Automatic music transcription based on convolutional neural network, constant Q transform and MFCC

Zhihang Meng¹, Wencheng Chen²

¹School of Automation, Beijing University of Posts and Telecommunications, Beijing ,100876, China
²School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing ,100876, China

¹mengzhihang@bupt.edu.cn ²e-mail: mzh1999@163.com

Abstract. Automatic music transcription is a function that relies on the efficiency of the computer's own calculation speed to recognize the pitch of external input audio and output it accurately. In the past methods, there are often problems such as inability to reflect music continuity or narrow identifiable frequency range. In order to avoid such problems, two methods different from the previous Mel Cepstral Coefficients and Constant Q Transform are selected to extract the features of music, and convolutional neural networks are used for training and recognition. Among them, the Mel Cepstrum coefficient is used to judge the timbre, and the constant Q transform is used to judge the pitch. After inputting the corresponding features into the neural network for training and learning, we can find that the recognition success rate has reached 95%.

1. Introduction

Automatic music transcription recognizes the pitch of external input audio and outputs the pitch in an accurate manner based on the high efficiency of computer operation [1]. It proves a valuable tool to professional players in terms of musical instrument tuning, music database retrieval and automatic music reading. At present, automatic music transcription is achieved mainly by applying NMF (non-negative matrix factorization) [2] in spectrogram or PLCA (probabilistic latent component analysis) [3] to estimate pitch. Despite accuracy in image recognition, the two approaches could hardly maintain continuity in music. The approaches could accurately recognize individual note, but this is not the case for continuous music. There are literatures [4] using BP neural network and Softmax regression model for this purpose. The method has greater accuracy though considerations are not given to the timbre of different musical instruments. There are also literatures [5] estimating the pitch of notes via the ratio of subharmonics to harmonics (SHR), which has proved applicable to continuous notes in a narrow range of frequency but unsuitable for automatic music transcription. There are literatures [6] utilizing short-time autocorrelation to detect pitches within one octave in a narrow range.

Targeting deficiencies of the abovementioned model, this paper adopted Mel cepstrum coefficient (MFCC) and constant Q transform (CQT) to extract the features of notes to ensure accuracy and practicality. The MFCC determines the category of musical instrument based on timbre, while constant Q transform derives the spectrogram, which will be identified by convolutional neural network.
2. Principles

2.1. Feature extraction of musical instrument signals

For the feature extraction of audio signals, we must first distinguish features of different musical instruments in a consistent manner. In addition, the extracted features should reflect the features of audio signals as comprehensively and accurately as possible.

It is known that music is different from pure sound. Whether it’s “notes of the ancient pentatonic scale” or the twelve-tone equal temperament in the modern world, there is a fixed standard for the pitch of music, which is the note we are familiar with. The pitches perceived by human ears is the frequency ratio of two tones. The simpler the ratio, the more harmonious the two tones sound to human ears. Concepts such as “interval”, “whole tone” and “half tone” in music principles are all of specific frequency-ratio in essence. That is to say, only the sounds of specific frequency-ratio have the potential of being wonderful music. Otherwise, it will sound like noise. At present, it is generally believed that the frequency ratio of 3:2 between two successive sounds features the maximized harmony. To keep the scales within the same octave, the frequency of the next scale should decrease by half every other scale. In order to satisfy this special frequency ratio, the frequency ratio of pairwise scales shall not be linear within one octave. The continuity of music has imposed challenges to the accurate extraction of sound features.

There are two kinds of transformation of audio signals, including frequency domain and inverse frequency domain. The former consists of Short-time Fourier Transform (STFT)\[^7\], constant Q transform (CQT), etc. The inverse frequency domain includes Mel-Frequency Cepstral Coefficient (MFCC). As an important low-frequency domain feature in music signals, the STFT is commonly used to create spectrograms despite its defects. The frequency of discrete Fourier transform is linearly spaced, but the frequency of notes doubles for every increase of octave (every 12 notes). This leads to some defects in distinguishing notes. CQT is a tool of time-frequency analysis just as important as the STFT, whose window length varies with frequency. This paper selected constant frequency-to-bandwidth ratio in which each note has four frequency points. This works well for convolution, because now the distance between the first harmonic and the second harmonic or the distance between the second harmonic and the third harmonic is the same for all notes, regardless of the fundamental frequency. The MFCC is extracted according to the perceptual characteristics of human ears and plays an important role in timbre recognition. Therefore, CQT and MFCC were selected as extraction features in this paper.

