Audio Key Identification using Neural Networks on Pitch Profile

Thitaree Tanprasert (Mint)

Harvey Mudd College

CS 152, Spring 2017

Introduction

Audio key identification is a problem widely explored in Music Information Retrieval field of research. With a polyphonic audio file as an input, I would like to determine the tonal pitch of the piece. Using signal transformation and filtering techniques, we can represent an audio file, mathematically. Therefore, we can view this problem as a classification problem of data in high-dimensional space, which may be solved with neural network models.

A representation of audio, which are frequently adopted in key identification research is called pitch profile, also known as chroma features. [1] [9] Pitch profile is a vector, containing the energy of each pitch class in a segment of audio. In this project, I choose to model a song as pitch profile, since it greatly reduces the dimension of the data and is convenient to use as input for neural network models.

Related Work

- **Pitch Profile**

  Pitch profile is widely used for audio key identification research. Researches by Zhu and Kankanhalli in 2005 and 2006 presents the method to create a pitch profile that represents audio, precisely, as well as a tone clustering algorithm for key determination. [9] [10] I adapt some of the methods in this project, as will be described in the next section. Pitch profile is also included in "Million Song Dataset", which is made public by LabROSA, Columbia. This labeled dataset contains pitch profile, as well as other musical characteristics. [1]

- **Other Neural Nets Approaches**

  In 2011, Sander Dieleman, Philemon Brakel and Benjamin Schrauwen develop a convolutional neural network model, which does 3 tasks: artist identification, genre classification, and key identification. The research was conducted on the Million Song Dataset. The highest accuracy obtained for key identification is 86.53%. The convolutional MLP model consists of 6 layers, including windowed logistic regression layer. It uses unsupervised pretraining and supervised fine-tuning. [1]

Method

1. Creating pitch profile

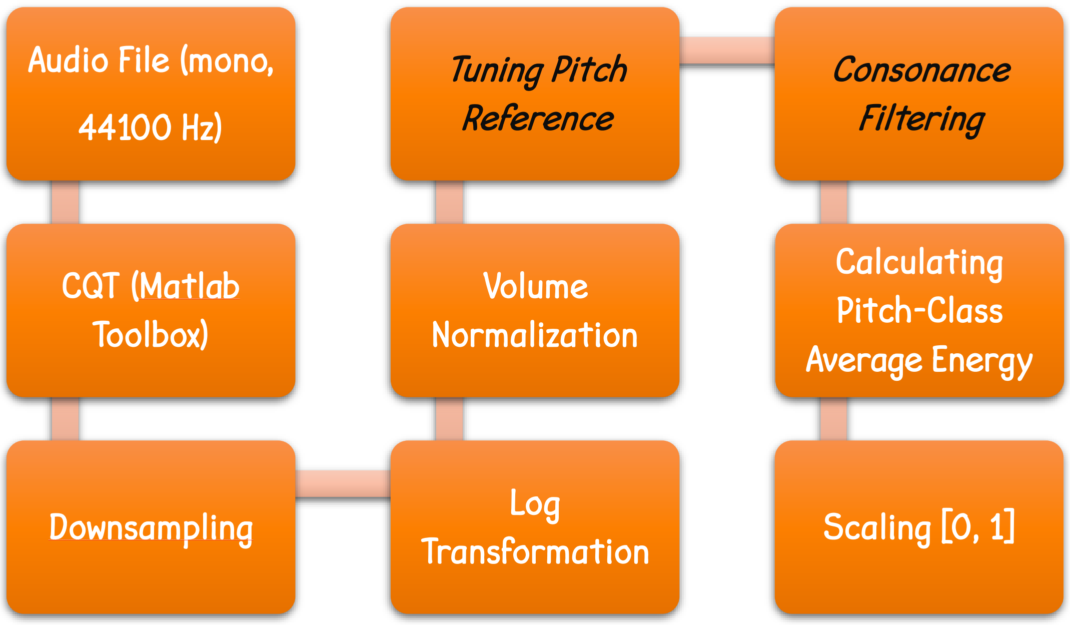

Figure 1: The steps for creating a pitch profile from an audio file.

