Music Recommendation System

James Gao
Jieqiao Luo
Xubo Lin
Moxiao Liu
Xinru Shi

Table of Contents

“Sorting out all digital music is very time-consuming and causes information fatigue. Therefore, it’s necessary to develop a music recommender system”

Introduction

- Music is one of the popular entertainment media in the digital era. It can be categorized into several categories: pop, rock, jazz, blues, folk etc.
- The availability of digital music is very abundant compared to the previous era. To reduce the difficulty of sorting out all the music genre, building a recommendation system is essential.
- Many existing music applications already have their own systems, like Spotify and Pandora.

Introduction

- What is a music recommendation system?

  It's a system that can search in the music libraries automatically and suggest suitable music to users.

- Generally, MRS could be divided into 3 main parts.

  Users

  Items

  User-item matching algorithms

Three parts of MRS

Users

Develop user modeling based on user profiling

Items

Item profiling comprises of three kinds of metadata: **editorial, cultural and acoustic**

User-item matching

Matching algorithm consists of **collaborative filtering** and **content-based filtering**.

What are collaborative and content-based filtering?

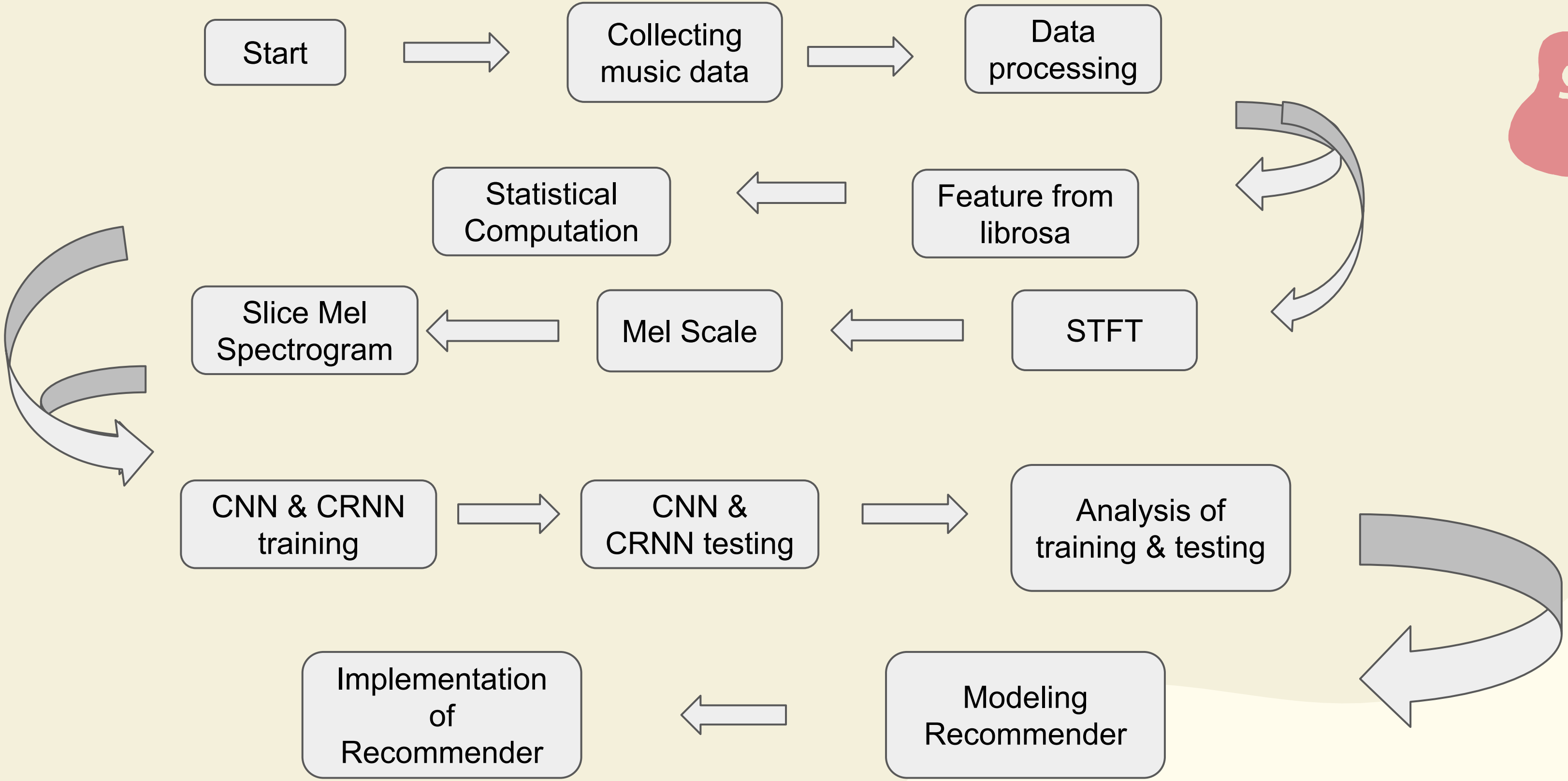
- **Collaborative filtering** used the collaborative power of the available assessment by users to make recommendations.
- **Content-based** approach typically employ a **2-stage approach**, extract traditional audio content features and predict user preferences. In this project, we tend to focus on content-based approach.

