The C Chord Frequency and FFT"

Each string has a separate frequency as you see in this image and observe the wavelength that is needed for understanding the string to capture them.

G string



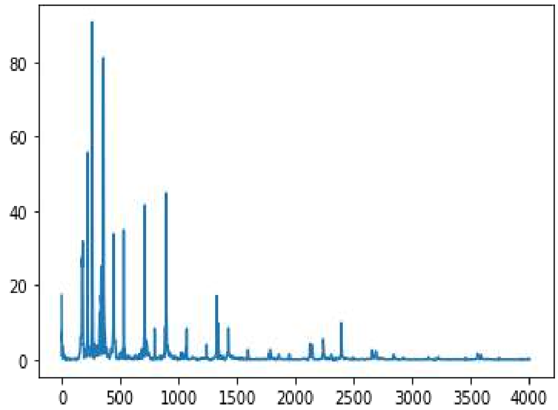
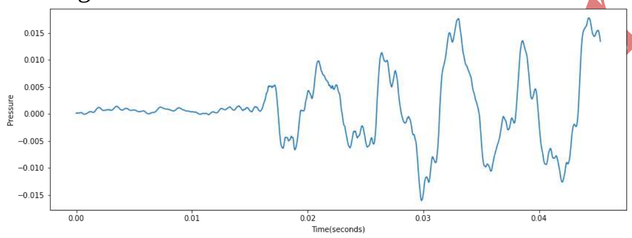


Figure 5:"The G Chord Frequency and FFT"

E String



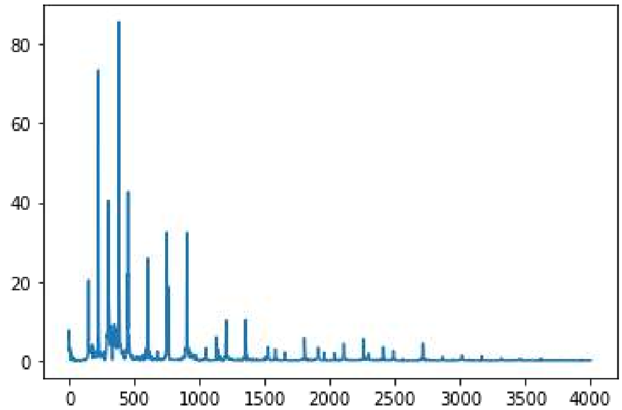
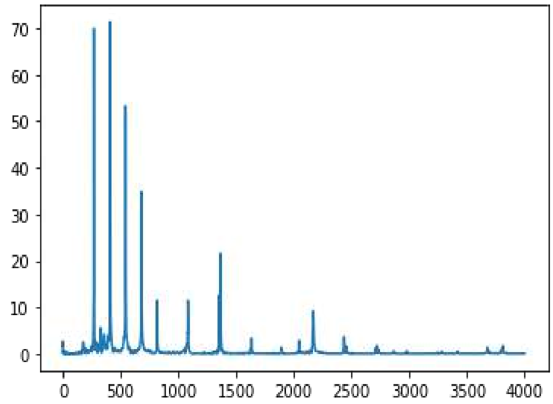


Figure 6:"The E Chord Frequency and FFT"

D string



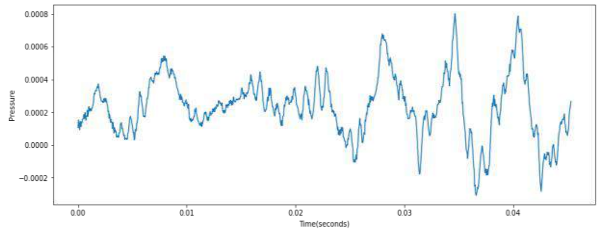
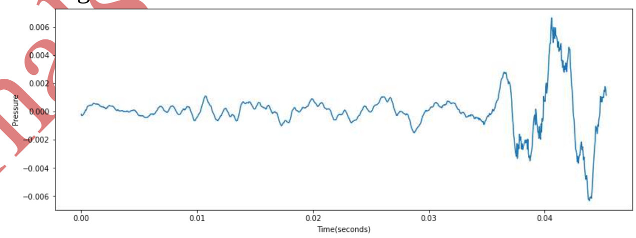


Figure 7:"The D Chord Frequency and FFT"

