Segregating Musical Chords for Automatic Music Transcription: A LSTM-RNN Approach

Himadri Mukherjee[1](✉), Ankita Dhar[1], Sk. Md. Obaidullah[2], K.C. Santosh[3],
Santanu Phadikar[4], and Kaushik Roy[1]

[1] Department of Computer Science, West Bengal State University, Kolkata, India
himadrim027@gmail.com,ankita.ankie@gmail.com,kaushik.mrg@gmail.com

[2] Department of Computer Science and Engineering, Aliah University, Kolkata, India
sk.obaidullah@gmail.com

[3] Department of Computer Science, The University of South Dakota,
Vermillion, SD, USA
santosh.kc@ieee.org

[4] Department of Computer Science and Engineering, Maulana Abul Kalam Azad
University of Technology, Kolkata, India
sphadikar@yahoo.com

Abstract. Notating or transcribing a music piece is very important for musicians. It not only helps them to communicate among each other but also helps in understanding a piece. This is very much essential for improvisations and performances. This makes automatic music transcription systems extremely important. Every music piece can be broadly categorized into two parts namely the lead section and the accompaniment section or background music (BGM). The BGM is very important in a piece as it sets the mood and makes a piece complete. Thus it is very much important to notate the BGM for properly understanding and performing a piece. One of the key components of BGM is known as chord which is constituted of two or more musical notes. Every composition is accompanied with a chord chart. In this paper, a long short term memory-recurrent neural network (LSTM-RNN)- based approach is presented for segregating musical chords from clips of short durations which can aid in automatic transcription. Experiments were performed on over 46800 clips and a highest accuracy of 99.91% has been obtained for the proposed system.

Keywords: Chord identification · Music signal · LSTM-RNN

1 Introduction

A music piece is composed of musical notes. These notes occur in different combinations and timings which makes melodies different. In order to study such music compositions it is very important to notate or transcribe them. This not

B. Deka et al. (Eds.): PReMI 2019, LNCS 11942, pp. 427–435, 2019.
https://doi.org/10.1007/978-3-030-34872-4_47

only helps in understanding them in a better way but also to communicate with other musicians. The BGM of a composition as important as the lead melody. It is the BGM which makes a piece sound complete and which goes on for almost the entire span of a piece. A change in the BGM can alter the mood of a composition and at times disrupt it completely. Thus it is very important to play the BGM flawlessly during performances to uphold the essence of a composition. One of the most important facets of BGM melody is known as a chord which is composed of two or more musical notes played simultaneously. Every composition has a chord chart associated with it whose transcription is essential.

Rajpurkar et al. [15] distinguished chords in real-time. They used hidden markov model (HMM) and Gaussian discriminant analysis in addition to chroma-based features and obtained an accuracy of 99.19%. Zhou and Lerch [18] used deep learning for distinguishing chords. They worked with 317 music pieces and obtained a recall value of 0.916 using max-pooling. Cheng et al. [4] distinguished chords for music classification and retrieval with the aid of N-gram technique and HMM. Different chord-based features like chord histogram and LCS were also involved in their experiments and a highest overall accuracy of 67.3% was obtained. Dylan Quenneville [14] has talked about multitudinous aspects of automatic music transcription. He has highlighted the basics of making music as well as that of transcription. He has talked about different techniques of pitch detection in the thick of fourier transform-based approaches, fundamental frequency-based approaches, harmonicity-based approaches to name a few.

Berket and Shi [3] presented a two phase model for music transcription. In the first phase, they used acoustic modelling to detect pitches and in the later phase it was transcribed. They worked with 138 MIDI files which were converted to audio. The train set consisted of 110 songs while the remaining were used for testing and reported results as high as 99.81%. Wats and Patra [17] used a non negative matrix factorization-based technique for automatic music transcription. They worked on the Disklavier dataset and obtained good results. Benetos et al. [1] presented an overview of automatic music transcription. They have touched on its various applications and challenges. They have also talked about several transcription techniques as well. Muludi et al. [12] frequency domain information and pitch class profile for chord identification. Their experiments involved 432 guitar chords and obtained an accuracy of 70.06%.

Osmalskyj et al. [13] used a neural network and pitch class profiles for guitar chord distinction. Their study involved other instruments in the thick of accordion, violin and piano. They performed instrument identification as well and obtained an error rate of 6.5% for chord identification. Benetos et al. [2] laid out different techniques and challenges which are involved in automatic music transcription. They have talked about various pitch tracking methods in the thick of feature-based approaches, statistical approaches, spectrogram factorization-based approaches and many more. They have also talked about several types of transcriptions including instrument and genre-based transcription as well as informed transcription. Kroher and Gomez [7] attempted to automatically transcribe flamenco singing from polyphonic tracks. They extracted predominant melody and eliminated contours of the accompaniments. Next the vocal contour was discretized into notes followed by assignment of a quantized pitch level. They experimented with three datasets totaling to more than 100 tracks and obtained results which was better than state of the art singing transcribers based on overall performance, onset detection and voicing accuracy. Costantini and Casali [5] used frequency analysis for chord identification. Experiments were performed with upto 4 note chords. Highest accuracies of 98%, 97% and 95% were obtained for the 2, 3 and 4 note chords.