Dataset

- General requirements for these datasets are **large scale, permissive licensing, available** and **quality audio** and **easily accessible**.
- The dataset we chose is called **Free Music Archive (FMA)**.
- There are **8** music genres in this FMA dataset, which are the labels we will use.
- '***Electronic', 'Experimental', 'Folk', 'Hip-Hop', 'Instrumental', 'International', 'Pop', 'Rock'.***

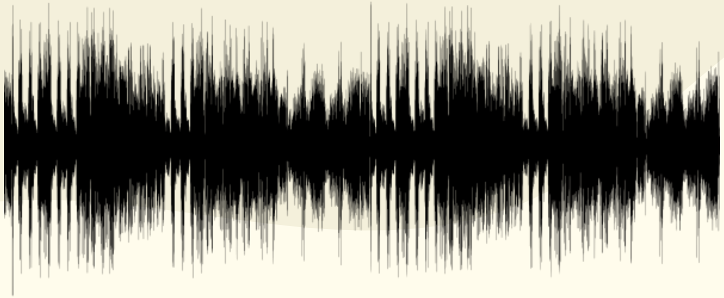
dataset[1]	#clips	#artists	year	audio
RWC [12]	465	-	2001	yes
CAL500 [45]	500	500	2007	yes
Ballroom [13]	698	-	2004	yes
GTZAN [46]	1,000	~ 300	2002	yes
MusiClef [36]	1,355	218	2012	yes
Artist20 [7]	1,413	20	2007	yes
ISMIR2004	1,458	-	2004	yes
Homburg [15]	1,886	1,463	2005	yes
103-Artists [30]	2,445	103	2005	yes
Unique [41]	3,115	3,115	2010	yes
1517-Artists [40]	3,180	1,517	2008	yes
LMD [42]	3,227	-	2007	no
EBallroom [23]	4,180	-	2016	no[2]
USPOP [1]	8,752	400	2003	no
CAL10k [44]	10,271	4,597	2010	no
MagnaTagATune [20]	25,863[3]	230	2009	yes[4]
Codaich [28]	26,420	1,941	2006	no
FMA	**106,574**	**16,341**	**2017**	**yes**
OMRAS2 [24]	152,410	6,938	2009	no
MSD [3]	1,000,000	44,745	2011	no[2]
AudioSet [10]	2,084,320	-	2017	no[2]
AcousticBrainz [32]	2,524,739[5]	-	2017	no

Basic Workflow



Concepts

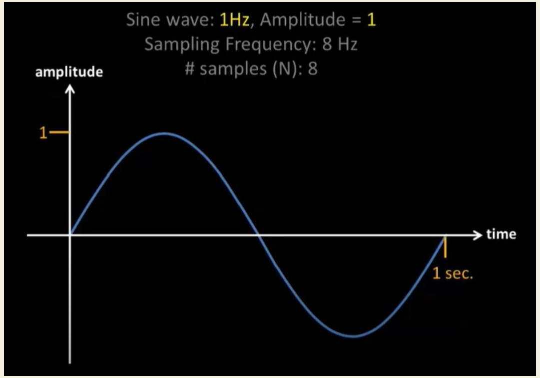
In our project, files in .mp3 format will be decoded into NumPy array real values.