2.1.1. Mel-Frequency Cepstral Coefficient

Mel-Frequency Cepstral Coefficients (MFCC) \[^8\text{-}9\] is a feature extraction approach based on the nonlinear characteristics of human hearing. The resemblance between the extracted features and human hearing make it an ideal tool to represent the features of music signals. The relationship between MFCC and frequency is given by Formula (1):

$$Mel(f) = 1125 \times \left(1 + \frac{f}{700}\right)$$

where \(f\) represents the perceived frequency in the unit of Mel, also called the Mel-scale. Its relationship with linear frequency is shown in Figure 1:

![Figure 1. The relationship between mel cepstrum coefficient and frequency](image-url)
To extract MFCC, the signal needs to be segmented and pre-emphasized to highlight the high-frequency part of the signal. The formula is as follows:

$$s_i(n) = si(n) - \beta \ast s_i(n-1)$$  \hspace{1cm} (2)

Then, the energy spectrum is obtained by discrete Fourier transform using the following formula:

$$S_i(n) = \sum_{n=1}^{N} s_i(n) \ast h(n) e^{-j2\pi kn/N}, 1 \leq k \leq K$$  \hspace{1cm} (3)

where K represents the length of discrete Fourier transform, h(n) represents the analysis window with the sampling length of N. The hamming window was selected in this paper. Windowing would smooth the signal and reduce the sidelobe size and spectrum leakage after Fourier transform. The hamming window function is as follows:

$$w(n, \alpha) = (1 - \alpha) - \alpha \cos\left(\frac{2\pi n}{N-1}\right), \hspace{1cm} 0 \leq n \leq N - 1$$  \hspace{1cm} (4)

In the above formula, \(\alpha\) is the coefficient (usually \(\alpha=0.46\)), and the periodic energy spectrum of each frame is estimated as follows:

$$P_i(k) = \frac{1}{N}|S_i(k)|^2$$  \hspace{1cm} (5)

Then, determine the upper and lower limits of signal frequency using Formula (6):

$$M^{-1} = 700 \exp\left(\frac{m}{1125}\right) - 1$$  \hspace{1cm} (6)

The selected value expressed by Mel scale is converted to the frequency domain, thus the corresponding position of the filter is determined. Then take the logarithm of the vector after filtering, and perform discrete cosine transform (DCT):

$$C(n) = \sum_{n=0}^{\frac{M}{2}} s(m) \cos\left(\frac{\pi n (m-0.5)}{M}\right), n = 1, 2, \ldots, L$$  \hspace{1cm} (7)

From which the MFCC is obtained.

The study of auditory mechanism of human ears found that when two sounds with different loudness act on human ears, the frequency with higher loudness will affect the perception of frequency components with lower loudness, making them imperceptible. This phenomenon is called masking effect. The masking effect makes it easy for bass to mask treble, while the other way around would be extremely difficult. That is to say, for chords that emit sounds simultaneously, the highest tone on the top layer is easily recognized while the two sounds at a lower layer can hardly be identified. This would affect the accuracy of recognition. Therefore, the introduction of MFCC could help establish a set of band-pass filters based on density of the critical bandwidth in the frequency band from low frequency to high frequency to filter the input signal. After obtaining the logarithmic energy of the filter output, DCT is performed to obtain MFCC as shown in Figure 2.

