Triad Sequence Detection Model

Quan Yuan
School of Computing Science
Simon Fraser University
qya23@sfu.ca

Yuheng Liu
School of Computing Science
Simon Fraser University
yla622@sfu.ca

Wenlong Wu
School of Computing Science
Simon Fraser University
wwa95@sfu.ca

Guanhua Wang
School of Computing Science
Simon Fraser University
gwa34@sfu.ca

Abstract

According to what we have researched, many of the audio studies are focused on genre classification. In this project, we plan to build a model for classifying the triads[1] and the order of triads in a song. CNN, SPP, and RNN will be three important techniques used in building the model. CNN is designed as a feature extractor with SPP built inside, and RNN learns the output of CNN. This is the basic workflow of the triad sequence detection model.

1. Introduction

1.1 Problem and Related Work

Sometimes people may want to learn to play a melody using an instrument, but there is no music score for it. This problem makes people hard to learn, and the goal of our project is to solve this problem by detecting the type of triads used on this melody and the order of triads. More specifically, there are three types of triads we want to classify, which are major, minor, and diminished triads.

Before starting to design the framework of the project, we have done a lot of researches on the triad classifier and did not find any related paper or code. Thus, we decided to study some general audio classification related papers. Karol[1] proposed an idea that using the Convolutional Neural Network to classify short audio clips of environmental sounds. He claims that CNN performs better than some common approaches with 44% baseline accuracy and 64.5 % best network on the ESC-50[2] dataset. We were inspired by Karol's research that introducing CNN in the process of building our model was a feasible method. Furthermore, CNN could be helpful for us to classify the triads at the beginning of the model establishment. We also referred to an existing research[2] achievement from Nicolas, Yoshua, etc. Their chord recognition research with RNN has a close connection to our model constructing and data sequences analysis steps.

Based on the researches above, as CNN performs well on image classification and RNN is applied to data on time sequence, we decided to combine CNN and RNN to classify the triads and the order of triads.

[1] In music, a triad is a set of three notes(or "pitch classes") that can be stacked vertically in thirds.
[2] The ESC-50 dataset is a labelled collection of 2000 environmental audio.

1.2 Dataset

The data in the dataset are all audio files. There are two types of data:
1. Contains only one triad
2. Contains a sequence of triads and melody
The whole data set was generated by ourselves because we did not find such pure audio files.

2. Approach

2.1. Data Pre-Processing and Feature Extraction

There are two types of input data, a single triad and a song with triad sequence and melody. Both of them are the audio files generated by GarageBand. Librosa is the package chosen for audio data processing. By using its load function, the audio data is loaded to be the one-dimensional data. However, it is not easy to extract features for triad from one-dimensional audio data. Thus, to solve this problem, the single triad data is converted to Mel-Spectrogram using librosa's function. And the data with triad sequence and melodies is divided into pieces of data first so that each piece roughly contains only one triad. Then every triad would be converted to a Mel-Spectrogram representing its features.

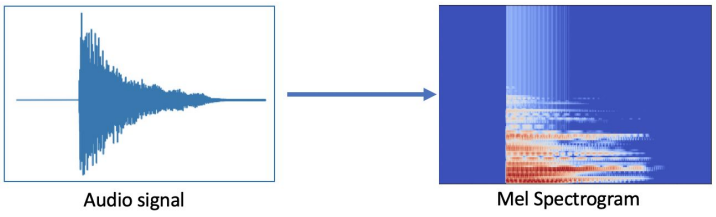

Figure 1. Data Transform

2.2. Modelling

The net combines CNN and RNN to predict the triad sequence in an audio file. The divided audio with only one triad is sent to CNN first. The output of CNN is a binary sequence that represents the class of each audio and it is also the input of RNN process. The net ends up obtaining an output sequence through RNN process.The final output sequentially shows the triads used in an audio file. The flow-chart of the model shows below.