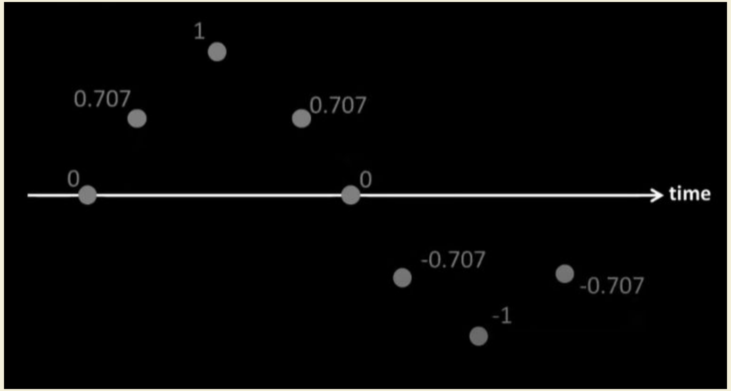


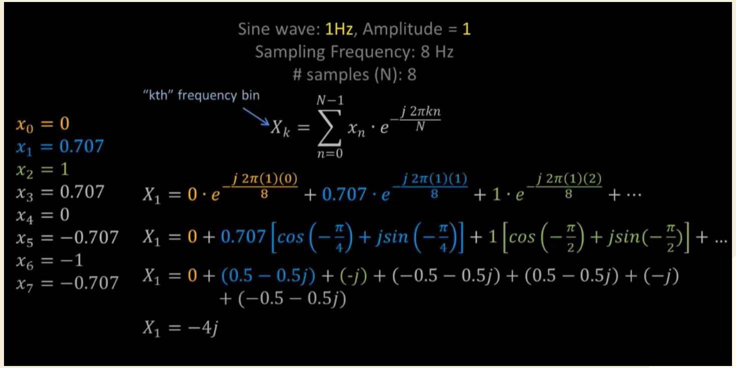
Transformation between wave and array

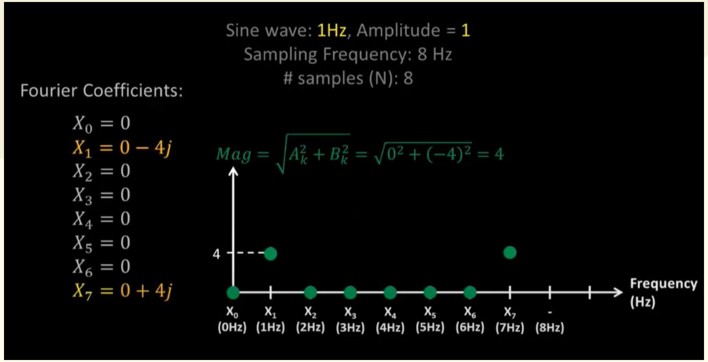
An n-degree polynomial p in real space can be represented by n+1 pairs of x, p(x), so if we consider sound wave as a continuous function with x label for time and y label for amplitude. We will be able to get an approximation of the original wave function.

Wave Decomposition









Short-Time Fourier Transform

Idea:

take small signal pieces of length L

look at the DFT of each piece

long window -> more DFT points -> more frequency resolution

long window -> more "things can happen" less precision in time

short window -> many time slices -> precise location of transitions

short window -> fewer DFT points -> poor frequency resolution

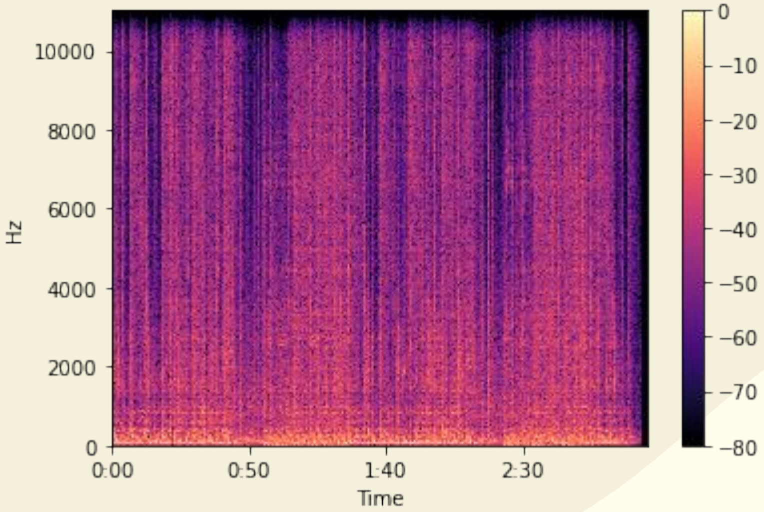
Spectrogram

The spectrogram is the color code of the magnitude of the Fourier transform.

Uses dark color dark hues for small values and whitish or brilliant hues for large values.

Puts the spectral slices one after another to obtain an image-like picture of the time-varying spectrum.

With frequency bins.



Genre Classification Workflow

Start → Raw MP3 music data → Data processing → Get feature from librosa → Statistical computation → 1D CNN training → 1D CNN testing → Analysis of training & testing

classify 16 music genres

Feature extraction librosa

zero-crossing rate(ZCR)

constant-Q transform(CQT)