Here, a system is proposed to distinguish chords from clips of very short duration. It works with LSTM-RNN based classification and has the potential of aiding in automatic music transcription for background music which is very vital. The system is illustrated in Fig. 1.

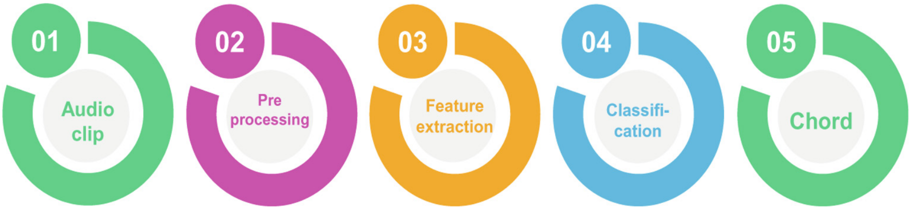

Fig. 1. Pictorial representation of the proposed system.

The rest of the paper consists of the details of dataset in Sect. 2. Sections 3 and 4 talk about the proposed method whose results respectively. Finally we have concluded in Sect. 5.

2 Dataset

Data is a very important aspect of any experiment. The quality of data plays a crucial part in development of robust systems as well. To the best of our knowledge, there is no publicly available dataset of chords and hence we put together a dataset of our own. In the present experiment, we consider two of the most popular chords from the major family (C and G) and two most popular chords from the minor family namely A minor (Am) and E minor (Em) [16]. The constituent notes of scales of the considered chords along with the notes of the chords is presented in Table 1. The chord pairs (G-Em) and (C-Am) have common notes which makes it difficult to distinguish them.

Volunteers were provided a Hertz acoustic guitar (HZR3801E) for playing the chords. They played different rhythm patterns and no metronome was used to allow relaxation with respect to tempo. Volunteers further used different type of plectrums which slightly change the sound thereby encompassing more variation. The audios were recorded with the primary line port of a computer having a motherboard (Gigabyte B150M-D3H). Further, studio ambience and use of

Table 1. Notes involved in the chords.

Scale	Notes	Chord	Similar notes
G	G,A,B,C,D,E,F#	G, B, D	G,B
Em	E,F#,G,A,B,C,D	E, G, B	G,B
C	C,D,E,F,G,A,B	C, E, G	C,E
Am	A,B,C,D,E,F,G	A, C, E	C,E

pre amplifiers was avoided to ensure real world scenario. The audio clips were recorded in .wav format at a bitrate of 1411 kbps.

Four datasets (D1-D4) having clips of lengths 0.25, 0.5, 1 and 2 s respectively were put together form the recorded data whose details are presented in Table 2. We worked with clips of such durations to test the efficiency of our system for short clips which is common in real world.

Table 2. Details of the generated datasets with number of clips per chord.

Datasets (length of clip in second)	Chords				
	C	G	Am	Em	Total
D1(0.25)	11613	11863	11537	11871	46884
D2(0.5)	5804	5928	5762	5930	23424
D3(1)	2899	2959	2876	2961	11695
D4(2)	1444	1476	1434	1475	5829

3 Proposed Method

3.1 Preprocessing

Framing. The clips were first subdivided into smaller segments called frames. This was mainly done to make the spectral contents stationary which otherwise show high deviations thereby making analysis a herculean task. The clips were divided into 256 point frames in overlapping mode with 100 common points (overlap) between two consecutive frames [11].

Windowing. Jitters are often observed in the frames due to loss of continuity at the boundaries. These disrupt frequency-based analysis in the form of spectral leakage. To tackle this, the frames are windowed with windowing function. Here we used hamming window [11] which is presented in Eq. (1).

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right), \tag{1}$$

Feature extraction where n is a sample point within a N sized frame.