F string



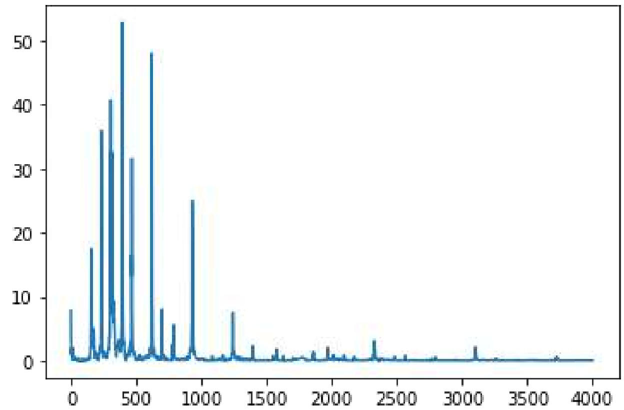


Figure 8: "The F Chord Frequency and FFT"

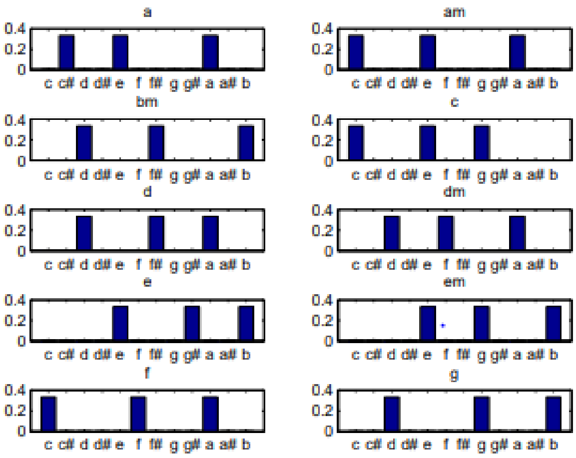


Figure 9: "PCP of each chord in the guitar"

This section first introduces the learning method chosen for the design of our system.

1. I want to check if a naive application of the chord definition suffices.
2. How do I build the training set? Should I use noise-free samples, noisy samples, or both types of samples to be included during training?
3. I want to evaluate the performance of our learning algorithm with our database.
4. To check the sound that is recorded by laptop and check with a machine learning chord.

Learning algorithm

Most techniques proposed in the literature for chord recognition do not use machine learning methods that use a pattern matching technique and heuristics to recognize chords. Although the techniques developed are efficient, they are complex since our final goal is not to develop the technique to recognize chords and analysis. The chosen algorithm is a feed-forward neural network using a classical gradient descent algorithm with a negative log-likelihood [1] as cost function.

Table 1: Neural Networks parameters

Parameters	Values
Number of hidden layers	1
Number of neurons in the hidden layer	35
Learning rate	0.001
Momentum	0.25
Weight decay	0.0

The neural network architecture is the following - There are 12 input attributes, which correspond to the 12 semi-tones of the PCP vector representing the chord. The neural network outputs a vector of 10 values, corresponding to the output neurons, each one being the probability of the detected chord to be issued from the corresponding chord. The final settings of the neural network are optimized by a 10-fold cross-validation on the learning database.

Table 2: Dataset of Noise, Noise-free and Mixed

TS / LS	Noise-free	Noisy	Mixed
Noise-free	4.0 %	5.0 %	4.0 %
Noisy	11.7 %	6.0 %	7.3 %

In this first experiment, I have created a synthetic and ideal sample of PCP for each chord manually. Then, using the Bhattacharyya distance [4], I have applied a nearest neighbours (1-NN) algorithm on our second subset. The classification error rates obtained are the following: 8 % for guitar, 20 % for piano, 19 % for violin, and 32 % for accordion. These results are clearly unsatisfactory. The conclusion is straightforward: learning based on real samples is necessary to reach the required performance level.

```
import numpy as np

from preprocessing.pitch_class_profiling import LongFileProfiler, PitchClassProfiler
from neural_network.train import Trainer
from util import config

class Spliter():
    def __init__(self, song_file, out_file="spliter_result.txt"):
        self.song_file = song_file
        self.out_file=out_file

    def split_song(self):
        trainer = Trainer()
        trainer.load()

        longFileProfiler = LongFileProfiler(self.song_file)
        profiles = longFileProfiler.get_profile()

        chords = []

        for profile in profiles:
            X = np.array( [profile] )
            prediction = trainer.model().predict(X)
            chord_index = np.argmax(prediction)

            chords.append( config()["pitches"][chord_index] )
        return chords

    def save_split(self):
        chords = self.split_song()
        chords_string = " ".join(chords)
        with open(self.out_file, "w") as f:
```

Figure 10: "Neural Networks"

Determining the optimal learning set explained, I created a database with noise-free chords and noisy chords. To justify that choice, i performed six tests to determine the best of the three following configurations:

- learning set with noise-free chord samples only,

- learning set with noisy chord samples only,
- and a learning set with mixed chord samples.

