The Four Music-ateers

50.039 Music Genre Classification

Sherry Lee Pei Ching (1004584)
Shang Xiangyuan (1004446)
Lee Jet Xuen (1004365)
Tay Sze Chang (1004301)

1. Introduction

Music still remains as one the most popular forms of art that are consumed by people nowadays. Many music streaming companies are trying to improve their service by improving the recommendation system. Part of the effort was done via recommending the same genre of music according to the users' preferences. Music genre is a conventional category identifying music with a similar musical form and musical style. Music genre classification aims to tag audio signals with genre labels, which will then be used in building the recommendation system.

In this project, we implemented a 4-layers CNN model to complete the task of music genre classification.

2. Datasets

2.1 GTZAN Dataset

The GTZAN dataset is the most-used public dataset for evaluation in machine learning research for music genre recognition. The dataset consists of a total of 1000 audio tracks with a 30s duration each. The dataset is divided into a total of 10 genres, each with 100 tracks. The 10 genres are blues, classical, country, disco, hiphop, jazz, metal, pop, reggae, rock. All the tracks are 22050Hz Mono 16-bit audio files in .wav format.

One thing worth taking note is that one of the audio files (jazz.0054.wav) was corrupted, so we downloaded a jazz music piece from YouTube and extracted the middle 30s of the music to replace the corrupted files.

2.2 Custom Dataset

Training on pure GTZAN dataset might lead to a biased model due to its small size and bias. Therefore, we created our own dataset, which was used in both training and testing phases. The final dataset that was used consists of both the pure GTZAN training set and 7 songs for each genre, 70 songs in total. The dataset was splitted into 80%, 10%, 10% for the training, validation, test dataset respectively.

The full-length songs we used were downloaded from YouTube, they are all the top songs recommended by Spotify List[1].

2.3 Data Preprocessing

The original audio signal was converted into a log scale mel-spectrogram with 128 mel bands and 16000 sample rate. A mel-spectrogram is a 2-dimensional representation of a music signal, it provides a mel-scaled frequency representation which involves compressing the frequency axis of short-time Fourier transform representation. We divided the original spectrogram into

smaller chunks to generate 128 x 128 feature maps. Each chunk represents approximately 3 seconds of audio sequence.

When processing the dataset, for the GTZAN dataset, we took the full track of each audio to generate the mel-spectrogram. But for the full songs in the custom dataset, we removed the first and last 10 seconds of each song as they tend to contribute little to the genre classification task.

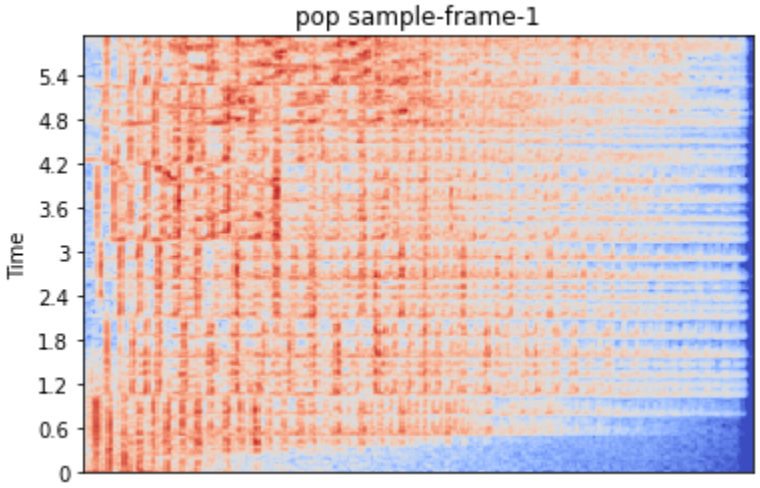

Fig. 1. Sample image of log mel-spectrogram

3. Model

3.1 Model Structure

For our model, we used 4 convolutional layers followed by 3 fully connected layers. The convolutional layers use kernels of size 3 x 3. While the input feature map of the convolutional layers is reduced from 128 x 128 to 2 x 2, the feature map channel number is increased from 1 to 512, as illustrated in Fig 2 below.