3.2 Feature Extraction

Each of the clips were used for extraction of the standard line spectral frequency (LSF) features at frame level. LSF [11] was chosen due to its higher quantization power [10]. Here, a sound signal is represented as the output of a filter H(z) whose inverse is G(z) where G $_{1....m}$ are the predictive coefficients

$$G(z) = 1 + g_1 z^{-1} + . + g_m z^{-n} \tag{2}$$

The LSF derived by decomposing G(z) into G$_x$(z) and G$_y$(z) which are detailed below

$$G_x(z) = G(z) + z^{-(m+1)} G(z^{-1}) \tag{3}$$

$$G_y(z) = G(z) - z^{-(m+1)} G(z^{-1}) \tag{4}$$

We had extracted 5, 10, 15, 20 and 25 dimensional features for the frames. Each of these dimensions correspond to bands that is 5 dimensional LSFs have 5 bands and so on. Next, these bands were graded in accordance with the total value of the coefficients. This band sequence was used as feature. It depicted the energy distribution pattern across the bands. Along with this, the mean and standard deviation of the spectral centroids per frame was also appended. When 5 dimensional LSF was extracted, a total of $5 \times 440 = 2200$ coefficients were obtained for a clip of only 1 s (1 s clip produced 440 frames). This dimension varied with disparate length of the clips. The band grades along with the mean and standard deviation of the centroids produced a $5+2 = 7$ dimensional feature when 5 dimensional LSFs were extracted. These were also independent of the clip lengths. So finally we obtained features of 7, 12, 17, 22 and 27 dimensions.

3.3 Long Short Term Memory-Recurrent Neural Network (LSTM-RNN) Based Classification

LSTM-RNN can preserve states as compared to standard neural networks [9] which makes them suitable for sequences. It further solves the vanishing gradient problem of simple RNNs [8]. A LSTM block comprises of a cell state and three gates namely forget gate, input gate and output gate. The input gate (i_n) helps to generate new state:

$$i_n = \sigma(Wt_i S_{n-1} + Wt_i X_n), \tag{5}$$

where Wt_i is the associated weight. The forget gate discards values form previous state to the present state:

$$f_n = \sigma(Wt_f S_{n-1} + Wt_f X_n), \tag{6}$$

where Wt_f is the associated weight. The output determines the next state as shown below:

$$o_n = \sigma(Wt_o S_{n-1} + Wt_o X_n), \tag{7}$$

where Wt_o is the associated weight. Our network comprised of a 100 dimensional LSTM layer. The output of this layer was passed through three fully connected layers of dimensions 100, 50 and 25 respectively. These layers had ReLU activation. The final layer was a 4 dimensional fully connected layer with softmax activation. We had initially used 5 fold cross validation with 100 epochs in our experiment and the network parameters were set after trials.

4 Result and Analysis

Each of the feature sets for the datasets D1-D4 were fed to the recurrent neural network as summarized in Table 3. It is observed that the best result was obtained for the 22 dimensional features on D3. To obtain better results, the training epochs were varied with 5 fold cross validation for 22 dimensional features of D3 as shown in Table 4. The best performance was obtained for 300 epochs. Increasing the training epochs even further led to over fitting and thus produced lower results. The confusions among the different classes for 300 iterations is presented in Table 5(a). It is observed that the highest confusion was among the minor chords. The clips were analyzed and it was found that the volunteers at times accidentally muted strings which interfered with the chord textures in the barred shapes. This could be one probable reason for such confusions.

Table 3. Results for different datasets with the disparate datasets.

Datasets	Feature dimensions				
	7	12	17	22	27
D1	62.03	74.36	79.29	75.41	77.74
D2	68.40	93.31	91.79	76.20	84.02
D3	71.01	85.16	93.64	**95.50**	92.70
D4	73.01	75.23	85.80	83.02	91.85

Table 4. Accuracy for different training epochs on D3 with 22 dimensional features.

Epochs	100	200	**300**	400	500
Accuracy (%)	95.50	95.45	**95.90**	94.50	94.24

In order to obtain further improvements, we varied the cross validation folds for 100 epochs for 22 dimensional features of D3. The obtained results are presented in Table 6. 20 folds produced the best result wherein the variation of the dataset was evenly distributed. The performance decreased on further increasing the folds of cross validation. The interclass confusions is presented in Table 5(b)

Table 5. (a) Confusion matrix for 300 epochs. (b) Confusion matrix for 20 fold cross validation. (c) Confusion matrix for 300 epochs 20 fold cross validation.

	C	G	Am	Em
C	2897	2	0	0
G	16	2903	40	0
Am	0	12	2840	24
Em	0	0	386	2575

(a)

	C	G	Am	Em
C	2899	0	0	0
G	9	2950	0	0
Am	0	3	2873	0
Em	0	0	0	2961

(b)

	C	G	Am	Em
C	2899	0	0	0
G	8	2951	0	0
Am	0	3	2873	0
Em	0	0	0	2961

(c)

wherein it is observed the chords C and Em were recognized with 100% accuracy. The confusions among the minor chords was also overcome in this setup. Finally the best fold value (20 fold) along with the best training epoch (300 epochs) were combined which produced an accuracy of 99.91 % (overall highest) whose confusions are presented in Table 5(c). It is observed that the confusions were exactly similar as compared to the 20 fold cross validation setup, only 1 more instance of G chord was identified correctly as compared to the former setup. Some of the other popular classifiers in the thick of bayesnet (BN), naïve bayes (NB), multi layer perceptron (MLP), random forest (RF), radial basis functional classifier (RBF) from [6] were also evaluated on D4 whose results are summarized in Table 7.

