DEEP LEARNING SPRING 2024 - FINAL PROJECT

AUDIO CLASSIFICATION AND MUSIC GENERATION

1. Introduction

Ever pondered on the process behind computers creating music and discerning various genres much like humans do? Can computers translate audio into mathematical representations to formulate algorithms for distinction? Moreover, if Artificial Intelligence were to compose music, could it surpass human creativity? These inquiries serve as the motivation for this project and I am here to explore the capabilities of computer in the music industry.

2. Previous solutions

After a lot of research on the internet, the solutions before was using regular machine learning classifications tasks (i.e. XGBoost, DecisionTree, Support Vector Machine, …) or 1D Convolutional Neural Network. Tthey use the statistical feature extracted from the data (i.e. mean) of the Mel-frequency cepstrum coefficients. These solutions do not give too high accuracy score on training (~ 50%) and it is hard to extract those feature on a large scale.

3. Dataset

I used the gtzan dataset for training, which is a collection of 10 genres with 100 audio files each, all having a length of 30 seconds.
For testing and evaluation, I extthe Free Music Archive large dataset, also with 100 audio file each of 10 genres, 30 second length.

Instead of using mean feature of the mel-frequency cepstrum coefficients, I will use the raw data itself and load it into a DataLoader object which will randomly choose random parts in the song.
Dataloader object size:

- 10000 for training
- 2000 for validation
- 1000 for testing
- Sequence length: 15s and 20s

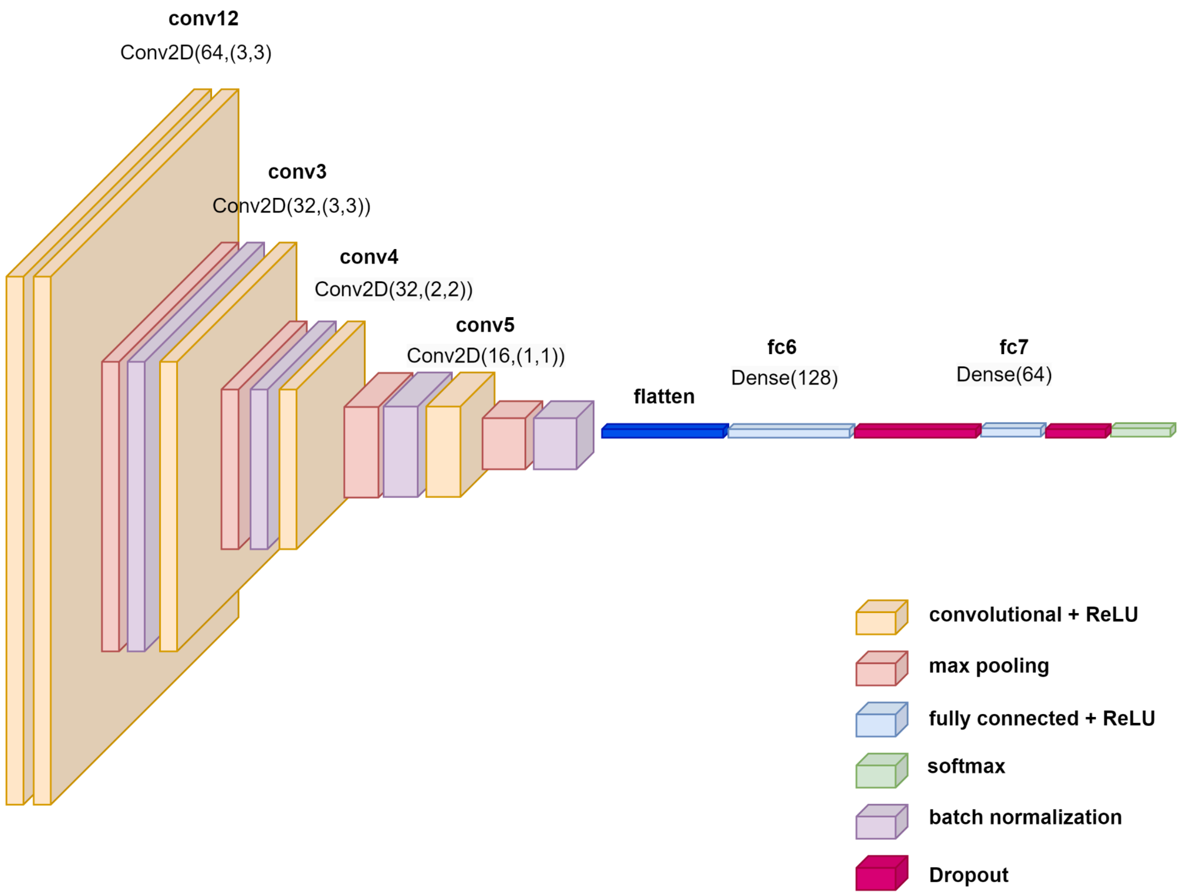
librosa.feature.mfcc extracting preset :