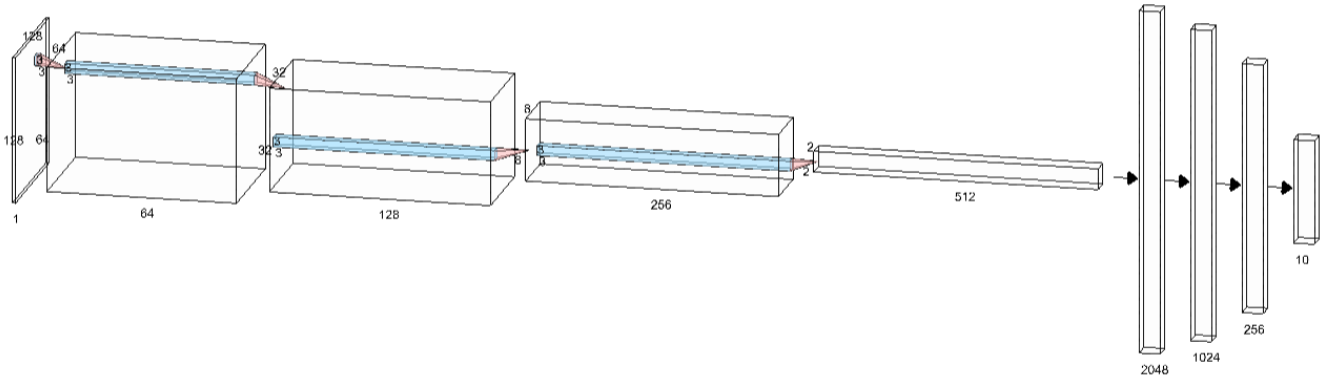

Fig 2: CNN model architecture. This model is inspired by Choi[2]

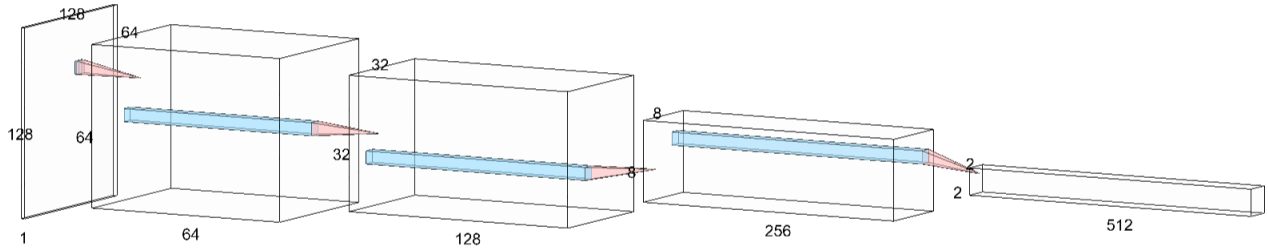

Fig 3: Convolutions layer block

Flattening the output into a 2048 x 1 vector, it is used as input for 3 fully connected layers. The last layer is a sigmoid activation function with a 10 x 1 vector as output. The output is the probability of the music belonging to the respective genre.

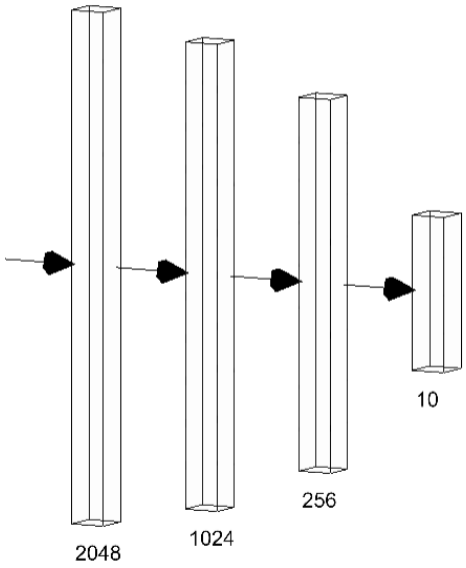

Fig 4: Fully connected layers

As suggested[3, 4], in order to increase the training speed, xavier uniform initializer was implemented in each convolutional layer. Additionally, to prevent overfitting, a drop out of 0.6 was added to all our fully connected layers.

