JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125 VOL 7, ISSUE 15, 2020

Automatic Note and Chord Recognition for Harmonium Music: A Deep Learning Approach

Surekha B. Puri[1], Dr.S. P. Mahajan[2]

[1]Department of Electronics and Telecommunication Engineering, College of Engineering Pune (COEP), Pune, Maharashtra, India

[2]Department of Electronics and Telecommunication Engineering, College of Engineering Pune (COEP), Pune, Maharashtra, India

Received: 01 May 2020 Revised: 23 June 2020 Accepted: 04 July 2020

ABSTRACT: Music has played a significant role in the history of mankind. Understanding the theoretical fundamentals of music makes learning musical instruments easier. While many researchers have tried various techniques for music note recognition or acoustic chord recognition for the piano and other musical instruments, no method has been developed for the harmonium. A Convolutional way of Recurrent Neural Network known as(CRNN) based upon automated harmonium musical note identification approach is presented in this research by considering different audio features like pitch onset and offset times, signal energy and so on and their combinations. The audio samples used as input to the proposed system have been collected from a professional music player. These samples have been used to train the prediction model. In this approach, 900 harmonium audio samples comprising of various chord combinations are trained and tested by Convolutional Neural Network referred to as(CNN) and Recurrent Neural Network is also known as (RNN) with different sample combinations. CNN and RNN have a good rate of accuracy, but CRNN is the most accurate. The proposed system attains 94% accuracy. The resultant prediction of chords is then passed to the Lilypond library of Python to generate a music sheet that can be directly used by professional or novice musicians for composing music. The proposed method attains promising works on the self-created schema.

KEYWORDS: - Music Information Retrieval, Audio Analysis, music analysis, Recurrent Neural Networks, Convolutional Neural network

I. I.INTRODUCTION

Monophonic music is when only one key is played at any point in time. In the case of polyphonic music, three or four notes are simultaneously played and this is called a chord. Monophonic music transcription is a relatively easy task. When different combinations of chords are played it is a very complex task to transcribe the chord. The paper mainly focuses on polyphonic chord transcription.

When musicians learn a new musical instrument, there needs to be an understanding of the chord progression. Unlike the piano, there is very little information on chords when learning a harmonium. The harmonium is an integral part of Indian Classical music and therefore, there exists a need for research on this topic.

Harmonium, also known as a Reed Organ, is a free-reed keyboard instrument. It has keys similar to the electric synthesizer. The main difference is that a performer can play the keys either with the right or the left hand, and use the other hand for pumping the harmonium bellow, the image of harmonium is as shown in figure 1.

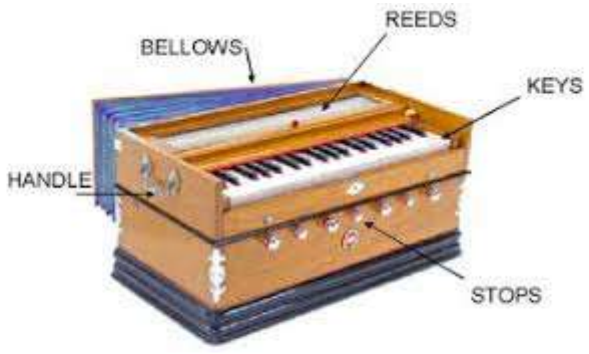


Fig. 1. Harmonium Instrument

Classical Indian music is usually represented by 7 elementary notes (Sa, Re, Ga, Ma, Pa, Dha and Ni), which then results in the formation of a 12-note scale with five interspersed half-notes. As per previous research, it is definitely clear that any key can become a 'Sa' or the starting note; but in this system, we have assumed the first black key to be the Sa (S). Generally, there are various chords in typical Western music, but in harmonium lessons, we only use the major and minor chords. Basic harmonium chords are also referred to as desi chords. The komal or flat and teevra or sharp harmonium notes with their respective numbers are mentioned below and shown in Table I.

Table I. Harmonium Komal and Teevra notes

C	Key: 1
C#	Key: 2
D	Key: 3
D#	Key: 4
E	Key: 5
F	Key: 6
F#	Key: 7
G	Key: 8
G#	Key: 9
A	Key: 10
A#	Key: 11
B	Key: 12

In the latest Western music, its base frequency of the scale is fixed. Whereas in Indian classical music it is not fixed like the 12-note scale in Western music, and also, the inter-tonal gaps may be different. In order to understand how the harmonium functions, it is essential to know how the notes correspond to each other. Comparing the notes of a keyboard with the notes of the alphabets is probably the best way to describe them. The first 7 notes are A, B, C, D, E, F, and G. If we try to differentiate between the notes, each note differs from the other notes in terms of sound. All of the seven notes of a keyboard are repeated over and over again. These notes often sound the same but the pitch of each of the notes differs. For example, if you are playing a C note and you eventually move towards the right until the next occurrence of a C note, you will realize that if you play them together, both the notes sound exactly the same, but one has a higher pitch and another one has a lower pitch comparatively. It is a widely known fact that Indian music is more melody-based music whereas Western music is harmony based. But this doesn't mean that Western music does not comprise of melody at all. In fact, in Western music, various instruments and voices have a harmonizing effect for any given melody and harmony plays a vital role for that piece of music. However, in Indian music, the melody has an upper hand. There is a strong belief that harmony doesn't really play a role in Indian music. But actually, harmony is of equal importance in any Indian classical music performance. It is just that the role of harmony in Indian classical music is different from Western music formats. In either Indian classical music or Western music, notes played singly are monophonic and multiple notes played simultaneously are polyphonic.