Figure 2. Mel cepstrum coefficient extraction process

Because this feature does not depend on the nature of the signal, it does not make any assumptions and restrictions on the input signal while utilizing the auditory model of the human ear. Therefore, compared with linear predictive cepstrum coefficient (LPCC) based on vocal tract model, this parameter has better robustness and adaptability to auditory characteristics of human ears, thus having a better performance in audio recognition.
2.1.2. Constant Q Transformation

Constant Q transform \[^{[10-11]}\] is an important tool of time-frequency analysis like STFT, especially suitable for music signal analysis. This is because in music, sounds are distributed exponentially. However, the spectrum obtained by Fourier transform is distributed linearly, leading to errors in the estimated values of some musical scales. Constant Q transform fixes this problem as the generated spectrum has a frequency axis of log scale instead of linear scale, and the window length changes with frequency. This can effectively improve the resolution accuracy in the low frequency region. The frequency component of the Kth semitone in the Nth frame of CQT can be expressed as:

\[
X_n^{\text{cqt}}(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(m)w_{N_q}(m)e^{-j2\pi nQ/N_i}
\]

where \(Q\) is a constant only related to \(\beta\), which is the number of spectral lines of one octave within one octave:

\[
Q = \frac{1}{2^{\lfloor \beta \rfloor} - 1}
\]

The piano spectrogram obtained by CQT is shown in Figure 3:

![Figure 3. Piano scale spectrogram](image)

2.2. Recognition network

2.2.1. Convolutional Neural Network

Convolutional Neural Network (CNN) \[^{[12-15]}\] is a kind of feedforward neural network. Its artificial neurons can respond to part of the surrounding units in the coverage, with an excellent performance for large-scale image processing. Convolutional neural network consists of three parts: input layer, hidden layer and full connection layer, in which the hidden layer includes a convolutional layer and a pooling layer. The convolution neural network boasts features such as displacement, scale and nonlinear deformation stability via combinations of convolution layer and pooling layer \[^{[16-17]}\].

The convolution layer is composed of multiple convolution kernels, which extract image features through discrete convolution calculations between matrices using Formula (10):

\[
\text{Cov}(x, y) = \sum_{i=0}^{k} \sum_{s=0}^{k} F(s, t) \times G(x-s, y-t)
\]

where \(s\) and \(t\) represent the width of convolution kernel in \(x\) and \(y\) axis while \(k\) represents the scale of convolution kernel.

The essence of pooling layer is dimension reduction, through which the parameters and calculation load in convolutional neural network are reduced while the over-fitting of networks can be suppressed. In convolutional neural networks, maximum pooling layer is often used for pooling to obtain the maximum value in local area. The formula is as follows:
\[
P = \max_{i \in \mathcal{O}} \{A^{i, l}\}
\]  

(11)

After being processed by convolution layer and pooling layer, the obtained feature map needs to be paved by full connection layer before being converted into one-dimensional vector input and then output by calculation. The formula is as follows:

\[
z_{j}^{l+1} = \sum_{i=1}^{n} W_{ij} a_{i} + b_{j}
\]  

(12)

An activation function needs to be set between the convolution layer and the pooling layer, which optimizes the parameters generated in the network, thus effectively improving the recognition capacity of the network. The commonly used activation function includes:

\[
sigmoid = \frac{1}{1 + e^{-x}}
\]  

(13)

\[
tanh = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}
\]  

(14)

\[
ReLU = \max(0, x)
\]  

(15)

In this paper, the spectrogram obtained by CQT and MFCC were used as training data of Convolutional Neural Network (CNN), with the frame window of the spectrogram serving as input to identify pitches in the central frame. Meanwhile, a threshold was set for the network to distinguish the simultaneous occurrence of multiple notes. The network would output a score, which, if higher than the threshold, means that a note is active in a given central frame.

### 3. Procedures

First, the convolution neural network (CNN) was written by pytorch, and the existing 2000 piano monophonic audio data was down-sampled and converted into mono by Librosa library. Then, CQT was used to obtain the spectrogram, which was classified into training data and test data.