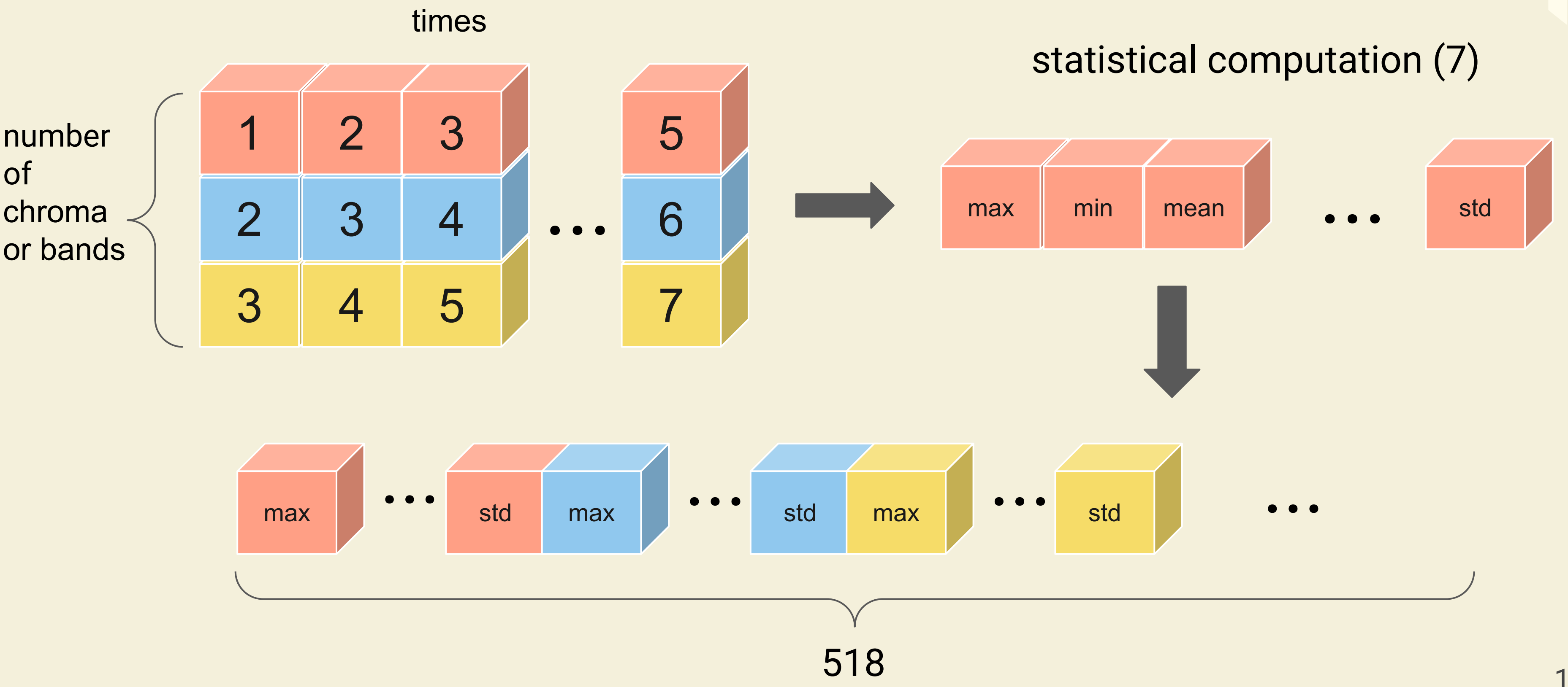
chroma_cqt, chroma_cens, tonnetz

Short-Time Fourier Transform(STFT)

chroma_stft, rms, spectral_centroid, spectral_bandwidth, spectral_contrast, spectral_rolloff, melspectrogram

Mel-frequency cepstral coefficients (MFCC)

For each Feature



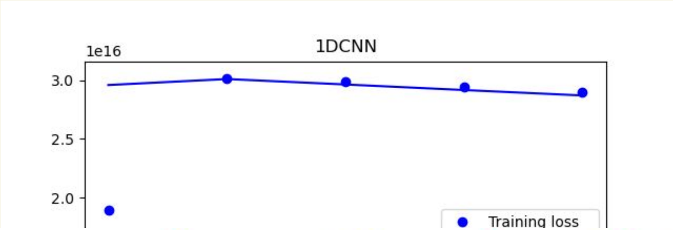
1D CNN

Model: "sequential"

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 487, 200)	6600
max_pooling1d (MaxPooling1D)	(None, 243, 200)	0
conv1d_1 (Conv1D)	(None, 212, 8)	51208
max_pooling1d_1 (MaxPooling1D)	(None, 106, 8)	0
conv1d_2 (Conv1D)	(None, 75, 8)	2056
max_pooling1d_2 (MaxPooling1[illegible]		
flatten (Flatten)		
dense (Dense)		
dense_1 (Dense)		