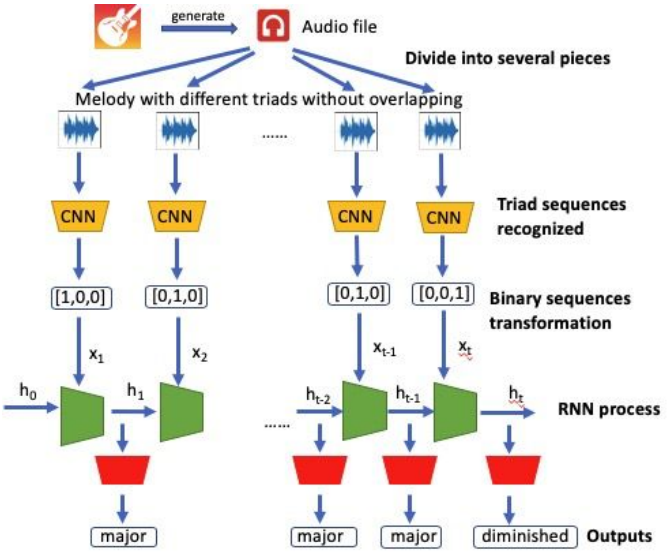

Figure 2. Model

CNN

In order to do a basic classification, CNN is designed as the first step of building the model. Due to the characteristic of extracting features efficiently from an image, CNN is an appropriate model to do the classification using a spectrogram. The CNN process contains two 2D-convolutional layers and each convolutional layer is followed by a pooling layer in order to reduce the dimensionality of the input. This showed a good performance with efficient training in a prior benchmark[3][4]. Also, ReLU is used as an activation function to introduce non-linear property. The SPP layer is followed by two convolutional layers in order to solve the problem that inputs have different sizes. The output is shown in a binary format through two fully-connected layers. At this step, we can get an initial prediction to conduct the later process.

SPP

SPP is a layer added in the CNN model between the convolutional layer and the fully-connected layer, and it is the most significant part of the project. As the input of the fully-connected layer should be of the same size, the size of the input image is always fixed for the CNN model. SPP successfully resolves the problem because the inputs of different sizes will be converted to the same size through SPP. The reason for using SPP is that the length and the speed of each audio file are different in reality, each triad will be of different lengths. Therefore, by using SPP the inputs to the fully-connected layer can be controlled to the same size. A figure explaining the details of SPP shows below.

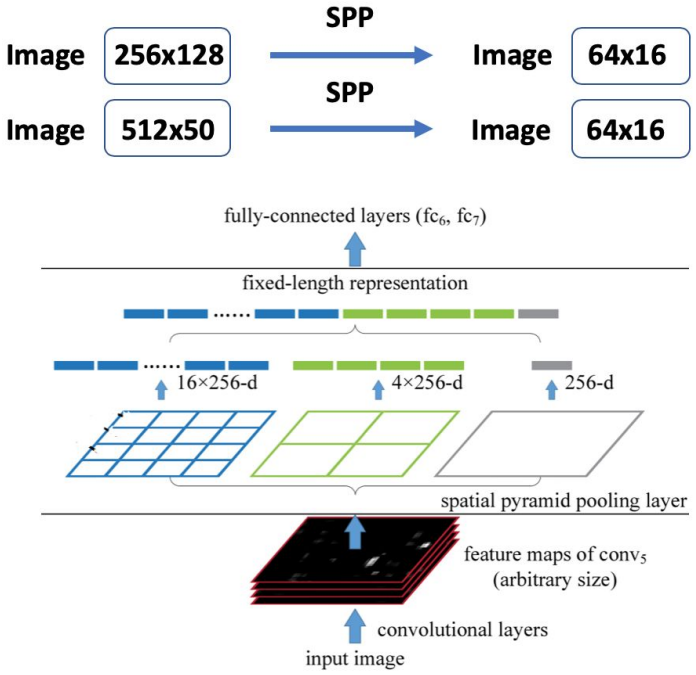

Figure 3. SPP

Initially when we applied SPP to the data, sometimes the image caused RuntimeError: pad should be smaller than half of the kernel size. To fix it, we researched a lot and found a solution which was updating padding according to the input data and then calculating the new kernel and stride[5]. Also, training with variable-size images increases scale-invariance and reduces over-fitting[6].