The term Polyphony comes from the Greek origin which means "many sounds". In musical terms, Polyphony means a combination of multiple tones or melodies played at a time. Hence, even an interval which is made up of two chords or tones played at the same time is primitively polyphonic [3]. Polyphony is basically related to counterpoints i.e. a combination of more than one distinct melody at once, and musically means the playing of multiple tones/melodies at once.

A standard harmonium usually has 36 keys in total. Here there are 3 sets of 12 keys each. The set of 12 notes is called an octave. The reason that there are just 12 notes and not any other number is to keep it simple and clean. Detection of the monophonic chords played with Western musical instruments such as the guitar, keyboard and mouth organ has been extensively researched by existing systems. The harmonium rarely has been used as a research instrument so far and so it has always been a challenge for researchers to work on harmonium chord recognition.

The proposed system makes use of the Convolutional Recurrent Neural Network (CRNN) approach for determining polyphonic chords played on the harmonium. The rest of the paper and our research work are depicted in this work as follows:

- Section II comprises of a comprehensive survey of the existing research done by other authors in the Music Information Retrieval domain.
- In Section III, an overview and detailed work of the proposed technique for the polyphonic chord recognition have been explained.
- Section IV gives the result of the analysis of the proposed system.
- Section V gives evaluation metrics of the proposed system on the basis of evaluation parameters such as precision, recall, F1 score and accuracy.
- The remaining sections give details about the future scope and conclusions drawn from the obtained results of the proposed system.

II. LITERATURE SURVEY

Automatic Music Transcription can be correlated to the Automatic Speech Recognition (ASR) system. However, Automatic Music Transcription has not been as extensively worked upon and researched as the ASR. With the advent of digitization, online audio streaming got a huge boost in the last decade. This created the need for music information retrieval, for music classification and recommendation [4]. Conventionally music transcription was done with the help of a feature named Short Term Fourier Transform (STFT). Robert Dobre propose a newer mechanism for music transcription using Constant Q Transform (CQT) [12]. Paris Smaragdis researched polyphonic music transcription based upon another factorization mechanism which is known as Non-Negative Matrix Factorization (NMF) [6]. This separates out the polyphonic chords or music into different components [4]. Cian O'Brien et al. consider that music transcription in itself is a very low-rank matrix [8].

2.1 The End-to-End Neural Network for Polyphonic Piano Music Transcription (2016) [1] Authors: Siddharth Sigtia and Emmanouil Benetos. :In this work, the proposed system modeling is similar to speech recognition systems which apply acoustic and music language modeling. The proportion or percentage of the existence of the pitch is determined using the neural network in an acoustic model. While on the other hand, Music Language Models (MLMs) are RNN models that correlate the pitch frequency over time.

2.2 A Hybrid continual Neural Network for Music Transcription (2015) [2] Authors: Siddhartha Sigtia, Emmanouil Benetos, and Simon Dixon:The authors look into the challenge of absorbing higher-level representative score-like facts into Automatic Music Transcription (AMT) systems to improve their performance. In this paper, instead of using just one acoustic model, two models are combined to form a hybrid model. RNN is good for predicting note onsets. RNN is cascaded with DNN to increase accuracy. It is observed that DNN+RNN gives more precise results over DNN or RNN alone. The audio is below-sampled from 44.1 kHz to 16 kHz. The preprocessing of the spectrogram is done by reducing the mean and dividing by the standard deviation of each frequency bin.

2.3 Polyphonic piano note transcription with recurrent neural networks (2012) [3] Authors: Sebastian Böck and Markus Schedl.:In this, the authors propose a novel approach for polyphonic piano note onset transcription. The basis of this is an RNN to detect the onsets and the pitches of the notes from spectral features at the same time. LSTM units are applied in a bidirectional neural network to model the context of the notes. In this system, Bi-LSTM networks are used. Short Term Fourier Transform (STFT) is used for preprocessing the audio data. Two different frame expanse of 2048 and 8192 samples are used to give a good resolution in time as well as frequency domain. The bi-directional network gives a wider temporal context. The short STFT window is time-sensitive while the larger window is frequency sensitive.

2.4 Frames and Onsets: Dual-objective piano transcription (2017) [5] Authors: Curtis Hawthorne.

The authors present the latest technique in polyphonic piano music transcription by using DNN and RNN which are trained to forecast onsets and frames jointly. The model proposed by the authors predicts pitch onset events and then makes use of those predictions to condition frame-wise pitch predictions. In this system, the DNN and the RNN are trained in cascade to predict frames and onsets. Onset events are predicted and then those onsets are used to predict pitches. Onsets and offsets are improved together and not separately. The authors have also calculated the velocity of key depressing, which also gives the intensity of sound.

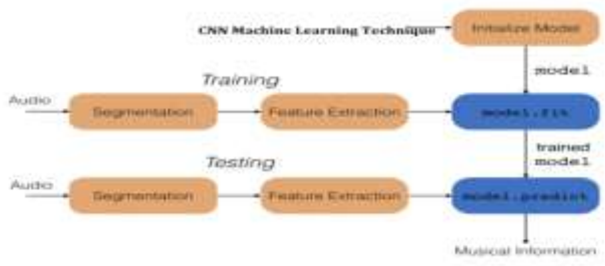
2.5 A polyphonic music sequence modeling is based on the study on LSTM networks. (2017) [6] Authors: Adrien Ycart, Emmanouil Benetos.

This paper discusses the influence of various LSTM parameters on the performance of music transcription. It shows that the more the sampling frequency, the better the prediction. However, the higher sampling frequency uses up more memory and more processing power, thereby increasing the processing time. Generally, 50 epochs sufficiently train the network. The learning rate has a profound influence on the convergence speed. A learning rate of 0.001 and 256 hidden layers were found to be optimum parameters.