Table 6. Accuracy for different folds of cross validation on D3 with 22 dimensional features.

Folds	5	10	15	**20**	25
Accuracy (%)	95.50	99.63	99.48	**99.90**	99.85

Table 7. Performance of different classification techniques on D3 with 22 dimensional features.

Classifier	NB	BN	MLP	RF	RBF	**LSTM-RNN**
Accuracy (%)	42.26	80.86	93.63	98.97	80.47	**99.91**

5 Conclusion

Here, a system is presented to distinguish chords from clips of short durations. The system works with LSTM-RNN based classification technique and produced encouraging results. In future, we will experiment with a larger set of chords and involve other instruments as well. We will introduce other tracks along with the chords to observe the system's performance. We also plan to identify and discard silent sections in the clips to obtain better results. Finally, we will make use of other acoustic features coupled with different modern machine learning techniques to obtain further improvement in our results.

Acknowledgement. The authors would like to thank Mr. Soukhin Bhattacherjee, Mr. Debajyoti Bose of Department of Electrical, Power & Energy, University of Petroleum and Energy Studies for their help during the entire course of this work. They also thank WWW.PresentationGO.com for the block diagram template.

References

1. Benetos, E., Dixon, S., Duan, Z., Ewert, S.: Automatic music transcription: an overview. IEEE Sig. Process. Mag. **36**(1), 20–30 (2018)
2. Benetos, E., Dixon, S., Giannoulis, D., Kirchhoff, H., Klapuri, A.: Automatic music transcription: challenges and future directions. J. Intell. Inf. Syst. **41**(3), 407–434 (2013)
3. Bereket, M., Shi, K.: An AI approach to automatic natural music transcription (2017)
4. Cheng, H.T., Yang, Y.H., Lin, Y.C., Liao, I.B., Chen, H.H.: Automatic chord recognition for music classification and retrieval. In: 2008 IEEE International Conference on Multimedia and Expo, pp. 1505–1508. IEEE (2008)
5. Costantini, G., Casali, D.: Recognition of musical chord notes. WSEAS Trans. Acoust. Music **1**(1), 17–20 (2004)
6. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. ACM SIGKDD Explor. Newsl. **11**(1), 10–18 (2009)
7. Kroher, N., Gómez, E.: Automatic transcription of flamenco singing from polyphonic music recordings. IEEE/ACM Trans. Audio Speech Lang. Process. (TASLP) **24**(5), 901–913 (2016)
8. Li, J., Mohamed, A., Zweig, G., Gong, Y.: LSTM time and frequency recurrence for automatic speech recognition. In: 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), pp. 187–191. IEEE (2015)
9. Lipton, Z.C., Berkowitz, J., Elkan, C.: A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:1506.00019 (2015)
10. Mukherjee, H., Dutta, M., Obaidullah, S.M., Santosh, K.C., Phadikar, S., Roy, K.: Lazy learning based segregation of Top-3 south indian languages with LSF-a feature. In: Santosh, K.C., Hegadi, R.S. (eds.) RTIP2R 2018. CCIS, vol. 1035, pp. 449–459. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-9181-1_40
11. Mukherjee, H., Obaidullah, S.M., Santosh, K., Phadikar, S., Roy, K.: Line spectral frequency-based features and extreme learning machine for voice activity detection from audio signal. Int. J. Speech Technol. **21**(4), 753–760 (2018)
12. Muludi, K., Loupatty, A.F.S., et al.: Chord identification using pitch class profile method with fast fourier transform feature extraction. Int. J. Comput. Sci. Issues (IJCSI) **11**(3), 139 (2014)
13. Osmalsky, J., Embrechts, J.J., Van Droogenbroeck, M., Pierard, S.: Neural networks for musical chords recognition. In: Journees d'informatique Musicale, pp. 39–46 (2012)
14. Quenneville, D.: Automatic Music Transcription. Ph.D. thesis, Middlebury College (2018)
15. Rajparkur, P., Girardeau, B., Migimatsu, T.: A supervised approach to musical chord recognition (2015)
16. Spotify, 6 Apr 2019. https://insights.spotify.com/us/2015/05/06/most-popular-keys-on-spotify/

17. Wats, N., Patra, S.: Automatic music transcription using accelerated multiplicative update for non-negative spectrogram factorization. In: 2017 International Conference on Intelligent Computing and Control (I2C2), pp. 1–5. IEEE (2017)
18. Zhou, X., Lerch, A.: Chord detection using deep learning. In: Proceedings of the 16th ISMIR Conference, vol. 53 (2015)