Music Analysis and Synthesis

Kaustubh R Kulkarni[1], Sowmiya Raksha R Naik[2]

[1]Adhoc Faculty Member, Department of Computer Engineering and Information Technology, Veermata Jijabai Technological Institute, Mumbai, India.
[2]Assistant Professor, Department of Computer Engineering and Information Technology, Veermata Jijabai Technological Institute, Mumbai, India.

Abstract - *This paper focuses on some of the approaches for music analysis and synthesis. The first step used for music analysis here is separating the sources and recognizing them. The next step is the estimation of perceptual attributes of music namely pitch, tempo and recognition of notes. Finally using the estimated pitch, tempo and notes recognized, a new audio file is synthesized. This audio file is combined with one or more of the original source audio file(s).*

***Key Words*: Neural network; Machine learning; Deep learning; Music; Music analysis; Music source separation and recognition, Music synthesis.**

1. INTRODUCTION

The audio files have some attributes that are perceptual. These attributes are very important in composing a new piece of music. Such attributes are pitch, tempo and notes. Since attributes in question are perceptual there are many challenges in estimating them.

Deep learning attempts to emulate human sensory organs. Human vision has been especially modeled very well by deep learning approaches. This paper works on emulating the human auditory perception. Humans have an understanding for good and bad music, which is, the music that goes or doesn't go very well with the vocals and is or is not melodious. The motivation is to build a system that can listen to a vocal or instrumental audio and can come up with an audio file of its own that will go well with the former.

Also the music that we hear in our daily life is a complex mixture of sounds from multiple superimposed sound sources each of which has their own perceptual attributes. The challenge in music source separation is that although human ear can filter the lyrics of the vocals or the notes of an instrument very well, the computing system is confused by the overlapping of different sources and has to be trained extensively to tell the sounds of different sources apart. Perceptual pitch estimation has the challenge that the frequency with the largest magnitude can be very different from the perceived pitch.

The proposed system analyses an input audio file by first separating it into four audio files corresponding to the sources that contributed to the input audio file: first is the vocals, second is the bass or other string instruments, third is the drum and other percussive instruments and fourth includes all remaining categories of instruments like wind instruments. The audio files of separated sources are further analyzed to estimate their perceptual attributes like pitch, tempo and notes. Then these attributes are used to generate a new audio file. This new audio file is then used to substitute one of the original sources in the input audio file.

The contribution of this paper is that it combines multiple music analysis techniques along with music source separation. Also the idea of generating a new audio file based on all of the perceptual attributes of the sources like pitch, tempo and note to substitute one of the individual sources of a music piece is also unique.

Section 2 reviews the previous works related to the paper. The music analysis and synthesis approaches used in the present work are documented in section 3. Finally, section 4 concludes the paper and outlines the possibilities for future work.

2. REVIEW OF PREVIOUS WORK

2.1 Music Source Separation and Recognition

The music signal has been represented using a matrix by Serrano et al [1] for instrument extraction with the help of non-negative matrix factorization (NMF). Serrano et al [1] have modified the factorization to consider all possible shifts in the pitch due to vibrato.

One of the variants of non-negative matrix factorization is non-negative matrix partial co-factorization (NMPCF), which has been used by Hu and Liu. [2] The non-vocal portions of signal consist of accompaniment component only and have been used in co-factorization as a part of prior knowledge. Hu and Liu [2] have also used a spectrogram of pure accompaniment and a spectrogram of clean singing voice as prior knowledge.

Neural networks have been used by Masood et al [3] for instrument recognition. The characteristics of music like dynamics, timbre, tonality, rhythm have been used to extract features from music. A sequence of frames called a context window [4] represented in the form of feature vectors have been given as an input to the neural network for instrument recognition.[3]

Sharma [4] has tried various neural network architectures such as multilayer perceptron (MLP), stacked MLP, convolutional neural network – hidden Markov Model (CNN-HMM), restricted Boltzmann machine (RBM), stacked auto encoder and deep belief network for extracting vocals from songs.