III. PROPOSED CHORD RECOGNITION METHODOLOGY

Musical chord recognition has always been an area of interest for the researchers. Monophonic music transcription has been the topic of research for a long period of time. But the time has come to go further and make a transcription of polyphonic music chord recognition possible. Only speech recognition or only speech analysis does not completely or accurately provide transcription of chords. Techniques that have been researched earlier include end-to-end neural network mechanisms for music transcription. But the plain neural networks lack the desired as well as the requisite efficiency and take longer to transcript the input audio file. Figure 2 depicts the model of the proposed system.

Convolutional Recurrent Neural Network (CRNN), is deep learning, machine learning technique for machines to comprehend the features of harmonium chord with foresight and remember the features suggest whether the name of the new chord fed to the classifier. The model is trained on recordings for some specific chords. After some initial testing, it was discovered that using an initial base learning rate of a few which worked well in fitting the training data - it provided a stable increment in accuracy and seemed to successfully converge. Once the improvements in the loss stagnated, the process is terminated manually and decremented the learning rate so as to try and increase the optimization of the loss function. Training took approximately an hour for 10 epochs for



each model.

Fig. 2. Proposed System Building Blocks

The model builds with a training dataset up to 1/5 epoch and each epoch contains multiple Keras level which calculate the loss and accuracy and iterate till model fits correctly.

The model builds with a training dataset up to 1/7 epoch and each epoch contains multiple keras levels and multiple steps each epoch which calculates the loss and accuracy and iterates till model fits correctly. The model builds with training dataset up to 1/10 epoch and each epoch contains multiple keras levels and contains 100 step each epoch and few more validations which calculate the loss and accuracy and iterate till model fits correctly.

Sequential API — The simplest form of API where you first call model = Sequential ( ) and keep adding on layers, e.g. model.add(Dense(...)) .

The model comprises of 2 Layer neural network having "RELU" as it first layer activation function and sigmoid as the next layer activation function. The neural network layers accept the numeric representation of the input chords and pass the correlated values to the next layer, and the next layer analyses the correlated values to predict the category of the trained samples.

Mathematical representation for model is as follows:

model = Sequential()

Add (Dense (30, input_dim=12, activation='relu'))

model.add(Dense (10, activation='sigmoid'))

model.compile (loss=self.loss_function, optimizer='adam', metrics =[metrics. categorical_ accuracy, metrics.top_k_categorical_accuracy])

The optimizer function used is 'adam' which ensures maximum optimization of the neural network.

model. Fit(Y,Z, epochs=100, batch size=10, verbose=0)

Epoch used is 100 to make sure that the model reaches the highest accuracy for predicting the input chord.

Recurrent Neural Networks (RNN) has been widely used in speech recognition & handwriting recognition due to its capability of dealing with sequential data. However, only one past unit is indulged in the frame of RNN, vanishing gradients during backpropagation becomes the major problem. One possible result is a special kind of RNNs, called Long Short-Time Memory Networks (LSTM), whose structure involving various past units enables it to grasp long-term dependencies.

For hidden layers, we use tanh for *M:*

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \qquad (2)$$

For the output layer, we use sigmoid for f :

$$f(x) = \frac{1}{1} + e - x \qquad (3)$$

LSTM being the extension of this, which uses the memory cell, in which 4 neural networks are indulged. The update process is done with these 4 neural networks. The memory cells of LSTM are as shown in Figure 3.below.

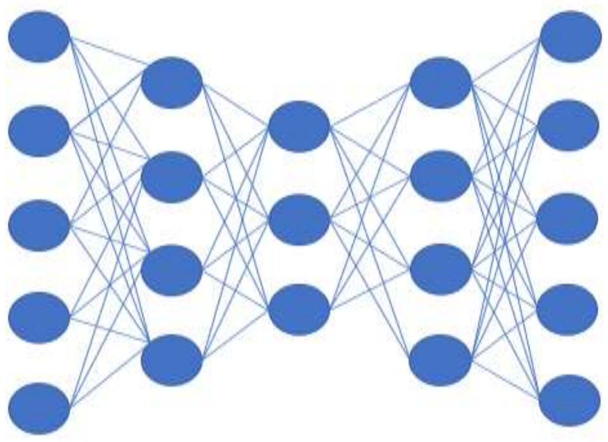


Fig.3. Simple Neural Network architecture

The proposed system makes use of Advanced Deep Neural Network which makes use of a convolutional mechanism for training audio samples and thereby uses the trained dataset for further determining the polyphonic chords from the input audio files.

The proposed methodology is the combination of various feature extractions and analyses for determining the chords of the harmonium that are played. The proposed model makes the use of an artificial intelligence technique called Convolutional Neural Network to train the samples of the harmonium dataset obtained from professional artists and then these samples are tested to obtain the desired results.

The proposed system is implemented in two phases:

a. Audio Feature Training Phase.

b. Audio Chord Detection Phase.

IV. MATHEMATICAL REPRESENTATION OF THE SYSTEM

The proposed system can be mathematically as a quintuple or DFA i.e. Deterministic Finite Automata, as the proposed system has finite input set, finite output and Finite states in which system transits.

It can be represented as follows:

$$S = \{I, O, P, q, F\}$$

Where,

I-> represents the input audio file given for chord recognition,

O-> represents Output chord recognition obtained after the proposed system executes over the provided input,

P-> represents the process that id performed over the input or intermediate input obtained while the proposed system executes,

q-> represents the various phases or states in which the system is while the entire system is executed,

F-> represents the final state where the output is obtained.