Total params: 91,180
Trainable params: 91,180
Non-trainable params: 0

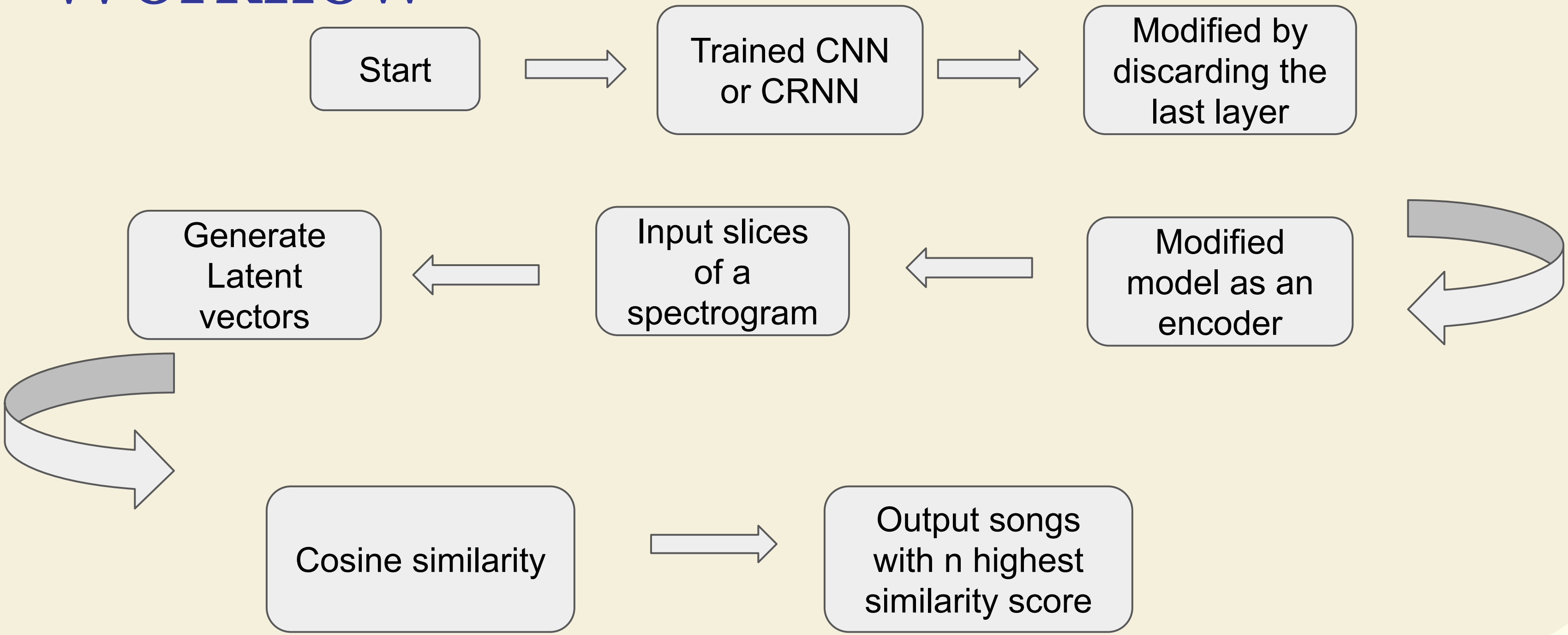
Accuracy：28.49%

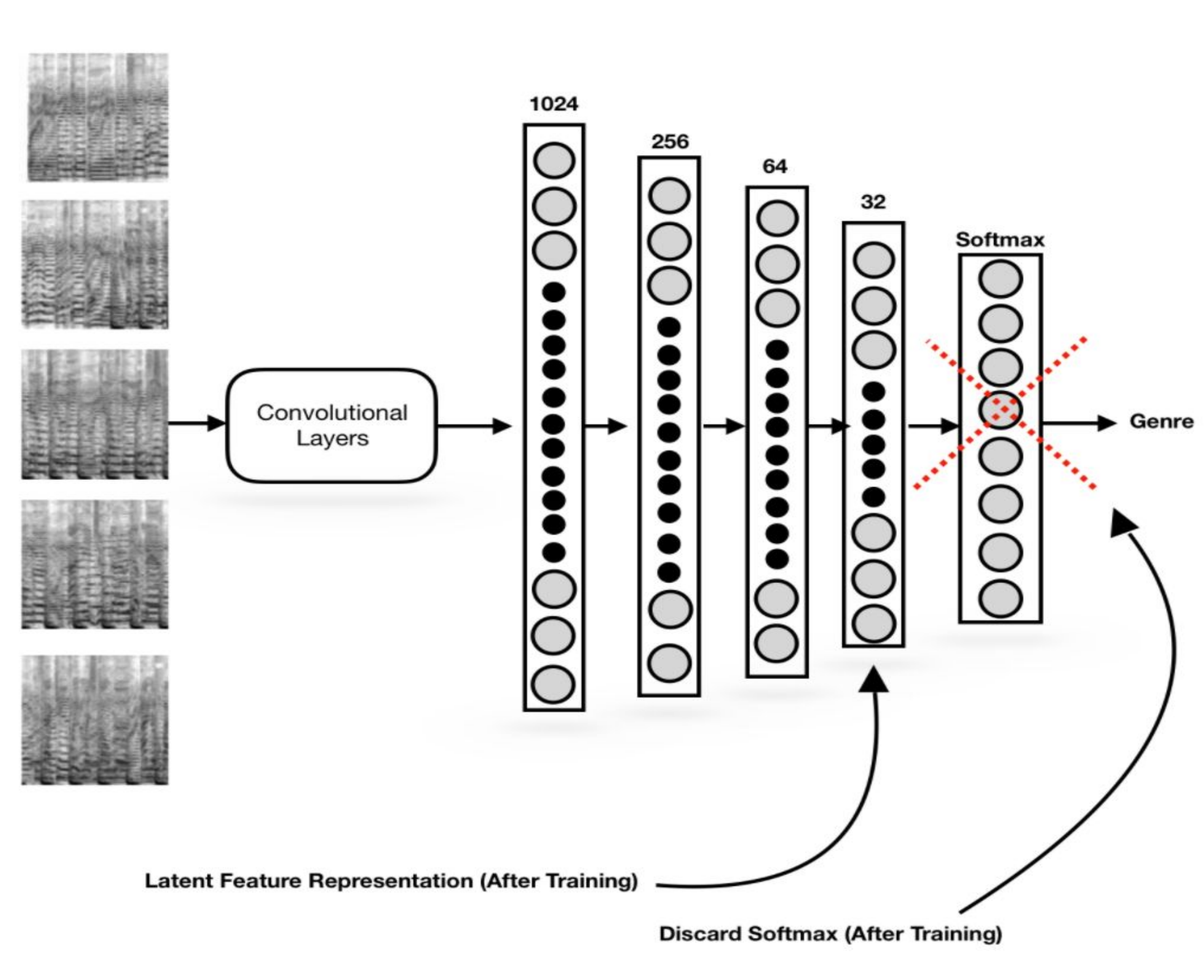


	dim	LR	SVC	MLP
mfcc	140.000000	11.23%	12.13%	10.49%
mfcc/contrast/chroma/centroid/tonnetz	322.000000	12.90%	13.41%	9.25%