3.2 Training Phase

During the training stage, we found out that learning rate of $1 * e^{5}$, batch size 20, and 40 epochs gave us an acceptable result. Binary Cross-Entropy (BCE) was used for back propagation, and RMSprop optimizer was used to adjust model parameters. The BCE Loss is given by:

$$l(x, y) = mean(L)$$

$$L = \{l_1, ..., l_N\}^T$$

$$l_n = -w_n[y_n \cdot log\, x_n + (1 - y_n) \cdot log(1 - x_n)]$$

The temporary prediction accuracy was stored for each epoch to adjust the learning rate. The accuracy was obtained by taking the number of successful predictions divided by the total tasks available. At the end of the training stage, we saved the model which gave the highest accuracy.

4. Results

We performed tests on the trained model with 2 types of audio format to increase the reliability of the model training result.

4.1 Test

The first audio format which was tested had the same format as the training data. The table belows summarized the performance of the trained model on all three datasets.

	Training Dataset	Validation Dataset	Test Dataset
Accuracy (%)	95.24	87.81	86.76
BCE Loss	0.0307	0.0667	0.0691

Table 1. Model accuracy

Overall the model achieved at least higher than 85% of accuracy in all three datasets, however, the model overfitted on the validation dataset after reaching the accuracy of 80% as shown in the figure below.