The system executes successfully if the provided input 'I' is from a valid instrument i.e. Harmonium. If the provided input is in the required format, the system processes the input and generates output for the music sheet representation corresponding to the input chords provided and recognized chords.

In order to perform the chord detection from the input audio samples, a large dataset of various audio samples of harmonium chords is required for training the Convolutional Neural Network. The CNN and RNN are the special cases of DNN which consider multiple input features on the basis of which the neural networks are trained in a very fast manner.

The training Phase of the CNN & RNN is as follows:

1. The various audio input samples are first processed for extracting various features from the sample and storing them in a reserved format of a .csv file which then is further used for training the network.

2. The audio samples are processed for silence removal and pitch-class profiling for pitch identification.

3. The extracted features are then correlated with the trained samples to fit the prediction, so the feature values are passed to the prediction module of the CNN as well as the RNN for predicting the chord in the input audio sample.

4. The obtained chord prediction is passed to the Lilypond library of Python to get the sheet music representation for music composers to get ready to use the representation for training musicians and composing music.

The existing systems make use of the extraction of features from the audio samples manually. The major features of the audio are CQT and Onset Detection for Pitch Class Identification. The various process taking

1. Audio Input and Silence Removal.
2. Audio Feature Analysis for Identifying Sampling Frequency and Frame rate.
3. Constant Q-Transform.
4. Onset Detection.
5. Note and Pitch Detection.
6. Chord Detection.

1. Audio Input and Silence Removal:

The silence removal method is important for feature extraction. First the audio file is converted into an audio signal. In this proposed method it is achieved by librosa.effects.trim. The effects.trim is used for trimming leading and trailing silence from an audio signal. With the use of matplotlib plot the audio signal is plotted with the preceding and trailing silence removed. This is shown in Figure 5.

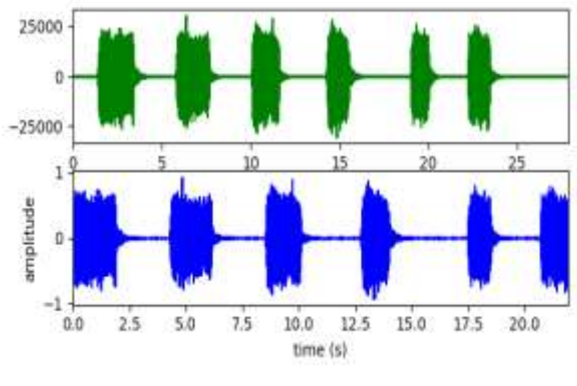


Fig.5. Silence Removed from input audio file.

Algorithm 1 Silence Removal Algorithm from Input Audio

Determine silence 'w' from the input audio file 'I' based on the amplitude of signal 'a'.

Input: The user chosen audio input file 'I'.

I ← new audio input file.

```
librosa.load(I) ←timeline, amplitude values
I’, index←librosa.effects.trim
(amplitude values, threshold decibel, frame_ length)
librosa.output.write_wav(I’)
```

2. Analysis on Audio file

In Audio Analysis after the silence is removed, the audio signal is visualized on the basis of the frequency spectrum and the positive and negative frequencies of the audio signal. Other parameters like frequency sampling, the duration of signal, channels and the time step between samples is analyzed. Figure 6 shows the audio file analysis.

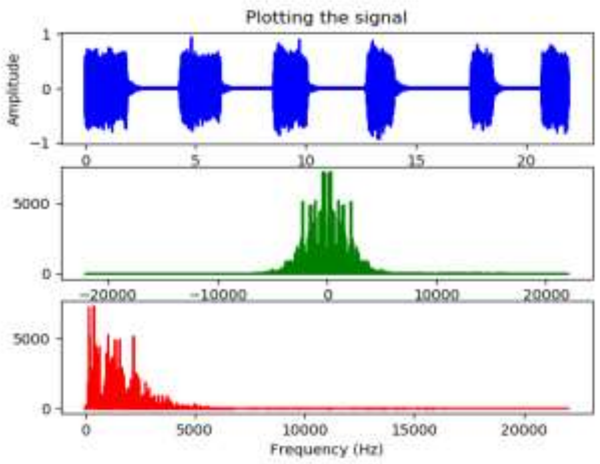


Fig. 6. Audio file analysis for frame rate and frequency detection

Algorithm 2 Analysis of audio file

Determine the sampling rate ‘fsrate’ based on signal ‘s’ value from the silence removed input file I’.

Input: The silence removed audio file I’.

```
Fsrate, s ←scipy.io. wavfile. read (I’)
Secs←len (s. shape)
ts←1.0/fsrate
FFT←scipy.fft (secs, ts)
freq←scipy. fftpack. fftfreq(secs, ts)
```

3. Constant Q-Transform:

Constant Q- Transform i.e. CQT being very dissimilar to the Fourier Transform, but very much like Mel Scale. A systematically efficient method for computing the constant-Q transform of a time-domain signal. CQT is a very important representation in the time-frequency domain where, the frequency slots are mathematically spaced & the numeric factors i.e. Q-Ratios (ratios of the center frequencies to band-widths) of all slots are equitable. An inverse transform is proposed which allows an inexpensive quality (around 55dB signal-to-noise ratio) reconstruction of the particular signal from its CQT coefficients. Figure 7 shows the Constant Q transform Chord spectrogram

Algorithm 3 Analysis of CQT Spectrogram

```
fmin=librosa.midi_to_hz(36)
hop_length=512
```

```
C =librosa.cqt(x, sr=sr, fmin=fmin, n_bins=72, hop_length=hop_length)
```