- n_fft = 2048,
- n_mfcc = 13 (default), 40
- hop_length = 512

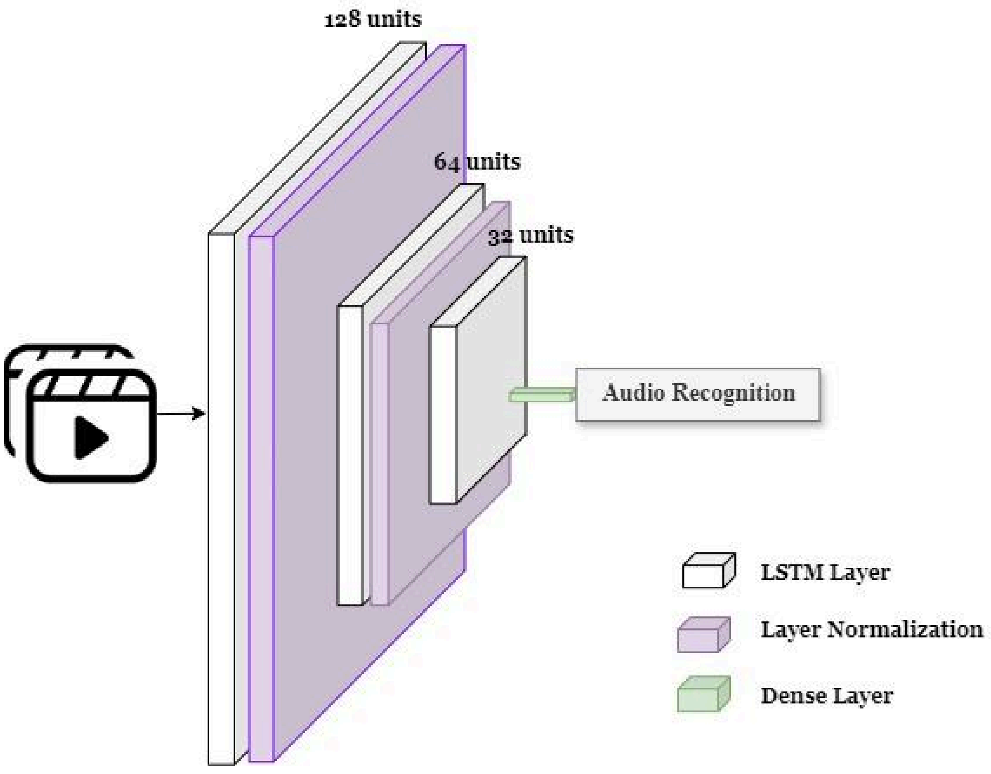
4. Proposed solution

I propose 2 solutions: 2D CNN and LSTM.
2D Convolutional Neural Network

Where the input shape is (864, n_mfcc,1) fors seq_len = 20
and (648, n_mfcc, 1) for seq_len =15



Long short-term memory on seq_length = 15 and n_mfcc = 40



5. Evaluation method.

We test our model on the fma dataset we generated, method we use is accuracy score. I also added a confusion matrix of the output for reference and discussion.

6. Results

Accuracy score	n_mfcc = 13 (default)	n_mfcc = 40
seq_len = 15	0.289	0.34
seq_len = 20	0.273	0.314