mfcc/contrast/chroma/centroid/zcr	287.000000	13.06%	13.64%	9.37%

LR SVC MLP: https://github.com/mdeff/fma

Mel-Spectrogram Recommendation Workflow



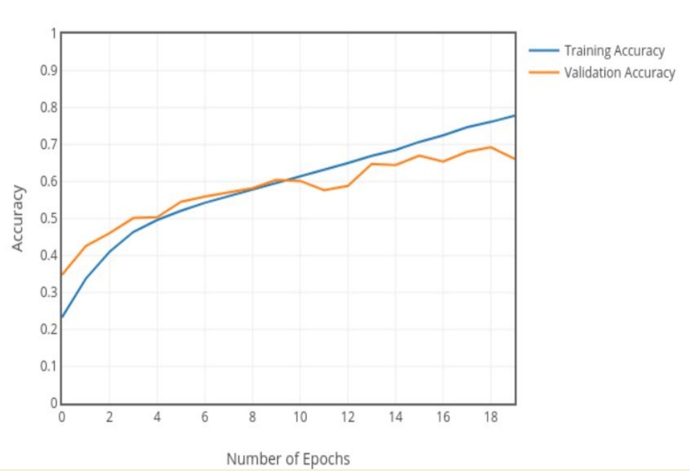


1024
256
64
32
Softmax
Convolutional Layers
Genre
Latent Feature Representation (After Training)
Discard Softmax (After Training)