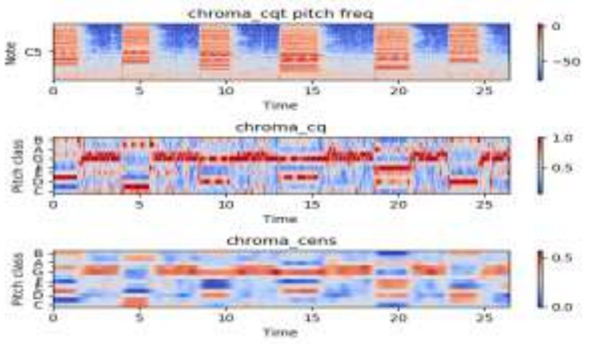


Fig. 7. Constant Q transform Chord spectrogram

The Chromagram can be constructed as a spectrogram, which presents the relationship between time and frequency spectrum. The frequency spectrum is formed by 12-dimensional Chroma vectors, which is a set of pitch-classes {C, C#, D, D#, E, F, F#, G, G#, A, A#, B}, and each element of the vector shows the strength of the input. The computation of the Chromagram is done by calculating the frequency and amplitudes of the corresponding note from the spectrogram. There are multiple ways to calculate it as discussed in the literature. Short-time Fourier Transform, Constant Q Transform and Fast Fourier Transform (FFT) are the three common ones that people use. We have implemented two of them during the progress of this project, the CQT method, and the FFT method. The CQT method is the approach that we have chosen for template matching later. Partial code for FFT has also been attached to the end of the chromagram.py file. A good thanks to visualize the Chromagram may be a two-dimensional image, showing time or number of frames on the x-axis and 12 pitch classes on the y-axis. We can easily identify which pitches are the strongest according to the color. Unlike the Fourier transform, but the same as the Mel- scale, the CQT uses a logarithmically spaced frequency axis.

4. Onset Detection:

Onset is the start of a musical note or an audio signal. All musical notes have an onset spotting of various tasks. An audio signal is the most crucial task in music information retrieval. The start marks the beginning of the transient part of a sound or the earliest moment when a transient can be reliably detected. In this proposed system the onset detection is attained with librosa.onset.onset_detect. Basically, it calculates a spectral novelty function and discovers peaks in the spectral novelty function. Then it backtracks from every peak to a preceding local minimum. It can be helpful for finding segmentation points such that the onset occurs shortly after the beginning of the segment. A plot of onsets with the time-domain waveform is shown in Figure 8.

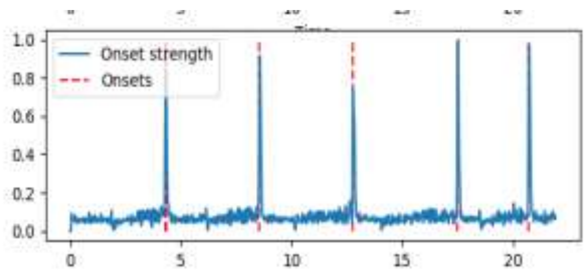


Fig. 8. Plot of Onset Detection

5. Musical Note Identification:

A chord, set of synchronized tones, & a series of chords over-time or chord progression forms the base of the harmony in music composition. Hence, for the analysis of the overall harmonic structure of a musical composition one often starts with the classification of each chord in the composition.

In this proposed technique the chords are identified from the given audio signal. The .wav file is opened with scipy.io.wavfile.read. It returns the sample rate (in samples/sec) and data from a .wav file. The data is returned as a NumPy array.

Algorithm 4 Pitch Profile Classifier for Note Identification

```
Determine the pitch of the profile based on the Input audio file and the I'.
Input: Analyzed Audio Data 'A'
Fsrate,s ←scipy.io.wavfile.read(I')
pcp←GeneratePcp (fsrate,s)
PcpNormalised←NormalizePcp (pcp)
```

Algorithm 5 GeneratePcp () and NormalizePcp (pcp)

```
Determine the pitch based on sampling rate and frequency
**GeneratePcp ()**
Input: sampled frequency
for sample =1 to N:
if (freq(sample) == 0):
return -1
else:
return (12*log2 (fs*1) / (N * freq(sample))) % 12
**NormalizePcp (pcp) **
Forpcp[index=0 to N samples]:
PcpNormalized←pcp[index]/sum(pcp)
return PcpNormalized
```

The Chromagram can be created as a spectrogram on the data of the audio file, which represents the association between the time and frequency spectrum. The frequency spectrum is formed by 12-dimensional Chroma vectors, which is a set of chord classes {C, C#, D, D#, E, F, F#, G, G#, A, A#, B}, and each individual component of the vector shows the strength of the input. The computation of the Chromagram is to analyze the frequency and amplitudes of the corresponding note from the spectrogram of the audio signal. A comparison between the typical distributions of chords and the vectors of a Chromagram detects the correct chord. The frequency distribution of the notes in different octaves of a harmonium is as mentioned in Table II:

TABLE II Frequency division of 3 octaves in harmonium

Notes of Harmonium	1st Octave	2nd Octave	3rd Octave
C	130	260	520
C#	141	280	558
D	148	296	590
D#	158	314	626

E	167	332	664
F	176	351	702
F#	186	372	744
G	196	394	787
G#	209	419	837
A	222	443	884
A#	235	470	941
B	251	500	995

6. Pitch Detection

The Pitch can be taken as the degree of sound frequency expressed in terms of Hertz. The higher the frequency is, the upper the pitch. Sound produces some waves that are measured with reference to the frequency it carries. Pitch can be defined as the position of a musical note in a musical scale.