Uhlich et al [5] have applied short time Fourier transform (STFT) on context window for feature vector generation. This feature vector has been fed to a deep neural network (DNN) with rectified linear unit (ReLU) layers. The phase information from input features has then been combined with the magnitude information from DNN outputs to estimate the STFT of the target instrument. Finally the signal has been converted back to the time domain using an inverse STFT. [5]

2.2 Pitch Estimation

"The pitch of a musical instrument note is primarily determined by its fundamental frequency of oscillation as perceived by a human". [6]
Singh and Kumar [6] have used the following pitch detection algorithms:

Autocorrelation function (ACF) is defined as [6]:

$$A_c(\tau) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)x(n+\tau) \quad (1)$$

where $x(n)$ is the input signal and τ is the lag (delay) value $\tau = 0, \pm 1, \dots$. The peaks in $A_c(\hat{o})$ have been used to estimate the pitch period and hence the pitch.

Average Magnitude Difference Function (AMDF), is defined as [6]:

$$x(m) = \frac{1}{N-m-1}\sum_{n=0}^{N-m-1} s(n+m) - s(n) \quad (2)$$

where $s(n)$ is the input signal and $0 \le m \le N$. The values of m for which $x(m)$ becomes minimum has been the pitch period.

Cepstrum of a signal is defined as [6]:

$$c[n] = F^{-1}\{\log F\{s(n)\}\} \quad (3)$$

where F is the DFT and F^{-1} is the IDFT. The frequency domain in spectrum is called quefrequency domain in the cepstrum. Peaks have been searched in the cepstrum and further procedure used to find the fundamental frequency has been the same as the autocorrelation method.

2.3 Tempo Estimation

"Tempo is indicated by beats per minute (BPM) or is expressed as tempo annotation in words such as 'fast'". [7]

Note onsets have been used for tempo estimation by methods like measuring inter-onset interval, calculating ACF of onset detection function (ODF), calculating DFT of ODF or a combination/variation of these methods. Note onsets are detected by abrupt increase in spectral amplitude of the signal. Note onsets have been used for tempo estimation by methods like measuring inter-onset interval, calculating ACF of onset detection function (ODF), calculating DFT of ODF or a combination/variation of these methods. Note onsets are detected by abrupt increase in spectral amplitude of the signal. The spectral amplitude is defined as 'the sum of the spectral bins at each instant in time' [8]:

$$SA[n] = \sum_{k=0}^{\frac{N}{2}-1} |X[n,k]| \quad (4)$$

To detect abrupt changes in the spectral amplitude, a bi-phase filter $h[n]$ is used. Detecting abrupt changes in spectral amplitude is used to detect note onsets [8]:

$$ODF[n] = SA[n] * h[n] \quad (5)$$

3. DESCRIPTION OF PRESENT WORK

3.1 Music source separation and recognition

The approach used for source separation is based on Uhlich et al [5] but using convolutional neural network (CNN). The input audio signal is processed by computing its Short-Time Fourier Transform (STFT) first. The STFT spectrogram consists of magnitude spectrogram and phase spectrogram. The magnitude spectrogram becomes the input of a convolutional neural network (CNN). The CNN outputs an estimate which is used to mask out sources by computing a time-frequency mask for each source.

The masks are applied to the input audio's magnitude spectrogram to estimate the magnitude spectrogram of the respective source. These estimated magnitude spectrograms are combined with the phase spectrogram of the input audio signal. Finally applying inverse STFT on each source's combined spectrogram, the audio signals corresponding to the separated sources are obtained.