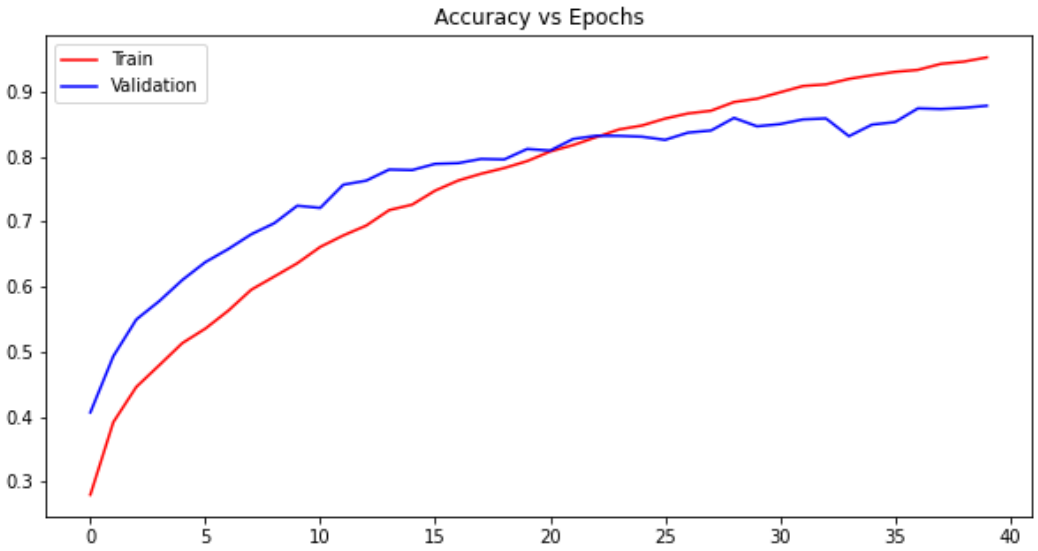

Fig. 3. Plot of Accuracy vs Epochs

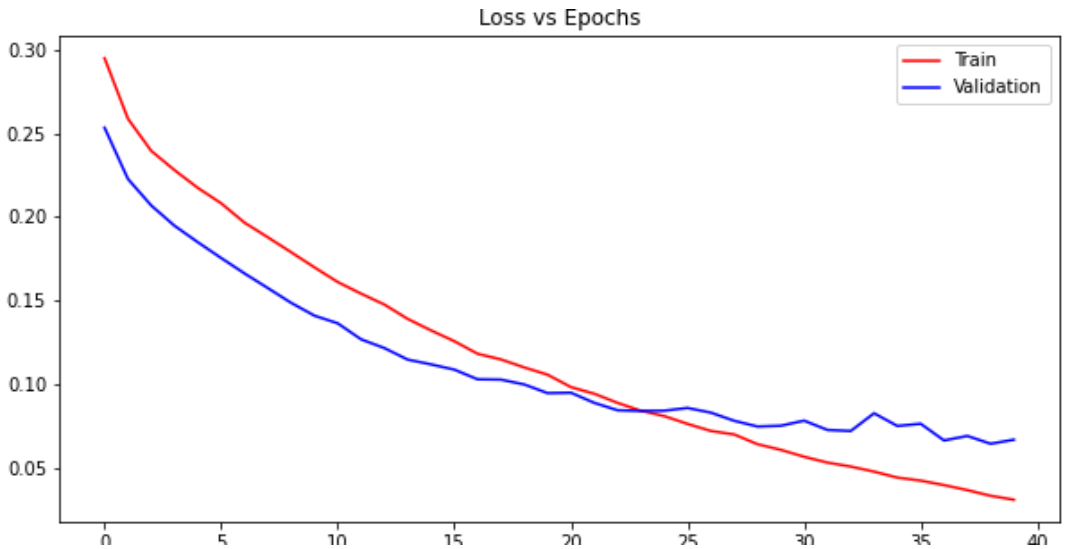

Fig. 4. Plot of Loss vs Epochs

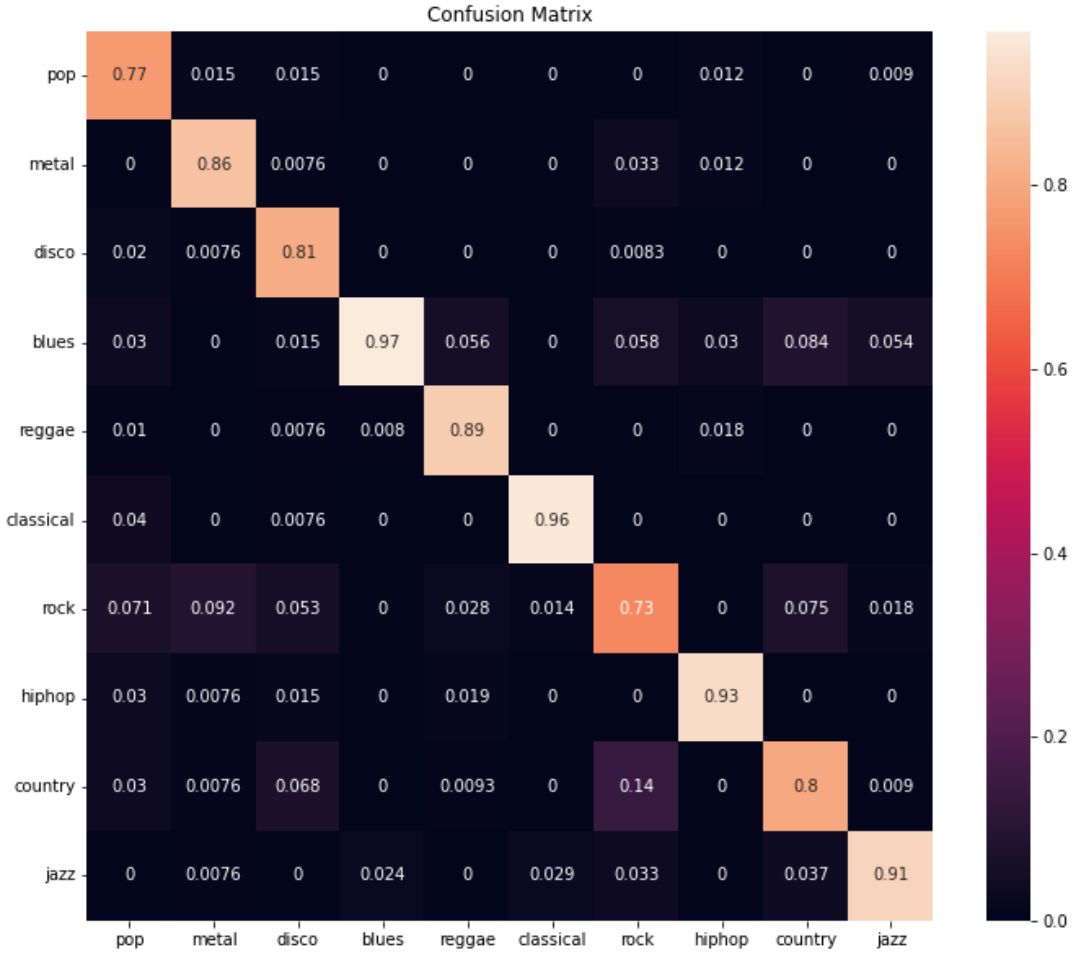

Fig. 5. Confusion matrix of training result

4.2 Test on Full-length Songs

In order to make our model more practical, we tested the models on 70 songs collected from the Internet. These songs are mostly based on the YouTube List recommendation, while some of them are sampled through random sampling using random songs generator by genre[6] which samples music from Spotify. This is to ensure unbiasedness when sampling.

4.2.1 Processing Full-Length Song

When processing these songs to be tested, we removed 15 seconds from the start and the end of the songs and converted them into smaller chunks of mel-spectrograms as per how we processed the training data. As modern songs often contain more than 1 single genre, a simple algorithm was proposed to analyze the genre composition for each song. The genre of a song was defined such that the number of chunks classified as one genre divided by the count of chunks generated by the first step.

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Test on sample hiphop
Test on Offset - Clout ft. Cardi B (Official Video).mp3
Genre hiphop    : 66.27%
Genre pop       : 28.57%
Genre blues     :  5.15%
**********************************************************************************************************
Test on 野狼disco.mp3
Genre hiphop    : 45.78%
Genre pop       : 26.66%
Genre metal     :  7.90%
Genre rock      :  6.21%
Genre jazz      :  4.26%
Genre classical :  3.73%
Genre reggae    :  2.94%
Genre blues     :  1.70%
Genre disco     :  0.81%
**********************************************************************************************************
Test on Wu-Tang Clan - C.R.E.A.M. (Official HD Video)-PBwAxmrE194.mp3
Genre hiphop    : 94.32%
Genre blues     :  4.49%
Genre disco     :  1.19%