The sample rate (in samples/sec) and data is extracted from an audio sample. Each chord has a unique frequency hence each chord has a different pitch. The spectrogram is created with different frequencies of an audio signal and with an analysis of the spectrogram, it is possible to detect the pitch and frequency of the chord. 6.1. Integration of Lilypond with python for generating Music Sheet

Lilypond, the tool for programmatically generating music sheets for musicians and music composers has taken an initiative to have the tool interface with Python for programmatic usage of the tool for music sheet generation. Lilypond has library names such as python.ly for being used with Python and the syntax of the Lilypond music sheet. The determined notes and chords are passed as input to the Lilypond script to be written and stored as .ly file. This .ly file is then executed using the Lilypond command. Chords in Lilypond can be represented as follows:

\chordmode{ c e g}

V. DATASET AND EXPERIMENTAL RESULTS

1. Database

For optimization of the harmonium note, the audio signal minimizes the intra-note variation in the starting stage. We have provided a few samples from every octave (Mandhar, Madhya, and Taar). For each of the 12 notes, a total of 900 multi chord recordings of harmonium were made with durations of about 8 to 10 secs or 20 to 25 seconds each with a sampling frequency of 44100Hz. The dataset is completely synthetic and is generated by professional artists over a period of time. It consists of various chords from the musical combinations and therefore can be used as input samples for training the proposed system CNN. The chords and the chord combinations used for training the neural network are so huge in a quantity that the system can easily find the appropriate chord played from the input recording samples. A few of the combinations of chords are mentioned in Tables III, IV, and V below:

TABLE III. Mandhar Saptak A Few Sample Chords from the Database

B major B D# F#	D major D F# A	E major E G# B	F# major F# A#C#	G# major G# B#D#	B major B D F

F# major F# A#C#	C major –C E G	G major G B D	C# major C# E#G#	G# major G# B#D#	D major – D F# A
A major A C# E	E major – E G Bb	B major – B D F	E major E G# B	B major B D# F#	F major – F A C
E major E G Bb	C major – C E G	A major – A C# E	F# major F# A# C#	E major E G# B	C# major C# E# G#

TABLE IV. Madhya Saptak A Few Sample Chords from the Database

F major F A C	E major E G# B	Eb major E G Bb	D major D F# A	C# major C# E#G#	C major C E G
B major B D# F#	Bb major Bb D F	A major A C# E	G# major G# B#D#	G major G B D	F# major F# A#C#
G# major G# B#D#	D major D F# A	G major G B D	C# major C# E# G#	F# major F# A# C#	C major C E G
B major B D# F#	F major – F A C	B major B D F	E major – E G# B	A major A C# E	E major E G B
E major E G# B	C# major C#E#G #	A major A C# E	F# major F# A# C#	Eb major Eb G Bb	C major C E G

TABLE V. Taar Saptak A Few Sample Chords from the Database

A major A C# E	E major E G Bb	B major B D F	E major E G# B	B major B D# F#	F major – F A C
E major E G Bb	C major C E G	A major – A C# E	F# major F# A# C#	E major – E G# B	C# major C# E# G#

B major B D F	G major G B D	F major – F A C	D major D F# A	B major B D# F#	G# major G# B# D#
G major – G B D	C major – C E G	Bb major – Bb D F	Eb major – Eb G Bb	D major – D F# A	F# major F# A# C#

Once the recordings deliver contented outcomes, so as to test the algorithm for intra-note variations & variations due to change in type of harmonium, in which we have recorded the three octaves for the different chords. Table VI shows the distribution of the database.

TABLE VI. Database division

Database samples	Distribution	No. of samples
Training	Training	700
	Validation	100
Testing	Testing	100

2. Neural Network Parameter Tuning:

From the features obtained, the parameters mentioned aloft, a total of 5 feature vector was obtained. In this research, an optimal selection of features is represented, other than selecting all features. The different mixture of the feature vectors is considered as an input to the CNN and the RNN. The mixture of features that are extracted is used for maximum accuracy and tested against the testing slices. These feature vectors behave as input to the neural network. [19].After extensive tuning and trying different parameters for the neural network, the following parameters were chosen as optimal for the desired outcomes with the least error. In this system, one of the well-known machine learning algorithms out there which is cast-off for harmonium musical chord classification i.e. Convolutional recurrent Neural Network is also known as (CRNN).

VI. RESULTS

This experiment depicts the proposed system was provided with various recordings as input where the recordings comprised of five to six different chords from every octave (Mandhar, Madhya, and Taar) to form various combinations of chords. Every recording has a different chord sequence and different chord combinations. Before the recordings are given as input, every recording is divided into chunks based on the detected onsets, and accordingly the features of every chunk are taken into consideration for training.

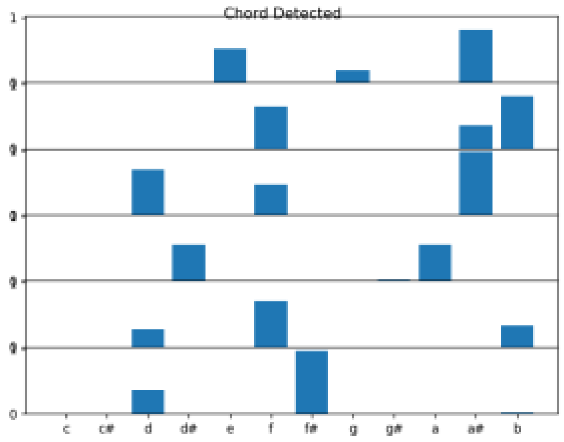


Fig. 9. Plot of the determined chords

This makes sure that a particular chunk, found after or before any other chunk (chord), can be still recognized accurately. As per the trained CNN & RNN Model, the determining notes of pitches are plotted graphically in Figure 9.