A total of 2000 monophonic audios produced by two different instruments (piano and flute) were processed in the same way, and MFCC was used to extract their features. There was a total of 1500 training data for each of the two musical instruments. The network was trained to distinguish the timbre of the musical instruments. Then, the spectrogram of different notes produced by one single instrument was used for training so that the network could recognize notes with different pitches produced by one single instrument. The input data was image of 60*60, with 16 convolution kernels of 5*5 at the first convolution layer to ensure that the image size would not change due to convolution. Sixteen feature maps of 60*60 were input into the pooling layer, which pooled in 2*2 scale with a step size of 2. A total of sixteen 30*30 feature maps were obtained after pooling. The second convolution layer received output from the previous layer, and used 32 convolution kernels of 5*5 to pool again on the same pooling layer. In this way, a total of 32 feature maps of 15*15 were obtained. On this basis, the feature maps were paved by the full connection layer, and the results were output and classified. In the network, the Rectified Linear Unit (ReLU) was used as the activation function as it was not likely to saturate and boasted better effect. Forward propagation was used for training at a learning rate of 0.001. The overall process flow is shown in Figure 4.

![Figure 4. Implementation process](image-url)
4. Experiment and Result

After training the network according to the processing steps, 500 test data were input into the network for verification. The experiment adopted two types of feature inputs, including single feature MFCC and CQT. It tested the capacity of the network in terms of timbre recognition and note identification under Windows environment.

4.1. Influence of different network parameters on experimental results

In the experiment, we randomly divided the audio datasets several times to generate different training sets and test sets to prevent the convolution neural network from over fitting due to limitations of training samples, and trained the network on different servers to verify the training results. Most of the results were accurate and consistent with the results of recognition. However, the training of one dataset showed a big error in the result, which indicating the possibility of over-fitting. To better adapt the network and avoid over-fitting, this paper adjusted the super-parameters and learning rate of the network, which though lowered the training speed, fixed the problem of over-fitting. Meanwhile, we also tried to increase the number of network layers and used the depth of extracted features to avoid over-fitting. However, under the same result, increasing network layers proved time consuming, for which it was not adopted. A comparison of the effects of different parameters are shown in Table 1.

| Learning rate | Network layers | Training rate | Recognition rate | Overfitting |
|---------------|----------------|---------------|------------------|-------------|
| 0.001         | 2              | normal        | 0.9472           | Yes         |
| 0.0005        | 2              | Slow          | 0.9374           | No          |
| 0.001         | 3              | Very slow     | 0.9361           | No          |

4.2. Comparative experiment with other methods

To make a more effective and intuitive comparison with other methods, several methods commonly used in extracting audio features were included in comparative experiments. The results showed that these methods had problems of accurately identifying continuous music and narrow recognition range. Compared with the method in this paper, there was a certain gap in recognition accuracy. The experimental results are shown in Table 2.

| Feature | Recognition rate of Monophonic | Recognition rate of whole audio | Recognition range |
|---------|-------------------------------|--------------------------------|-------------------|
| NMF     | 0.9172                        | 0.7124                         | Normal            |
| PLCA    | 0.9254                        | 0.7037                         | Normal            |
| Softmax | 0.9473                        | 0.9356                         | Normal            |
| SHR     | 0.9214                        | 0.9203                         | Narrow            |
| STA     | 0.9145                        | 0.9027                         | Very narrow       |

4.3. Experimental results of feature extraction using CQT and MFCC

In this experiment, four evaluation criterions were used to evaluate and analyze the experimental results, including accuracy, precision, recall and F-measure (or F-Score). Under the training conditions adopted by the procedure, the results of each criteria are shown in Table 3.
From Table 3, it is noted that MFCC and CQT features alone could better identify timbre and notes. Given the randomness of the network, in order to make the experimental results more convincing, many comparative experiments were carried out under the same training model with different number of training samples, and the results are shown in Table 4.

| Feature | Accuracy | Precision | Recall | F-Score |
|---------|----------|-----------|--------|---------|
| CQT     | 0.9472   | 0.9497    | 0.9472 | 0.9484  |
| MFCC    | 0.9394   | 0.9413    | 0.9394 | 0.9403  |

| Number of inputs | CQT | MFCC |
|------------------|-----|------|
| 1000             | 0.9057 | 0.8977 |
| 1500             | 0.9283 | 0.9219 |
| 2000             | 0.9476 | 0.9371 |
| 2500             | 0.9468 | 0.9367 |
| 3000             | 0.9472 | 0.9394 |

From Table 4, it is found that the recognition rate has an upward trend with the increase of samples. However, when the number of samples reaches more than 2000, the recognition rate tends to be stabilized without significant growth.