RNN

RNN is critical at the processes of handling the binary sequences and doing the classification to give the different triad recognizing results. It can be regarded as the ending step in our model to produce the

recognizing results. There are some rules of the usage of the triad in the music composition such as using a fixed triad order. RNN is used to learn those rules and fix the outputs predicted by the previous CNN model. The outputs of the RNN stage should be a sequence of types of triads. The activation function is tanh and it has one fully-connected layer after the RNN layer. The output of each unit is a number that represents the class of a triad.

2.3. Training and Testing

For the CNN stage, we created some pure audio files with only one triad played by piano and used them as inputs. Cross-Validation was used to choose the parameters and the number of layers, including the parameters of the SPP layer. The structure and parameters are shown in the Table I.

Table 1. Structure and Parameters

Conv2D Layer	input size=1, output size=16, kernel=5, padding=1
BatchNorm2d Layer	num_features=16
ReLU Layer	
MaxPool2d	kernel_size=(2, 2)
Conv2D Layer	input size=16, output size=32, kernel=3, padding=0
BatchNorm2d Layer	num_features=32
ReLU Layer	
MaxPool2d	kernel_size=(2, 2)
SPP Layer	output_num = (32, 16, 8)
fully-connected layer 1(linear transform)	input_size=43008, output_size=512
fully-connected layer 2(linear transform)	input_size=512, output_size=3

Due to the SPP layer, the model could process inputs with different size. During the training phase, we unified the size of the training set. The length of each audio was 3 seconds. During the testing phase, some new data played by other instruments was generated and was used to test the effect of the CNN model. Audio files played by piano and other instruments were used as testing inputs. The accuracy of using audios played by piano was much higher than using audios played by other instruments. However, the accuracy of using audios played by other instruments shows that the CNN model is able to find out some specific features which represent the type of a triad. After testing, we used the whole dataset to build and train our final CNN model used in the whole net.

For the RNN stage, we also created some audio files as the training and testing dataset. Each of them was a piece of melody with the accompaniment using a different combination of triads. The orders of triads used in the audio files were fixed. Several popular orders of triads were selected to be used. We set the initial hidden state to a zero matrix and used the output of the CNN model as the input of the RNN model. The purpose of the training was to let the RNN model learn those orders so that it could

fix the mistakes made by the previous CNN model. For the allocation of the dataset, we separated the training and testing sets in a 7: 3 ratio.

To examine the final model quantitatively, both the number of correctly classified triads and the number of correctly classified audio sequence were calculated.

3. Comparative Experiments

As mentioned above, SPP was used to transform the images from the convolutional layer into the images of the same size. According to Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition[1], SPP shows outstanding performance in the accuracy of image classification. However, unlike the normal images, the images used in this project were Mel-Spectrograms transformed from audio data, so we tried to find whether SPP performs well on those Mel-Spectrograms or not.

The table below shows the results of using CNN with and without SPP to classify the triads.

Table 2. Results of Comparative Experiments

	With SPP	Without SPP
Loss	0.20-0.30	0.05
Accuracy(Tested on same tone)	81%	90%
Accuracy(Tested on different tone)	32%	53%

When we were building the CNN model for the one-triad dataset, the running loss for every epoch was printed out. With SPP, after 500 epochs the running loss remained stable at the range between 0.20 and 0.30. However, the running loss of building the CNN model without SPP could be lowered to 0.05. Therefore, the model with SPP resulted in a lower accuracy on classifying the triads than the model without SPP. Obviously, SPP did not perform well on Mel-Spectrogram. The reasons for this might be the limited size of datasets and that the way extracting features from Mel-Spectrogram was different from that of other types of images. Also, for both the models with SPP and without SPP, the accuracy heavily decreased when tested on a different tone. Thus, we knew that the triad could be affected by tone.

4. Results

As mentioned before, we tested the effect of the whole net using audio files with a piece of melody and its accompaniment. The testing result is shown below.