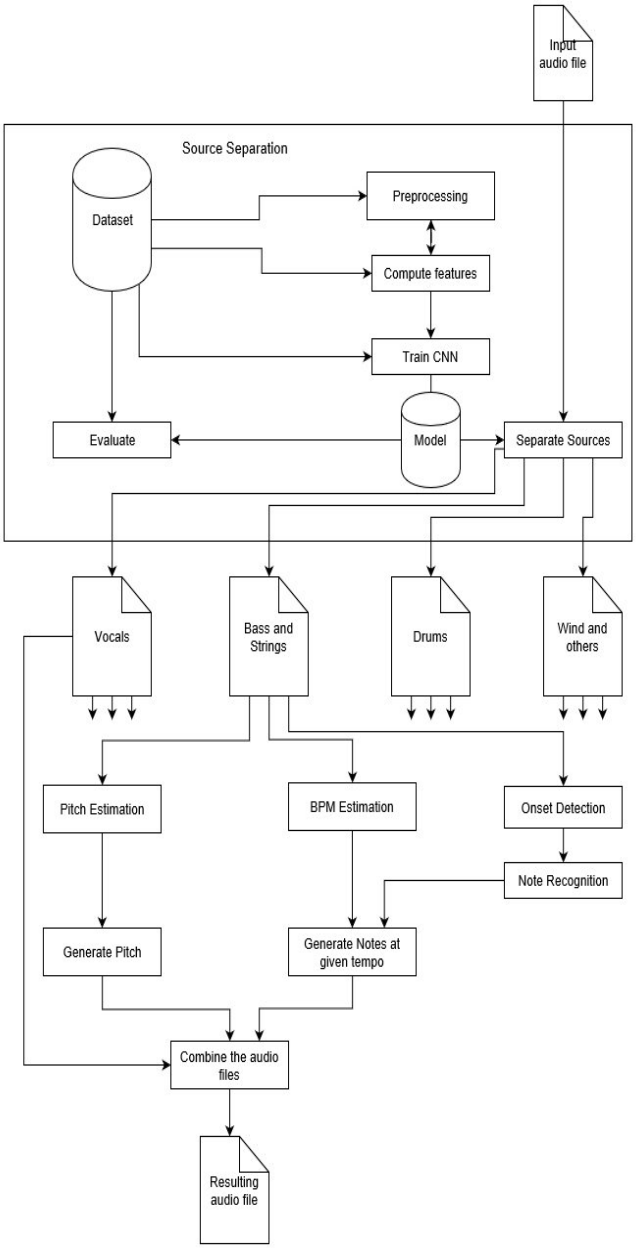

Fig – 1: Proposed System

The first convolution layer is responsible to train the model to learn timbral features. The second convolution layer models temporal information for the instruments that may be identified from the features learned in the previous convolution layer.

The output of the second convolution layer is connected to a fully connected Rectified Linear Unit (ReLU) layer. This layer achieves dimensionality reduction to reduce the total parameters of the network and it is also called as a bottleneck layer. This layer combines the features learned from the previous layers, with a ReLU. The output of the first fully connected layer is passed to another fully connected layer, with a ReLU and the same size as the output of the second convolution layer.

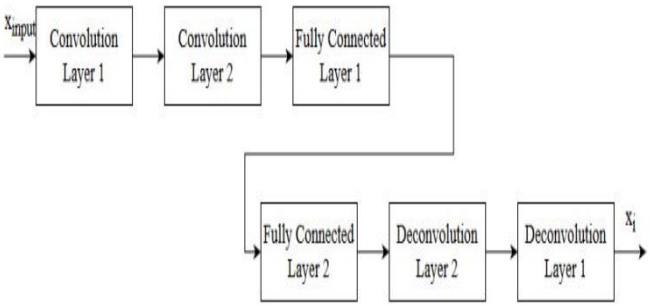

Fig – 2: Architecture of CNN used in music source separation and recognition

Thereafter, this layer is reshaped to the same dimensions as the second convolution layer and passed through successive deconvolution layers. The CNN outputs an estimate x_i for each of the sources i.

From the output of the network x_i, a mask m_i for the source i is computed, as follows:

$$m_i(f) = \frac{|x_i(f)|}{\sum_{i=1}^{N}|x_i(f)|} \quad (6)$$

where N is the total number of sources to be estimated. The estimated mask is then applied to the input mixture signal $x_{input}(f)$ to estimate the sources $\tilde{x}_i$.

$$\tilde{x}_i(f) = m_i(f)x_{input}(f) \quad (7)$$

The algorithm used for training the neural network parameters is stochastic gradient descent with AdaDelta optimization.

The dataset is first processed to be suitable for feature extraction. Then a model is produced as a result of training. The model is then used to separate an input audio file into a voice audio file, bass and other string instrumental audio file, drums instrumental audio file and wind and other instruments' instrumental audio file.

The separated audio files are analysed independently in three different processes. The first process used is that the audio files are analysed to estimate the pitch values present in them at different time windows and finally to compute the approximate pitch value for its overall duration.

The second process is to analyse the tempo of the audio file again at different time windows and again for the computing the approximate tempo of the overall audio file. The third process is to find note onsets in the audio file and then use the pitch value at that point in time to recognize the note present there in the audio file.

3.2 Pitch Estimation

The cepstrum has a peak corresponding to the high frequency source. Squaring the cepstrum makes the peak more prominent and easier to detect.