Figure 9 shows the detected pitch from the audio input and the corresponding energy with which it is used in the audio sample. The detected polyphonic chord is plotted in Figure 10. The detected chord of the input audio sample comprises of multiple notes as the input audio sample is a polyphonic audio sample. The following output chunks are detected for the input sample for which the chord sequence was C major – C E G, C# major – C# D# G#, D major – D F# A, E major – E G# B, F major – F A C and the charts display the results retrieved from the test sample.

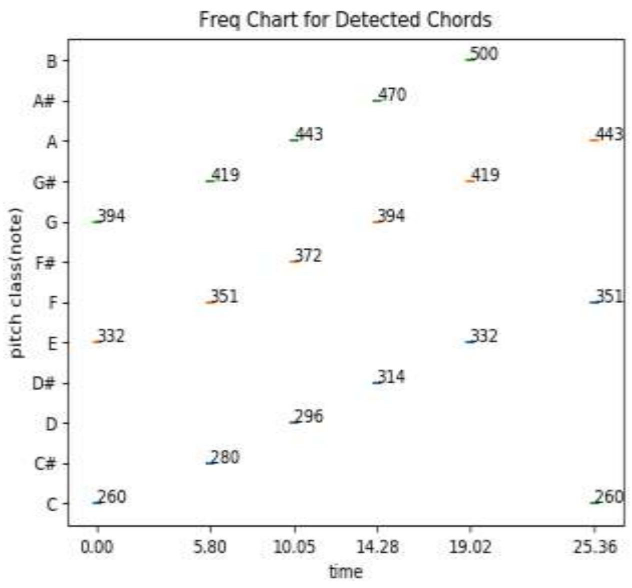


Fig.10. Frequency Plot for Detected Chords from corresponding octave

The results obtained for the predicted chord from the above classification results are passed on to the Lilypond library to generate the final music sheet representation of the predicted chords and musical notations so that obtained result can be used by musicians or music composers for generating music from the symbolic representation of musical notes and chords. The proposed system makes use of SK-learn and sci-kit learn modules along with the Tensor Flow module for implementation of the predictive model and CRNN for the training of the sample and chord prediction. The results obtained for the predicted chord from the above classification results are passed on to the Lilypond library to generate the final music sheet representation of the predicted chords and musical notations. Figure 11. Shows the music sheet generated by the system for the entire chord sequence.

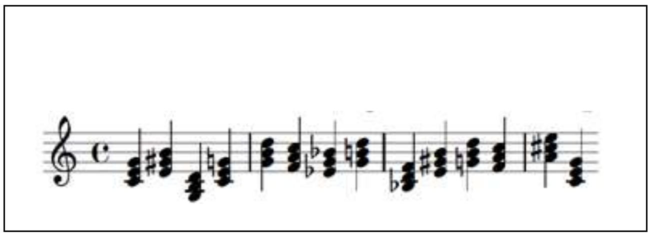


Fig. 11. Harmonium Musical Chord Mode Notation Sheet of an audio signal.

VII. EVALUATION METRICS

The proposed system evaluates the precision & recall for determining the efficiency of the implementation of the proposed system. Precision tries to solve the following:

What proportion of positive identifications was actually right?

Precision is defined as:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (4)$$

Recall tries to solve the following question: What proportion of original positives was identified correctly?

Mathematically, recall is defined as:

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (5)$$

Precision & recall values of independent implementation mechanisms are as displayed in the Table VII. Figure 12 is a graphical representation of different models of neural networks that have been used in this technique.

F1 Score is defined as the score which details regarding the performance of the system which is calculated with respect to the true negative, true positive, False negative and false positive. Mathematically, F1 Score is represented as follows:

$$F1Score = 2 * \frac{(precision * recall)}{(precision + recall)} \quad (6)$$

TABLE VII. Evaluation Metrics Using Precision & Recall

Parameter	Precision	Recall	F1 Score
CNN	0.45	0.67	0.61
RNN	0.53	0.72	0.64
CRNN	0.63	0.86	0.77

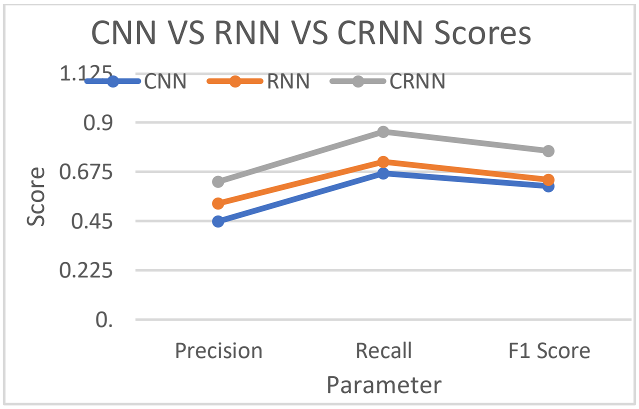


Fig.12.Evaluation for Different Models of neural networks

Accuracy: Accuracy is an instinctive measurement of performance and it is merely a ratio of an observation that has been predicted correctly to the total of the observations. The consideration is that the higher the accuracy, the better the model. It is indeed true that accuracy is a great measure but, only when one has symmetric datasets where values of false positive and false negatives are almost similar. Therefore, one has to look at other parameters to evaluate the performance of a Model. For our model, we achieved an accuracy rate of approximately 94%.

$$Accuracy = \frac{TruePositive+TrueNegative}{TruePositive+FalsePositive+FalseNegative} (7)$$

As per the trained CNN & RNN Models the determined pitch is plotted graphically in the figure 9. The figure shows the detected pitch from the audio input and the corresponding energy with which it's used in the audio sample. The detected polyphonic chord in a single chunk is plotted in the figure 10. The detected chord of the input audio sample comprises of multiple notes as the input audio sample is a polyphonic audio sample.