Table 3. Result

	One Triad	Triad Sequence
Accuracy(correctly classified)	88%	40%

To test the model, we created some songs with the accompaniment using a different combinations of triads, there were several fixed orders in the combinations. The well-trained CNN model is used to make a basic prediction and the outputs of the CNN model are delivered to the RNN model as the inputs. Since the size of the dataset is not large, the training time was around one hour. The accuracies

of the prediction with the testing set were 88% on a single triad and 40% on a whole audio sequence. Which means that the accuracy of classifying the class of a single triad in the whole audio files is high but it always has a few misclassified triads in an audio sequence. Since each divided audio has not only one triad but also other sounds, the accuracy of classifying a single triad is high enough to demonstrate that the net is capable to find out the critical feature. For the result of classifying an audio sequence, each sequence appeared one or two misclassified units, which means the model could learn the most part of the fixed triad order.

5. Summary

In order to help music amateurs analyze the composition of a song easily, a model with the combination of CNN and RNN is built to detect single triads in an audio file. According to the results, the hypothesis that using CNN to classify the triads and RNN to learn the sequence of triads is feasible. And as the most important part maintains the same input size, SPP slightly lowers the accuracy, but the decrease is acceptable. It is recommended to train the model by larger datasets with different tones and the real song files.

6. Future Work

Although the RNN stage could use the speed/tempo provided in the metadata file to split the audio file equally into several pieces which each piece only has one triad, it is better to detect the start point and the endpoint of a triad automatically. Because the time of one triad used in a song is uncertain. If we just split the audio file into several pieces with the same length/minimum value, there will be some redundant or unnecessary predictions. Building a model to learn the variance of the spectrogram may be a good solution to solve this problem.

For the CNN stage, we just used Mel-Spectrogram as inputs and it was limited for finding out some certain features of the dataset. Constant-Q Transform(CQT) is another spectrogram with the pitch information. Using the combination of CQT and Mel-Spectrogram as the input could extract more information of a triad in order to increase the accuracy. Meanwhile, we could use this feature to achieve a more complicated classification such as figuring out which level a triad is.

The dataset used in this project was generated by ourselves, so the composition of an audio file is simple. As a matter of fact, a song is full of different sounds and all these sounds could influence the effect of our model. We plan to use some real songs as inputs to train and test our model and probably reduce the influence of the “noise” in the audio files.

7. Contribution

Task	**Member(s)**
Dataset Generation	Quan Yuan,Wenlong Wu
Dataset ETL	Quan Yuan, Yuheng Liu
CNN implementation	Quan Yuan, Yuheng Liu
Training and Testing on CNN model	Quan Yuan, Yuheng Liu, Wenlong Wu, Guanhua Wang
SPP implementation	Yuheng Liu, Guanhua Wang

RNN implementation	Quan Yuan, Wenlong Wu
Training and Testing on the whole net	Quan Yuan, Yuheng Liu, Wenlong Wu, Guanhua Wang
Poster	Quan Yuan, Yuheng Liu, Wenlong Wu, Guanhua Wang
Report	Quan Yuan, Yuheng Liu, Wenlong Wu, Guanhua Wang

References

[1] K. J. Piczak, "Environmental sound classification with convolutional neural networks," 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP), Boston, MA, 2015, pp. 1-6.

[2] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, “Audio chord recognition with recurrent neural networks,” in Proc. ISMIR, 2013, pp. 335–340.

[3] K. Choi, G. Fazekas, M. Sandler and K. Cho, "Convolutional recurrent neural networks for music classification," 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, 2017, pp. 2392-2396.

[4] K. Choi, G. Fazekas, M. Sandler and K. Cho, "A Comparison of Audio Signal Preprocessing Methods for Deep Neural Networks on Music Tagging," 2018 26th European Signal Processing Conference (EUSIPCO), Rome, 2018, pp. 1870-1874.

[5] Marsggbo, sppnet-pytorch, (2018), GitHub repository: https://github.com/marsggbo/sppnet pytorch/blob/master/SPP_Layer.py

[6] K. He, X. Zhang, S. Ren and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 9, pp. 1904-1916, 1 Sept. 2015.