a. Constant Q Transform

The audio file is first converted from stereo channels to mono, with the sampling rate 44100 Hz. It then goes through constant Q transform, which is done by a CQT toolbox in Matlab (2013 version). Here, we obtain a large matrix, in which each row corresponds to an audio frequency. The frequencies range 5 octaves from C3 (130.81 Hz) to C8 (4186.01 Hz) and are divided into bins. [7] For instance, 12 bins per octave means that each bin approximately correspond to a pitch in 12-equal temperament tuning system, and consecutive rows correspond to frequencies that are 100 cents apart. Each column corresponds to a 12.4 millisecond-long time frame of the audio. The cell at row *i* and column *j* of the matrix contains the "energy" or amplitude of frequency bin *i* in time frame *j*.

b. Downsampling and Log Transformation

Since the matrix obtained from CQT is very large and contain details which are not necessary in our case, I average the energy over 5 time frames to reduce the number of columns of the matrix. Moreover, I convert the amplitude of the audio into dB units, which corresponds more directly to how human's ears perceive audio. The code for this part is taken from [8].

c. Volume Normalization

The absolute volume of the audio is irrelevant for key identification. I would like to be able to identify the relative energy of each pitch in certain time frame. Therefore, I apply L2 normalization on the column of the matrix in order to reduce the disturbance due to change in absolute volume.

d. Tuning Pitch Reference

For some pieces, the tuning system used for that pitch may not exactly match the standard frequencies in 12-equal temperament tuning. The idea of tuning pitch reference is to divide a semitone interval into several sub-bins, then determine which of these sub-bins contains the highest average energy. Then, the sub-bin is then used to represent the energy of that pitch.

In this project, I divide a semitone into 10 sub-bin, each 10 cents apart. It should be noted that most of the popular songs nowadays is produced, electronically, so most of the pitches are precise. However, there have been previous researches showing that there are pop songs which has the tuning reference deviating up to 50 cents from the standard tuning reference. [9]

e. Consonance Filtering

A sound that we hear in daily life is made of several waves. Most of these waves are the overtones of the fundamental frequency, which can be heard most clearly. In order to emphasize the frequency, in which we actually hear in the music, I use consonance filtering technique. The idea is the notes that can be heard simultaneously in music usually belong to a diatonic scale. In each time frame, I extract the pitch which has the highest energy, and keep repeating this process until the pitch extracted does not belong to the same diatonic scale as the previously extracted pitches. The pitches which are not extracted are then set to 0. [9]

f. Averaging Pitch-class Energy and Scaling

The energy from all frames are averaged, and then the energy of pitches which belong to the same pitch class is averaged again. From this, I obtain a vector of n values, where each value contains the average energy of the n-th pitch class from the audio file. I, then, subtract the minimal energy in the vector from every value in the vector, so that the minimum value is 0, and the other values represent how much energy of the corresponding pitch exceeds the minimum energy. Finally, the vector is scaled so that the highest value is 1, while the smallest remains at 0. The resulting pitch profiles are shown in Figure 2-4 in the Result and Discussion section.

2. Configuring Neural Nets

The neural network models developed in this project is multi-layered neural nets with back propagation, implemented in Python 3.0. There is one model for each pitch, and thus there are 12 models in total. Each model takes the same pitch profile as the input as outputs 1 if the pitch profile is in its key, or 0 if not. For a pitch profile, we will have 12 outputs from 12 models. The model which gives the highest output (closest to 1) is selected as the final identification.

Each neural network model has 1 hidden layer. The size of the hidden layer and the targeted MSE are to be determined experimentally, as will be described in the next section. These two variables are very important, because it is related to overfitting. Other variables, including number of hidden layer, learning rates, and momentum are determined, empirically and fixed for all experiments.

Experiment

- **Dataset**

For this project, I create a labeled dataset consisting of 120 pop songs – 10 for each key – covering several genres. The reason for creating my own dataset is because, firstly, I would like to limit this project to songs without pitch-shifting, and secondly, I find that the performance of the model drops significantly when I include both Western pop and Korean pop songs, compared to when I work only on Western pop songs. Among 120 songs in this dataset, 76 songs are Western pop and 44 are Korean pop. The dataset can be found here.