VIII. CONCLUSION

In this paper, the pre-processed audio file is used to determine advanced parameters like onset times, note detection and frequency analysis. The desired parameters of the audio file are then given as input to the neural network for training and the trained dataset is obtained for predicting and estimating the chord from the test dataset. As per the analysis, for the feed-forward neural network generated so far, the proposed system achieves an accuracy of up to 88%. The extracted features are classified using the Feed Forward Neural Network algorithm. In comparison, to the previous techniques proposed for monophonic and polyphonic piano note detection, is being proposed for polyphonic harmonium note detection has gained more efficiency as there are a few octaves to be processed in a harmonium in comparison with a piano & therefore, the parameters to be compared are a few in number in order to decide the note from 3 octaves. The proposed system is also quite faster as compared to existing systems as the number of training samples required will be less as compared with a piano and thus the time required to predict the note will be less as the number of iterations gets reduced.

IX. FUTURE DIRECTIONS

The results of the tested recordings show that the proposed system will determine the chord sequences from the recordings which are played at a slower pace and the working of this will require vigorous input of raga recordings which will not only determine the chord sequence, but will also determine the played raga in the test sample.

X. REFERENCES

[1] S. Sigtia, E. Benetos and S. Dixon, "An End-to-End Neural Network for Polyphonic Piano Music Transcription," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 24, no. 5, May 2016, pp. 927-939.

[2] Siddharth Sigtia, Emmanouil Benetos, Nicolas Boulanger-Lewandowski, Tillman Weyde, Artur S. d'Avila Garcez, Simon Dixon," A Hybrid Recurrent Neural Network For Music Transcription", School of Electronic Engineering and Computer Science Centre for Digital Music (C4DM) .

[3] Sebastian Böck ; Markus Schedl, "Polyphonic piano note transcription with recurrent neural networks", 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 25-30 March 2012.

[4] Fu'ze Cong ; Shuchang Liu ; Li Guo ; Geraint A. Wiggins, "A Parallel Fusion Approach to Piano Music Transcription Based on Convolutional Neural Network", : 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 15-20 April 2018.

[5] R. Sridhar and T. Geetha, "Swara identification for south Indianclassical music," in Information Technology, 2006. ICIT'06. 9th InternationalConference on. IEEE, 2006, pp.

[6] Adrien Ycart and Emmanouil Benetos, "A study on lstm networks for polyphonic music sequence modelling", Centre for Digital Music, Queen Mary University of London, UK.

[7] J. Xu, B. Tang, H. He, and H. Man, "Semi-Supervised Feature Selection Based on Relevance and Redundancy Criteria," IEEE Trans. on Neural Networks and Learning Systems (TNNLS), vol. 28, issue 9, pp. 1974 - 1984, 2017.

[8] Ashwini S. Deo, "The metrical organization of Classical Sanskrit verse", Journal of Linguistics, March 2007, pp. 1-58.

[9] Savitha S Upadhya "Pitch Detection in Time and Frequency Domain"2012 International Conference on Communication, Information & Computing Technology (ICCICT), Oct. 19-20, Mumbai, IndiaBethanie Hansen, "Introduction to MusicAppreciation", American Public University System ePress, revised edition 2014.

[10] S. Shetty and K. Achary, "Raga mining of Indian music by extracting arohana-avarohana pattern," International Journal of Recent Trends inEngineering, vol. 1, no. 1, pp. 362–366, 2009.

[11] Li Hui, Bei-qian Dai, Lu Wei"A Pitch Detection Algorithm Based On AMDF AND ACF"142440469X/06/$20.00 ©2006 IEEE, ICASSP 2006

[12] A. Cogliati and Z. Duan, "Piano music transcription modeling note temporal evolution," 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, 2015, pp. 429-433.

[13] Emmanouil Benetos, Simon Dixon, Holger Kirchhoff, Anssi Klapuri, "Automatic Music Transcription: Challenges and Future Directions", Journal of Intelligent Information Systems, Vol. 41, Issue 3, pp 407-434, December 2013.

[14] A. Klapuri and M. Davy, Signal processing methods for music transcription. Springer Science & Business Media, ISBN-10: 0-387-30667-6, 2007

[15] T. Berg-Kirkpatrick, J. Andreas, and D. Klein, "Unsupervised transcription of piano music," in Advances in Neural Information Processing Systems (NIPS), pp. 1538–1546,2014.

[16] Gowrishankar B. S. and Dr. Nagappa U Bhajantri "An Exhaustive Review of Automatic Music Transcription Techniques" International Conference on Signal Processing, Communication

[17] Alan Marsden, "Music analysis by computer: ontology and epistemology", © Springer International Publishing Switzerland 2016 D. Meredith (ed.), Computational Music Analysis, DOI 10.1007/978-3-319-25931-4_1

[18] D. G. Morin, "Deep neural networks for piano music transcription," Jun. 2017. Available: https://github.com/diegomorin8/Deep-Neural-Networks-for-Piano-Music-Transcription

[19] S. B. Puri, S. P. Mahajan (2019) "Optimum Feature Selection for Harmonium Note Identification Using ANN", 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT).

[20] S. B. Puri, S. P. Mahajan (2017) "Review on Automatic Music Transcription System", 2017.International Conference on Computing, Communication, Control and Automation (ICCUBEA)