To perform the test, I split the database in different sets. First, the original database of 2000 samples were split into two sub-databases of 1000 elements. The first one only contains noise-free chords and the second one contains noisy chords. Then, I created three learning sets containing 70% of each sub-database. The first set contains 700 noise-free chords, the second 700 noisy chords, and the last one 350 noise-free chords and 350 noisy samples, taken randomly. For the tests, I created two Test Sets (TS) with 30% of each sub-database. One test set contains noise-free chords and the second one contains noisy chords only. It is worth noticing that the chords used for the TS are not included in the LS.

First, I trained the model with a noise-free learning set and tested it with the noise-free and noisy test sets (first column of the table). Next, I trained the model with a noisy learning set and tested it with a noise-free and noisy test sets (second column). Finally, I performed the same tests with a learning set containing both noise-free and noisy chord samples (third column). The chords are recorded with different guitars distributed equally in each set.

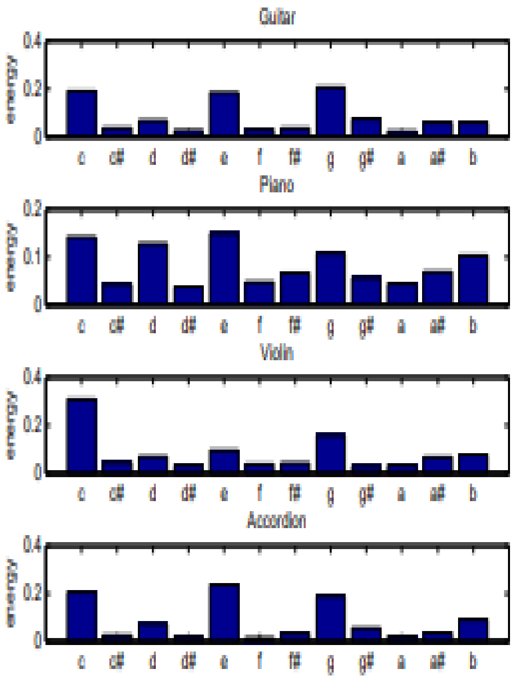


Figure 11: "PCP of various other music instrument "

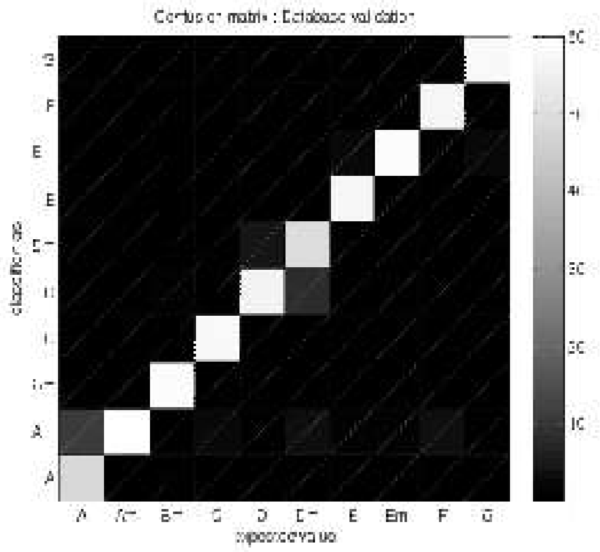


Figure 12: "Confusion Metrix"

I conclude that building the model using a noise-free learning set produces the highest error rate for the noisy test set. Moreover, the optimal learning set (noisy or mixed) depends on the conditions under which the model must be used. Unfortunately, I was not able to guess it in advance. However, I consider that the mixed learning set produces models that are less dependent on the noise in the database than with a noisy learning set, and therefore, I believe it is preferable to use a mixed learning set.

Validating the database from our previous observations, I have decided to train our final model with both noise-free and noisy chord samples. The confusion matrix for the final network trained with a learning set of 1400 chords, that is 140 chords per class and a test set of 600 chords, that is 60 per class. The main database was split in two parts and thus, the result is slightly biased due to the size reduction of the database. Despite that bias, I can observe that the prediction of each class is quite good. Indeed, the rate of correct classification for each class is almost identical. Moreover, classification errors are not concentrated in a unique position.