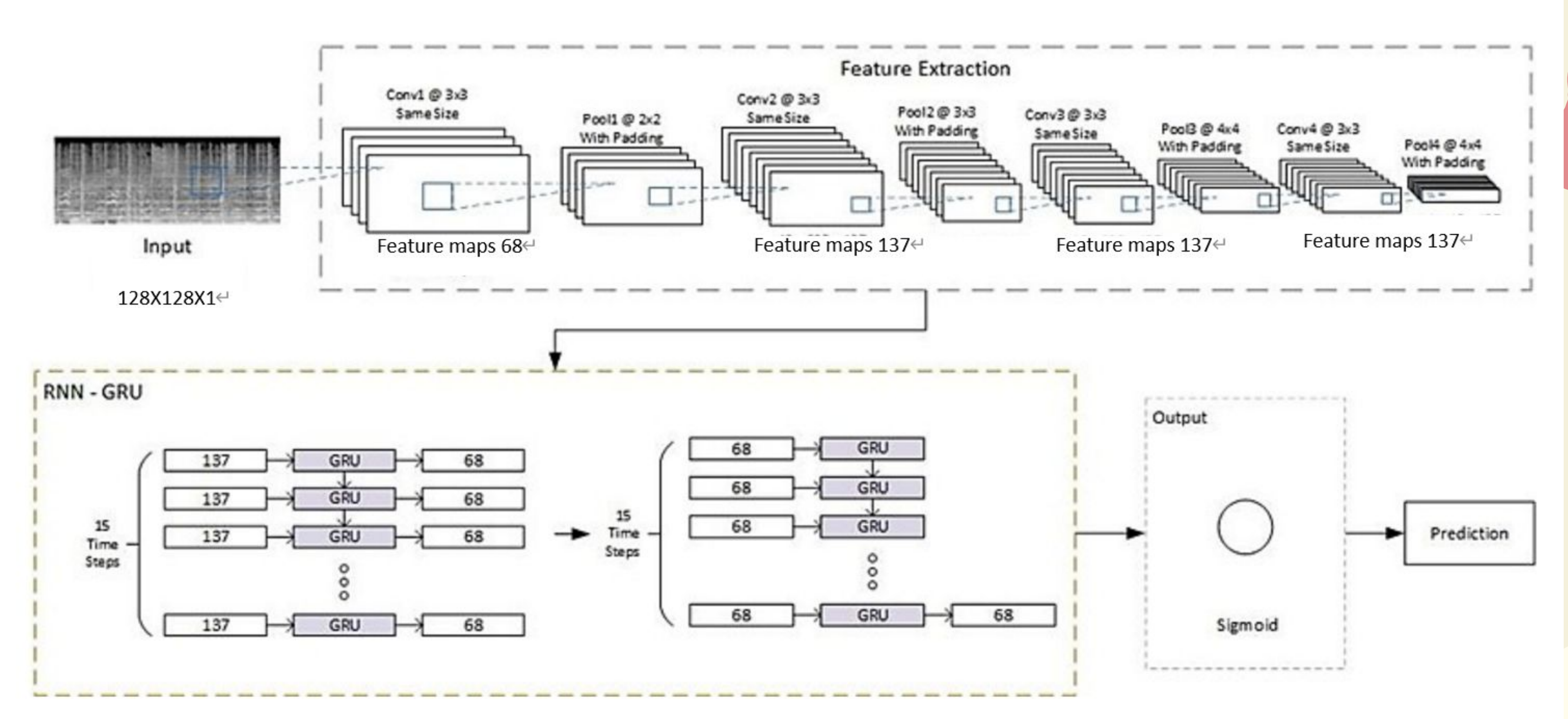
CNN Algorithm for Mel-Spectrogram Approach

Layer (type)	Output Shape	Param #
input_3 (InputLayer)	[(None, 128, 128, 1)]	0
conv2d_10 (Conv2D)	(None, 128, 128, 47)	470
max_pooling2d_10 (MaxPooling2D)	(None, 64, 32, 47)	0
conv2d_11 (Conv2D)	(None, 64, 32, 95)	40280
max_pooling2d_11 (MaxPooling2D)	(None, 32, 8, 95)	0
conv2d_12 (Conv2D)	(None, 32, 8, 95)	81320
max_pooling2d_12 (MaxPooling2D)	(None, 16, 2, 95)	0
conv2d_13 (Conv2D)	(None, 16, 2, 142)	121552
max_pooling2d_13 (MaxPooling2D)	(None, 6, 1, 142)	0
conv2d_14 (Conv2D)	(None, 6, 1, 190)	243010
max_pooling2d_14 (MaxPooling2D)	(None, 2, 1, 190)	0
flatten_2 (Flatten)	(None, 380)	0
dense_2 (Dense)	(None, 8)	3048

Total params: 489,680
Trainable params: 489,680
Non-trainable params: 0



CRNN Algorithm for Mel-Spectrogram Approach



Accuracy : 0.75

Summary

Process

Step 1

Input one song, use
CNN to classification

acc : 30%

Step 2

acc : 70%<75%

Compare the similarity
of this song with the all
other sons in this class

CRNN > CNN

Step 3

Output n number most
similar songs (n
choose by the user)

Summary

Some Problems

- Little difference between music files
- hyperparameter
- Which methods are better