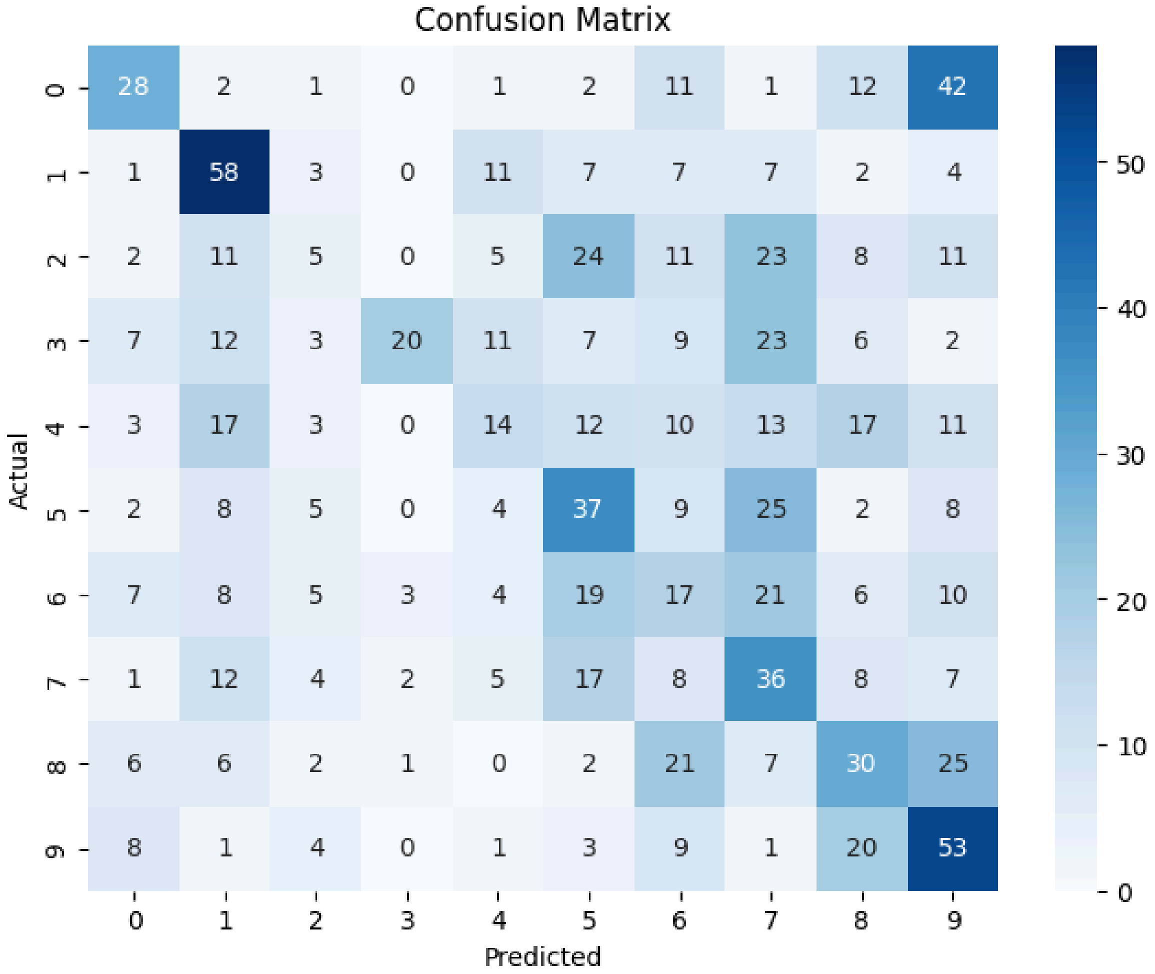
LSTM accuracy : 0.27

7. Discussion

This confusion matrix is from CNN model with seq_len=15 and n_mfcc = 40

We have the labels :

```
'Disco': 0, 'Blues': 5,
'Metal': 1, 'Pop': 6,
'Country': 2, 'Hip-Hop': 7,
'Classical': 3, 'Rock': 8,
'Reggae': 4, 'Jazz': 9
```



- Metal and Jazz are easiest recognizable, with 58% and 53% accuracy.
- Blues are 37% guessing accuracy, with their most error guess onto hip-hop, which is understandable since Blues is the fundamental of hip-hop in the 90s
- Pop tends to be the hardest genre to be misunderstood with other genres, with wrong predictions spans everywhere, and to be mistaken with blues and hip-hop more than accurate guess.
- For some reason classical are misunderstood with hiphop (23 guess?). Although there are just 26 songs guessed classical, 20 of them are correct, therefore it can be that other genres adapt from classical music rhythm
- Country is the hardest to differentiate with just 5 out of 100 correct guess. They are popularly wronged as blues, pop…
- Reggae has 10+ wrong predictions in 6 others genre, this should be the hardest to predict since this genre seems to be a mix of everything.

8. Potential Fixes

- To increase the accuracy score, we can use raw audio data, in addition with:
    + Song’s name (some genre are easily classified just by their title)
    + Song’s lyrics
    + Year of Release
- The data need to be distinct between genres, there should be no mixed genre songs.
- More accurate testing data. After some trials, I see that the fma_data has some flaws in it.