The audio file whose pitch is to be detected restricted to .wav format is opened using wave library in Python. First the FFT of the signal is calculated. Then a Blackman window is applied to the result of FFT. It is used for smoothening discontinuities at the terminals of the sampled signal. The windowed FFT values are then squared. The bin with the highest values is found. A quadratic interpolation is applied around the peak. Finally using the logarithm of the peak value and its two neighboring values, the pitch is estimated.

3.3 Tempo estimation

The number of frames and frame rate of the audio file are determined. The entire file is read and stored in the form of an array. It is then divided into chunks called windows. The tempo, in terms of beats per minute, is first calculated individually for each of these windows. The final tempo is computed as the median of the tempos of all the windows.

For each of the individual windows, first a discrete wavelet transform (DWT) is applied. Then they are filtered along one-dimension with a discrete IIR (infinite impulse response) filter. Finally autocorrelation function (ACF) is used to highlight the periodicity in the filtered signal; peaks are detected and used to find the tempo.

3.4 Note recognition

The wav file is loaded and resampled to 22.05 KHz. The hop size used is of 512 frames, which at 22.05 KHz, is approximately 23ms. First the frame wise beat strength profile, also called onset envelope, is obtained. As a normalization step to make the threshold more consistent, the onset envelope is shifted appropriately to be non-negative. Peaks are picked from the onset envelope using threshold parameters.

Then the notes present in the music clip are recognized using the knowledge of note onset time and pitch value at that point in time. The note with a pitch value of 16.35 Hz has been labeled as C0. For a note with a pitch value P Hz, the number of half steps from C0 to P is:

$$h = 12\log_2\left(\frac{P}{C0}\right) \quad (8)$$

The sequence of note labels that are assigned at every step starting from C is C , C♯, D, D♯, E, F, F♯, G,G♯,A,A♯ and B.

3.5 Music Generation

The pitch values are used to generate a new audio file, similarly a new audio file is generated using the tempo values and the recognized notes. These two audio files are combined along with one or more of the originally separated audio files to produce a new audio file that is similar to the input audio file except that one or more of the instruments in it have been replaced by other instrument/s with similar, not necessarily identical, pitch, tempo and note values.

4. CONCLUSIONS

This paper has reviewed some of the approaches that can be used for musical source separation and recognition, tempo and pitch estimation and recognition of musical notes. The performance of source separation and recognition can be by using a deep learning approach. The pitch, note and tempo can be estimated accurately and this information can be used to generate a new audio file. This audio file can then be combined with the existing audio file but with different source than the former audio file.

ACKNOWLEDGEMENT

We thank Prof. Dr. V B Nikam, Head of Department of Computer Engineering and Information Technology, Veermata Jijabai Technological Institute, Mumbai who provided valuable insights and expertise that greatly assisted the research.

REFERENCES

1. F. J. Rodriguez-Serrano, S. Ewert, P. Vera-Candeas and M. Sandler, "A score-informed shift-invariant extension of complex matrix factorization for improving the separation of overlapped partials in music recordings," 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, 2016, pp. 61-65.

2. Y. Hu and G. Liu, "Separation of Singing Voice Using Nonnegative Matrix Partial Co-Factorization for Singer Identification," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 23, no. 4, pp. 643-653, April 2015.

3. S. Masood, S. Gupta and S. Khan, "Novel approach for musical instrument identification using neural network," 2015 Annual IEEE India Conference (INDICON), New Delhi, 2015, pp. 1-5.

4. V. Sharma, "A Deep Neural Network based approach for vocal extraction from songs", 2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), Kuala Lumpur, 2015, pp. 116-121.

5. S. Uhlich, F. Giron and Y. Mitsufuji, "Deep neural network based instrument extraction from music," 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, 2015, pp. 2135-2139.

6. C. P. Singh and T. K. Kumar, "Efficient pitch detection algorithms for pitched musical instrument sounds: A comparative performance evaluation," 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), New Delhi, 2014, pp. 1876-1880.

7. F. H. F. Wu, "Musical tempo octave error reducing based on the statistics of tempogram," 2015 23rd Mediterranean Conference on Control and Automation (MED), Torremolinos, 2015, pp. 993-998.

8. T. P. Vinutha, S. Sankagiri and P. Rao, "Reliable tempo detection for structural segmentation in sarod concerts," 2016 Twenty Second National Conference on Communication *(NCC)*, Guwahati, 2016, pp. 1-6.