5. Conclusion
This paper used the output audio files of existing music scores as the training data for neural network. For the test data, MFCC and CQT were used to extract features, which were input into CNN for recognition, featuring a higher recognition rate. Among them, MFCC was mainly used as the identification feature of timbres produced by different musical instruments, and classified audios at the beginning. CQT was used to derive the spectrogram. As a feature of note recognition, CNN was used to process the spectrogram and output the result because of its excellent performance in image recognition. In future studies, we should try different convolution kernel parameters in evaluating feature contributions to time domain and frequency domain, while incorporating a variety of musical instruments to test the accuracy of the network. Meanwhile, we need to collect audio data from external musical instruments for experiment and verification.

References
[1] Jonathan Sleep. (2017) Automatic music transcription with convolutional neural networks using intuitive filter shapes. https://doi.org/10.15368/theses.2017.95.
[2] Zhang, F., Li, L. (2019) Semi-supervised Learning Approach Based on Non-negative Matrix Factorization and Harmonic Functions. Statistics & Decision., 22:71-73.
[3] Siddharth, S., Emmanouil, B., Simon, D. (2015) An end-to-end neural network for polyphonic piano music transcription. https://arxiv.org/abs/1508.01774.
[4] Chen, Y.W., Li, K., Han, Y., Wang, Y.P. (2019) Musical Note Recognition of Musical Instruments Based on MFCC and Constant Q Transform. http://kns.cnki.net/kcms/detail/50.1075.TP.20191122.1411.008.html
[5] Xu, P.J., Guo, L., Liu, S.C. (2011) Note recognition combining pitch-detection and onset-detection. Journal of Computer Applications., 31(s2):172-175.
[6] Wu, J.J., Meng, L.L. (2009) The recognition of the musical pitch. Electronic Measurement Technology., 32(4):126-128.
[7] Liu, Y.P., Wang, J.W., Pei, S.T., Ji, X.X., Liu, Y.J. (2019) Method for identifying and classifying partial discharge based on short time Fourier transform and sparse representation. Electrical Measurement & Instrumentation., 56(23):31-36.

[8] Lin, L., Wang, R.D., Yan, D.Q., Li, C. (2018) A playback speech detection algorithm based on log inverse Mel-frequency spectral coefficient. Telecommunications Science., 5:91-98.

[9] Song, Z.Y. (2013) Application of MATLAB in Speech Signal Analysis and Synthesis. Beihang University Press, Beijing.

[10] Brown, J.C. (1998) Calculation of a constant Q spectral transform. Journal of the Acoustical Society of America., 89(1):425-434.

[11] Ding, Z.Z., Dai, L.R. (2015) A study of constant Q transform in music signal analysis. Acoustic technology., 24(4):259-263.

[12] Li, Y.D., Hao, Z.B., Lei, H. (2016) Survey of convolutional neural network. Journal of Computer Applications., 36(9):2508-2515.

[13] Zhou, F.Y., Jin, L.P., Dong, J. (2017) Review of Convolutional Neural Network. Chinese Journal of Computers., 40(6):1229-1251.

[14] He, K., Zhang, X., Ren, S. (2016) Deep residual learning for imagerecognition. In: Proc-eedings of the IEEE conference on computer vision and pattern recognition. Las Vegas. pp. 770–778.

[15] Jin, L.J., Zhan, J.M., Chen, J.H., Wang, T. (2020) Drill pipe fault diagnosis method based on on e-dimensional convolutional neural network. http://kns.cnki.net/kcms/detail/33.1245.T.20200403.1019.008.html

[16] Krizhevsky, A., Sutskeyer, I., Hinton, G.E. (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. Lake Tahoe. pp. 1097–1105.

[17] Bouvrie, J. (2016) Notes on convolutional neural networks. http://cogprints.org/5869/1/cnn_tutorial.pdf.