**********************************************************************************************************
Test on Still D.R.E..mp3
Genre hiphop    : 96.91%
Genre reggae    :  3.09%
**********************************************************************************************************
Test on Eminem - Lose Yourself[1080p]..mp3
Genre hiphop    : 80.83%
Genre reggae    : 11.49%
Genre pop       :  7.68%
**********************************************************************************************************
Test on Flo Rida - Low ft. T-Pain [Apple Bottom Jeans] (Lyrics).mp3
Genre hiphop    : 72.62%
Genre pop       : 27.38%
**********************************************************************************************************
Test on Ms. Jackson.mp3
Genre hiphop    : 71.44%
Genre blues     : 16.92%
Genre pop       :  9.99%
Genre disco     :  1.65%
**********************************************************************************************************
Test on Thrift Shop (feat. Wanz).mp3
Genre hiphop    : 81.29%
Genre blues     : 11.29%
Genre rock      :  3.14%
Genre pop       :  2.42%
Genre country   :  1.86%
**********************************************************************************************************
Final Score for hiphop :0.8706542511947796
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Fig. 6. Sample result of test on full-length song

4.2.2 Score Calculation of Test on Full-length Songs

In order to evaluate the overall performance of the model on full-length songs, we took the top three genres of a test song then compared it with its ground truth label(genre). If any of them matched the ground truth label, we would add the score to the model score of this genre class. In an ideal situation, where each song is perfectly classified 100% to its ground truth label(genre), we should get a score of 7 for each genre as we have 7 songs for each genre. Then the score was normalized by dividing the final score by 7. The final result is shown in the bar chart below.