It should be noted that I only indicate the tonic pitch of the song for this project and do not distinguish between major and minor mode, due to time constraint. Although this reduces the number of possible outputs, it also poses a significant disadvantage: the model trained on songs from one mode will not recognize songs in another mode, even though they have the same tonic pitch.

- **Experiment Setup**

In all experiment, each neural network model has 1 hidden layer. The learning rates are fixed to be 0.05 for the hidden layer and 0.02 for the output layer. Momentum is set to be 0.2 to accelerate the training process, slightly. The experiment is for determining the most optimal size of hidden layer and targeted MSE for training. Each experiment differs in the type of pitch profiles used for training. The dataset is divided into training set (80%) and testing set (20%).

  - **Experiment 1:** 24 bins/octave (Each bin is quarter-note apart) without tuning pitch reference and consonance filtering.
  - **Experiment 2:** 12 bins/octave (according to 12-equal temperament tuning) with tuning pitch reference but without consonance filtering.
  - **Experiment 3:** 12 bins/octave with both tuning pitch reference and consonance filtering.

Result & Discussion

- **Pitch Profiles**

Figure 2-4 below show pitch profiles of the same song generated for 3 experiments. The orange column shows the tonic pitch of the song. For Figure 2, the bins with even number index always have less energy than the bar on its left. This is because the tuning of pop songs nowadays is mostly precise, as explained earlier. Figure 3 and 4 only have 12 bins, since its tuning reference is determined. It should be noted that in Figure 3, 2 pitches have highest energy. Therefore, this case will fail if we use standard probing for pitch-class with maximum pitch class energy. There are also some small differences between Figure 3 and Figure 4 for other pitch classes, which are the effects of consonance filtering.

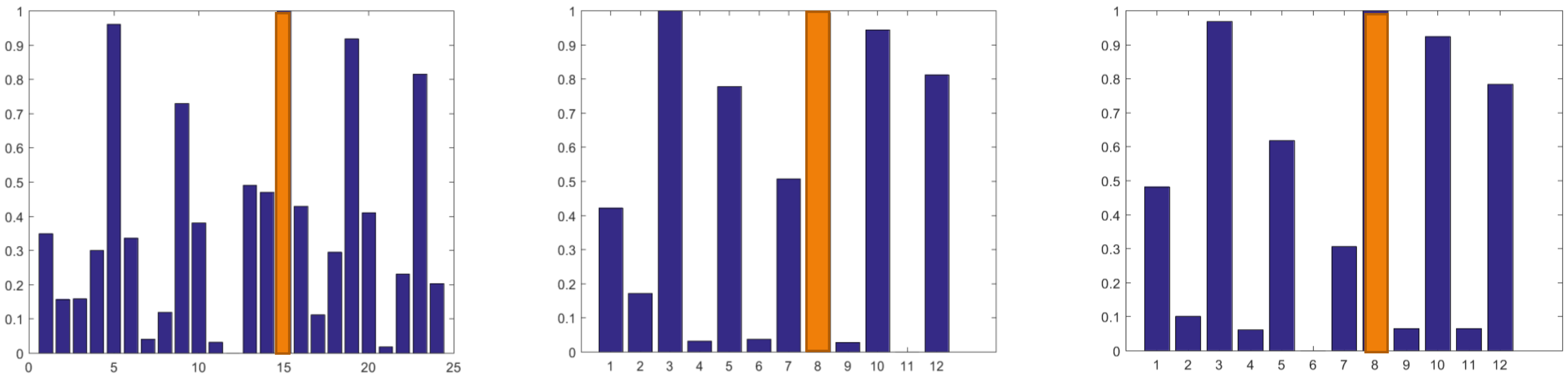

Figure 2-4 shows pitch profiles of the song "Youth" by Troye Sivan. Figure 2 (left) shows pitch profile constructed from result of CQT with 24 bins. Figure 3 (middle) shows pitch profile constructed from performing tuning pitch reference on CQT with 120 bins, reducing the dimension of the data down to 12. Figure 4 (right) shows pitch profile constructed with both tuning pitch reference and consonance filtering.