- Different features
- Different network
- Problem of hyperparameter

Summary

Future work

Preprossing

Unify the size of the array

Hyperparameter

Choose better hyperparameter

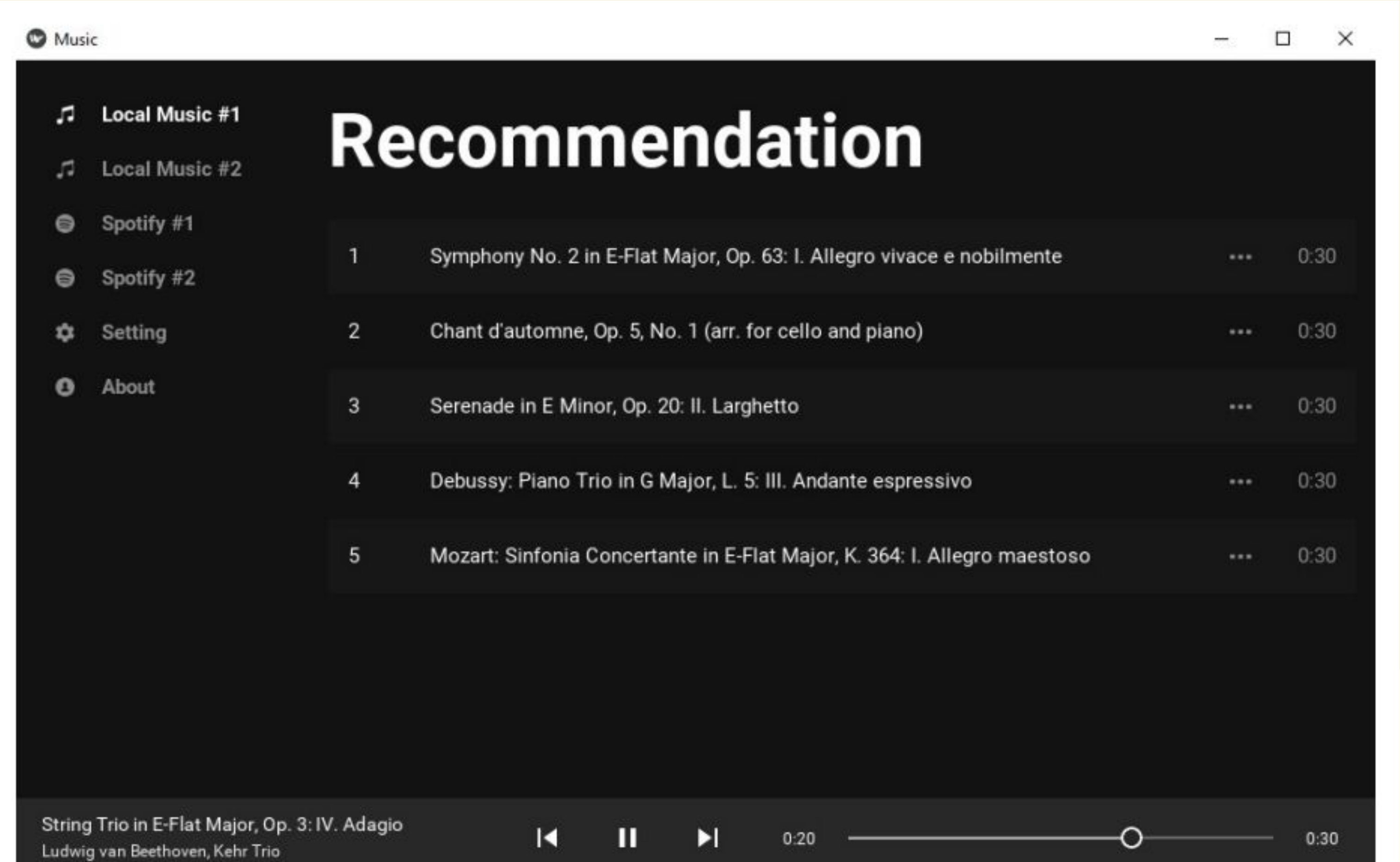
Model

Improve model

Compared and Confirm method

Compared our two method and find the best way

Future work


Music
Local Music #1
Local Music #2
Spotify #1
Spotify #2
Setting
About
Recommendation
1 Symphony No. 2 in E-Flat Major, Op. 63: I. Allegro vivace e nobilmente 0:30
2 Chant d'automne, Op. 5, No. 1 (arr. for cello and piano) 0:30
3 Serenade in E Minor, Op. 20: II. Larghetto 0:30
4 Debussy: Piano Trio in G Major, L. 5: III. Andante espressivo 0:30
5 Mozart: Sinfonia Concertante in E-Flat Major, K. 364: I. Allegro maestoso 0:30
String Trio in E-Flat Major, Op. 3: IV. Adagio
Ludwig van Beethoven, Kehr Trio
0:20
0:30

Thanks

References

1.Adiyansjaha, A. A S Gunawana, D.Suhartonoa, ”Music Recommender System Based on Genre using Convolutional Recurrent Neural Networks”. The 4th International Conference on Computer Science and Computational Intelligence (ICCSCI 2019)

https://www.sciencedirect.com/science/article/pii/S1877050919310646

2.Dataset:https://github.com/mdeff/fma

3.M. Schedl, “Deep Learning in Music Recommendation Systems”,Front. Appl. Math. Stat., 29 August 2019

https://www.frontiersin.org/articles/10.3389/fams.2019.00044/full