Overall, most of the genres had achieved at least a score of 0.5, some of the genres are well predicted, except for pop and country.

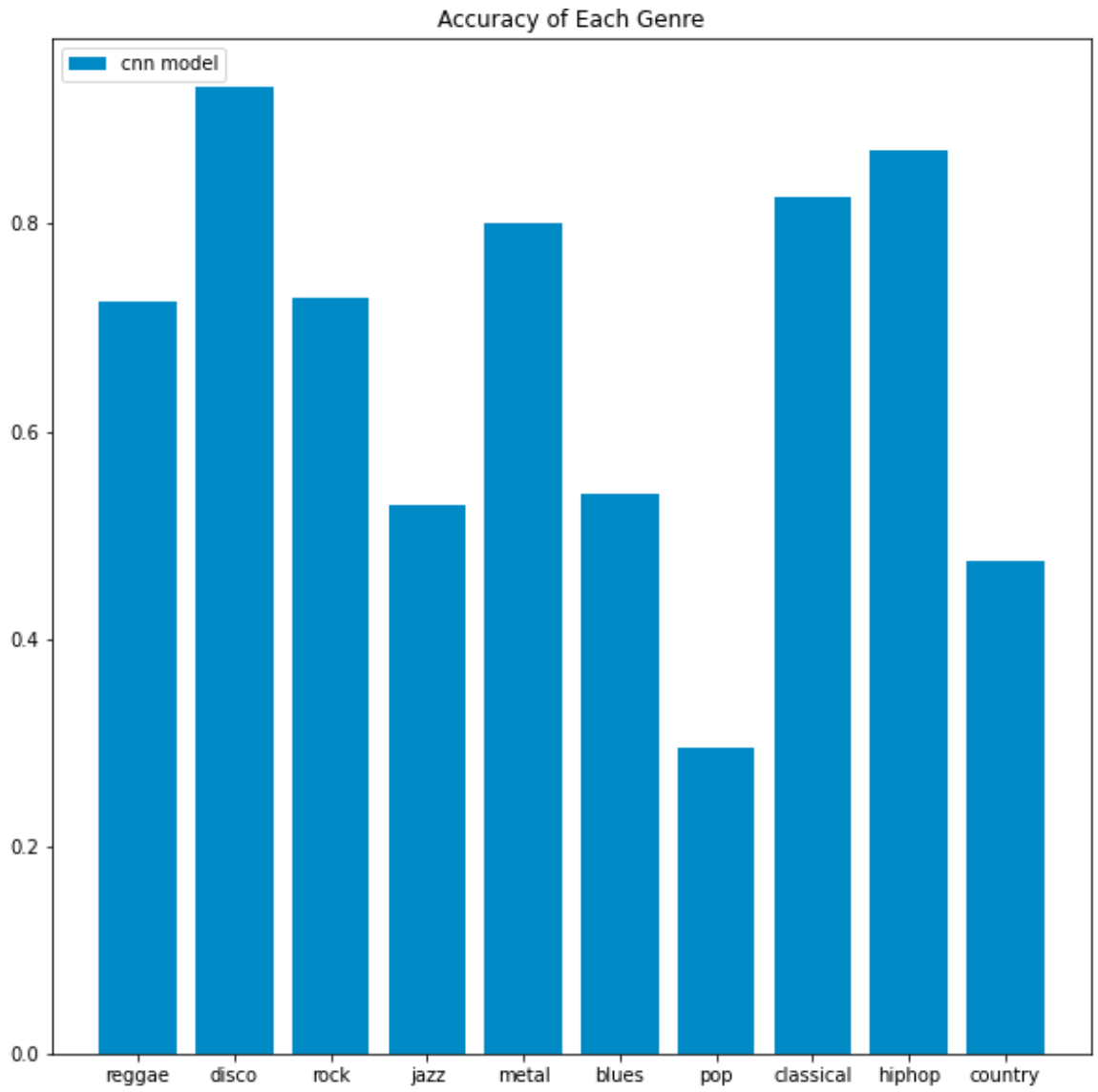

Fig. 7. Plot of accuracy of each genre on full-length test songs

5. Discussion

An interesting result is that the model tends to perform better on genres that have explicit features such as disco, hip hop and classical. The model fails to recognize pop as it has less than 0.4 score when performing tests on full-length songs. The similar trend was shown in the confusion matrix (Figure 5), where the score for pop is the lowest (0.77) among all the genres.