- **Identification Performance**

Each experiment is run on various combination of targeted MSE and number of hidden neurons, in order to obtain the highest accuracy possible. For each combination of targeted MSE and number of hidden neurons, the experiment is run 5 times, and the resulting accuracy is averaged, as shown below in Table 1-3.

Targeted MSE **# of Hidden Neurons**	**0.0005**	**0.001**	**0.005**	**0.01**
3	76.665	76.666	82.499	76.667
4	81.665	79.166	83.333	77.5
5	79.165	77.498	**85.834**	80.0
6	80.833	78.498	79.999	79.167

Table 1 shows the accuracy of neural net models which take pitch profiles with 24 bins as input. The best performance is obtained with 5 hidden neurons and targeted MSE of 0.005.

Targeted MSE **# of Hidden Neurons**	**0.0005**	**0.001**	**0.005**	**0.01**
2	76.664	72.5	83.333	77.5
3	75.832	76.667	77.499	**83.334**
4	71.666	73.331	73.612	73.331

Table 2 shows the accuracy of neural net models which take pitch profiles with 12 bins without consonance filtering as input. The best performance is obtained with 3 hidden neurons and targeted MSE of 0.01.

Targeted MSE **# of Hidden Neurons**	**0.0005**	**0.001**	**0.005**	**0.01**
2	79.168	77.499	**84.159**	74.994
3	75.833	73.332	75.8328	73.334
4	71.666	70.832	83.334	73.335

Table 3 shows the accuracy of neural net models which take pitch profiles with 12 bins with consonance filtering as input. The best performance is obtained with 2 hidden neurons and targeted MSE of 0.005.

We can see that for experiment 1, the size of input is the largest, so the number of hidden neurons is also the highest. The number of hidden neurons for experiment 2 has to be slightly higher than that of experiment 3, possibly because the input for experiment 3 has been further preprocessed with consonance filtering.

Furthermore, we can see that too many hidden neurons or too small targeted MSE do not improve the accuracy. If we observe the result of each of the 5 run, there are a few runs which give very high accuracy, but there would also be times when the accuracy is very low, so the average comes out to be low. This is likely to be an effect of overfitting the training data.

- **Comparison with other approaches**

As shown in Figure 4 below, the performance accuracies of multi-layered neural nets are above 80% for all 3 experiments. It is slightly below the accuracy obtained by of convolutional MLP as presented in [1]. It should also be noted that the performance accuracy for convolutional MLP approaches is based on different datasets, as explained in Related Work section. Therefore, it is inconclusive which approach is better for audio key identification task.