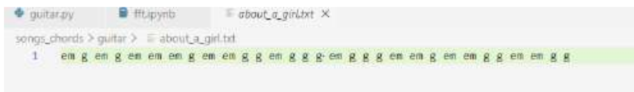


Figure 13:"Song played from the guitar"

After this I made a GUI application using python programming that displays the various scales of chord with the major and minor and then records the sound using the laptop mic and saves the file. Which then passes through the machine to see if the chord played matches perfectly with the frequency.

This way, we will be able to learn and even differentiate sounds.

```
import tkinter as tk
from tkinter import ttk
import random
import time

win = tk.Tk()
win.title("Chords to Play")


'''
Chord Theater Presentation Box
'''
# Container to hold Chord Theater
labelsFrame = ttk.LabelFrame(win, text='Chord Theater')
labelsFrame.grid(column=1, row=0, padx=40, pady=40)

# Place labels into the container element
chordPresenter = ttk.Label(labelsFrame, text=" ")
chordPresenter.grid(column=0, row=0, padx=80, pady=50)


'''chord radiobuttons'''
#Container for radiobuttons
chordtype_frame = tk.Frame(win)
chordtype_frame.grid(column=0, row=0)

#RadioControls:
def RadioControls():
    if radVar.get() ==1:
        chordPresenter.configure(text="Let's Play \n Major Chords!")
    elif radVar.get()==2:
        chordPresenter.configure(text="Let's Play \n Minor Chords!")
```

Figure 14: "GUI Algorithm"

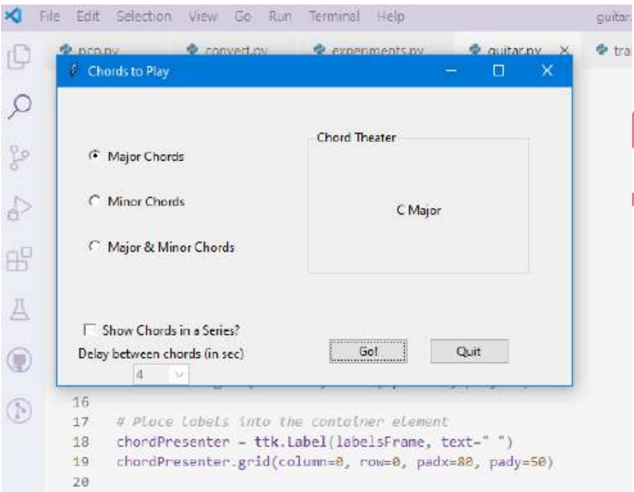


Figure 15: "Learning Platform"

IV. CONCLUSION

This paper was a start for online music teaching and with help of machine learning techniques, using a feed-forward neural network rigorization of raw audio and all the major and minor chord, we will be able to use them and predict the chord of the guitar. This way it will help learners learn, grow, and have fun with music.

ACKNOWLEDGMENT

This research was supported by [ON MY OWN TECHNOLOGY]. Special thanks to Sir Radhakrishna Madas from OMOTEC who provided insight and expertise that greatly assisted the research,

References

[1] "Play Guitar," [Online]. Available: https://playguitarnotes.com/beginners/what-are-guitar-chords/.

[2] "BWeducation," [Online]. Available: http://bweducation.businessworld.in/article/How-COVID-19-Changed-Music-Education-In-India-/18-03-2021-384266/.

[3] T. Fujishima., " Realtime chord recognition of musical sound: a system using common lisp music.," no. In Proceedings ot the International Computer Music Conference (ICMC)., p. 464–467, 1999.

[4] R. T. a. J. F. T. Hastie, " The elements of statistical learning: data mining, inference, and prediction.," no. Springer Series in Statistics., 2009.

[5] F. K. a. M. C. M. Müller, "Audio matching via chroma-based statistical features.," no. In Proceedings of the International Conference on Music Information Retrieval (ISMIR),, p. 288–295, 2005.

[6] J.-J. A. a. F. Pachet., "Music similarity measures: What's the use?," no. In Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR), pp. 157-163, 2002.

AUTHOR PROFILE

Avni Prateek Mehta
School : RN Podar School, Mumbai, Maharashtra, India, avnimehta.p@gmail.com

Reetu Jain
Chief Mentor ON MY OWN TECHNOLOGY
Mumbai, Maharashtra,
reetu.jain@onmyowntechnology.com