Based on the confusion matrix, we can tell that some of the music was wrongly classified as others, however this can be a proof that some of the music was influenced by other genres or it originated from other genres. Furthermore, there has been a lot of fusion music being developed over the time, such as country rock, country blues, etc. For example, in Figure 8, the song was

classified as hip hop with the highest level of confidence (45.78%), however, the score for pop was 26.66% along with 7 other genres but with much lower scores.

```
*****************************************************************************************************
Test on 野狼disco.mp3
Genre hiphop    : 45.78%
Genre pop       : 26.66%
Genre metal     :  7.90%
Genre rock      :  6.21%
Genre jazz      :  4.26%
Genre classical :  3.73%
Genre reggae    :  2.94%
Genre blues     :  1.70%
Genre disco     :  0.81%
*****************************************************************************************************
```

Fig. 8. Result on a test song

One shortcoming of our model was the small size of our dataset and caused overfitting. Even though we increased the size of our dataset with the custom dataset that we collected online, it was still a relatively small set to train a model. It might not be able to fully capture the relationship between different genres such as fusion, originality etc.

6. Conclusion

In the modern days, songs are mostly a fusion of multiple genres, hence it posed a challenge for deep learning neural networks to classify them to a single genre. Due to the limitation of resources and time, we can only achieve the result presented in the report, however, we believe there are many areas in which we can improve on. Although the model achieved a relatively decent performance, further improvements can be focused on enhancing the model. Such improvements can be made by adding an extra Recurrent Neural Network layer such as GRU to capture the temporal features of mel-spectrogram.

7. Contribution

The table belows summarizes each of our individual contributions to the project. Each member contributed to the project in different parts of the work, some of us focused more on the data collection, while others designed the model and tuning the parameters while training the model. While some of us were training the model, the rest started working on the report.

We suggested working in pairs due to our overwhelming workload from other modules, it was suggested in this way such that if one of us was occupied by other work, another member would be able to help out during this period.

	Jet Xuen	Sze Chang	Sherry	Xiangyuan
Primary Research	yes	yes	yes	yes

Data Collection	-	-	yes	yes
Data Preprocessing	yes	yes	-	-
Designing Model	yes	yes	yes	yes
Training & Tuning Parameters	yes	yes	-	-
Result Visualization	-	-	yes	yes
Report	yes	yes	yes	yes

Table 2. Team contribution

The model was trained entirely on Google Colab to avoid the potential problem which might arise from the usage of different operating systems and different versions of libraries used. It is strongly suggested to replicate the work using Colab to avoid any unnecessary problems.

GitHub Link: https://github.com/Sky0307/2022DLProject

References

1. Spotify List https://everynoise.com/everynoise1d.cgi?scope=all
2. Choi, K., Fazekas, G., Sandler, M., Cho, K. (2017). Transfer learn- ing for music classification and regression tasks. arXiv preprint arXiv:1703.09179
3. James Dellinger (2019) Weight Initialization in Neural Networks: A Journey From the Basics to Kaiming, https://towardsdatascience.com/weight-initialization-in-neural-networks-a-journey-from-the-basics-to-kaiming-954fb9b47c79
4. Xavier Glorot, Yoshua Bengio (2010) Understanding the difficulty of training deep feedforward neural networks
5. *Random Song Generator: Connected to Spotify and YouTube*. Chosic. (2022, April 11). Retrieved April 21, 2022, from https://www.chosic.com/random-songs-generator-with-links-to-spotify-and-youtube