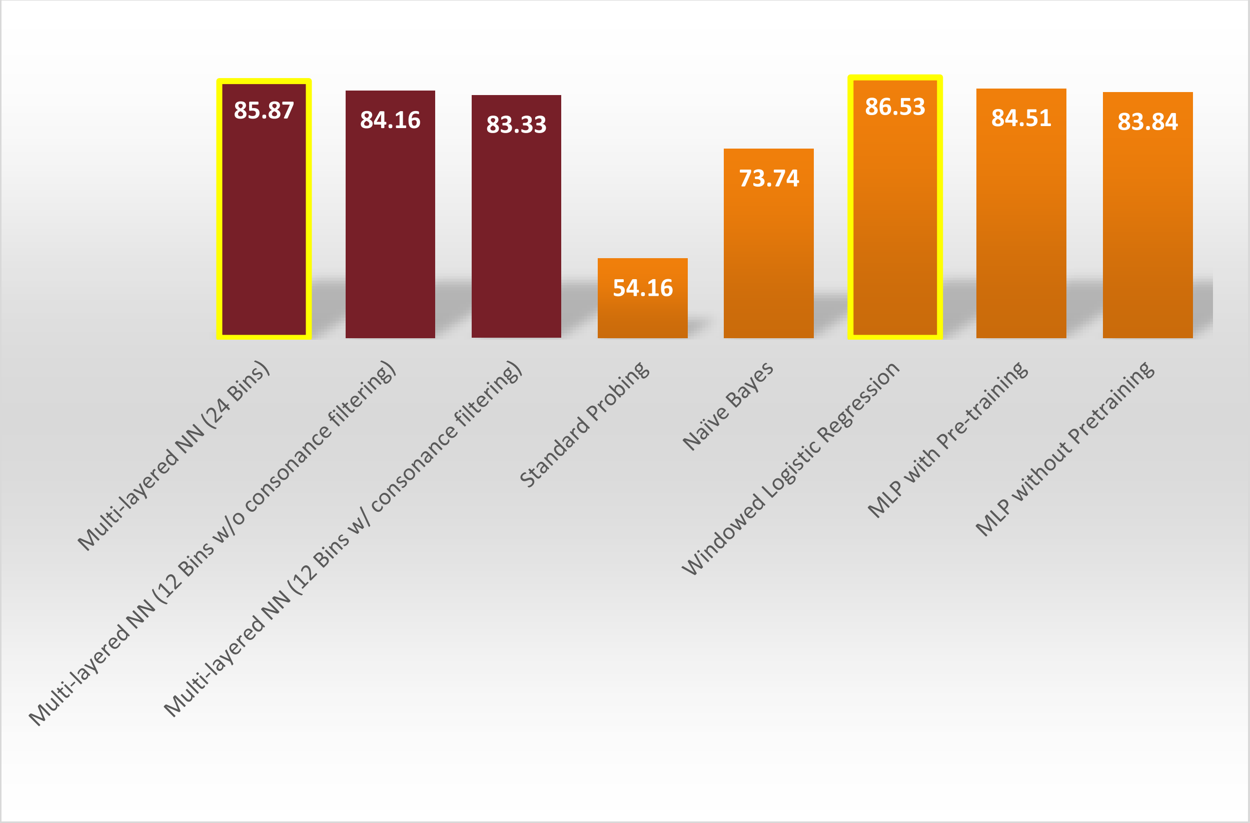

Figure 4 shows the comparison of performance accuracy of audio key identification task by several approaches.

Conclusion

Multi-layered neural nets can identify the tonic pitch of audio with accuracy above 80% for all types of pitch profiles. The accuracy is close to that of convolutional MLP on timbre and chroma features.

Future Work

This technique could be extended to identify the mode of the song (major and minor). Moreover, it could be used with temporal weighting technique, where the tonic pitch is determined separately for the beginning segment, the middle segment, and the ending segment of the song, and the tonic pitch of the song is determined by weighting these 3 tonic pitches. Temporal weighting with more division can also be used to determine the tonic pitches for songs with pitch-shifting.

Reference

1. Dieleman, Sander, Philémon Brakel, and Benjamin Schrauwen. "Audio-based music classification with a pretrained convolutional network." *12th International Society for Music Information Retrieval Conference (ISMIR-2011)*. University of Miami, 2011.

2. Izmirli, Özgür. "Audio Key Finding Using Low-Dimensional Spaces." *ISMIR*. 2006.

3. Levine, Nathan J., "Exploring Algorithmic Musical Key Recognition" (2015). CMC Senior Theses. Paper 1101. http://scholarship.claremont.edu/cmc_theses/1101

4. Liu, Yuxiang, et al. "Clustering Music Recordings by Their Keys." *ISMIR*. 2008.

5. Ni, Yizhao, et al. "An end-to-end machine learning system for harmonic analysis of music." *IEEE Transactions on Audio, Speech, and Language Processing* 20.6 (2012): 1771-1783.

6. Pauws, Steffen. "Musical key extraction from audio." *ISMIR*. 2004.

7. Temperley, David. "What's Key for Key? The Krumhansl-Schmuckler Key-Finding Algorithm Reconsidered." *Music Perception: An Interdisciplinary Journal*, vol. 17, no. 1, 1999, pp. 65–100., www.jstor.org/stable/40285812.

8. Tsai, T. J., Thomas Prätzlich, and M. Meinard. "Known-artist live song id: A hashprint approach." *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*. 2016.

9. Zhu, Yongwei, Mohan S. Kankanhalli, and Sheng Gao. "Music key detection for musical audio." *Multimedia Modelling Conference, 2005. MMM 2005. Proceedings of the 11th International*. IEEE, 2005.

10. Zhu, Yongwei, and Mohan S. Kankanhalli. "Precise pitch profile feature extraction from musical audio for key detection." *IEEE Transactions on Multimedia* 8.3 (2006): 575-584.