# Music Genre Classification with CNN Model Evaluation

Yoonhee Jang
*Saint Johnsbury Academy Jeju*
Jeju, South Korea
jangyoonhee0809@gmail.com

**Abstract— Building a music genre classification model is one of the steps to consider the copyright. Through tuning, I was able to find the reasonable setting for the model. CNN shows high accuracy results rather than other Deep Neural Networks. It shows better results when the validation split is small and the filter size is bigger than 32. Additionally, batch size gives partial impact on the accuracy. Epochs can be the main factor to control the result but it is not for the music data.**

**Keywords-music genre classification, CNN, deep learning, model evaluation, GTZAN**

## I. Introduction

The world is rapidly advancing. AI is able to generate everything by learning what humans produce. AI does not have any limitations. It already started working in various fields like Chat GPT, Midjourney, and etc. Even in our daily life, AI is substantially used. For example, people create cover songs using their favorite singer's voice by AI voice generators. As AI developed, it is emerging as an ethical problem. The author or owner of the work are arguing AI's indiscreet learning which is against copyright and increase anxiety of AI exploit. Especially in the music field, it is a deliberate problem. To determine copyright infringement or not, similarity works as a key factor. Indeed, some process has to be available to determine the genre. Then, it observes deeply what part is similar between two same genre songs. Humans cannot determine every single copyright infringement problem. For these reasons, in this paper, I will build the model to classify music genres to the goal of distinguishing copyright infringement or not. In this process, I went through paper [1] process. It looked up data shapes when music changes into the data. To build a Deep learning model to identify the similarity of songs that AI made or humans made, it has to be able to classify a music genre. For this reason, I used GTZAN dataset as train data. In addition, I will focus on how the result is changed depending on the changing of parameters. It will be the key information for future work.

## II. Related works

The earliest work on paper [2] was made using a music genre classification model using Convolution Neural Network and other learning models. They are labeled with 10 genres of 30 second audio clips. They find the CNN has high accuracy to classify the music genre compared other models. They compare accuracy of deep-learning and accuracy of machine learning. It shows that accuracy is depends on the type of data and amount of data. Jessica Dias et.al.[4] set out music genre classification for music recommendation systems. It expected automatic genre classification to help users. They used CNN and GTZAN dataset for training. When comparing the accuracy with other algorithms which are SVM, KNN, it shows CNN had the most accuracy. Gabriel Gessle and Simon Åkesson's research [5] built music genre classification models using GTZAN dataset through CNN and other deep learning algorithms. They compared CNN and LSTM. As a result, CNN had more accuracy than LSTM.

## III. Music genre classification with CNN

As deep learning model, result is critical. Accuracy concludes the value of deep learning model. Despite this fact, there are not many papers about the accuracy. Most of papers focus on the importance of superior dataset for better accuracy or compare CNN and other models. There were no contexts about what and which parameter contributed on the accuracy and how we can improve accuracy with the tuning parameter. Therefore, in this paper, I will focus on what parameters affect the accuracy and how can we use this information to improve works.

### *A. Data*

To make a high accuracy model, a great amount of data is crucial. On the other hand, it is hard to find well organized data (large amounts of data, and recent data for reliability). In this condition, the GTZAN dataset satisfies all of the criteria. GTZAN is the dataset focused on music genre classification. All data in the set is collected between in 2000 and in 2001 from daily life. The GTZAN data set is composed of 10 genre music and 100 songs for each genre with 30 seconds length. For each song, it has chrome graph images (Fig.1). In previous works, paper [2] and [3], they succeeded in getting a highly accurate model. In the other hand, the other research paper that didn't know used GTZAN data set, that generate dataset their own, the method was similar with GTZAN data set.

Thus, I used the GTZAN dataset. In Particular, it was a music feature data in the dataset. There is two type of feature data file. One is analyzed for 3 seconds, and other is analyzed for 30 seconds. This music's feature files show each song's chroma mean, root mean square mean and extra. 3 seconds feature file has about 9000 rows and 30 seconds feature file has

979-8-3503-1327-7/23/$31.00 ©2023 IEEE

about 1000 rows. For the more data training, I use 3 seconds features data (Fig.2).

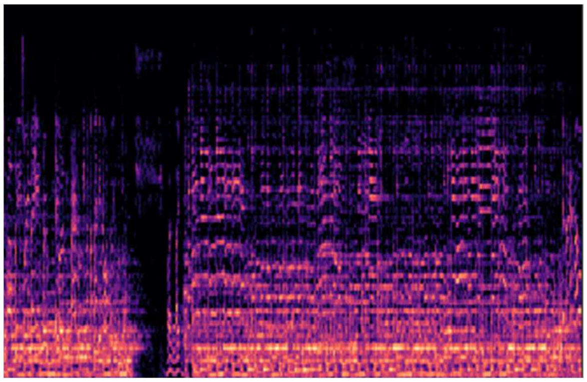
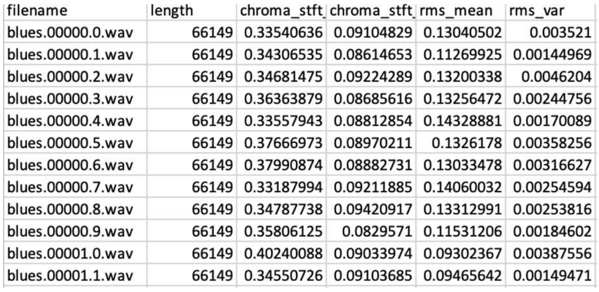


FIGURE 1. Example of GTZAN Chroma Graph



| filename | length | chroma_stft_ | chroma_stft_ | rms_mean | rms_var |
|---|---|---|---|---|---|
| blues.00000.0.wav | 66149 | 0.33540636 | 0.09104829 | 0.13040502 | 0.003521 |
| blues.00000.1.wav | 66149 | 0.34306535 | 0.08614653 | 0.11269925 | 0.00144969 |
| blues.00000.2.wav | 66149 | 0.34681475 | 0.09224289 | 0.13200338 | 0.0046204 |
| blues.00000.3.wav | 66149 | 0.36363879 | 0.08685616 | 0.13256472 | 0.00244756 |
| blues.00000.4.wav | 66149 | 0.33557943 | 0.08812854 | 0.14328881 | 0.00170089 |
| blues.00000.5.wav | 66149 | 0.37666973 | 0.08970211 | 0.1326178 | 0.00358256 |
| blues.00000.6.wav | 66149 | 0.37990874 | 0.08882731 | 0.13033478 | 0.00316627 |
| blues.00000.7.wav | 66149 | 0.33187994 | 0.09211885 | 0.14060032 | 0.00254594 |
| blues.00000.8.wav | 66149 | 0.34787738 | 0.09420917 | 0.13312991 | 0.00253816 |
| blues.00000.9.wav | 66149 | 0.35806125 | 0.0829571 | 0.11531206 | 0.00184602 |
| blues.00001.0.wav | 66149 | 0.40240088 | 0.09033974 | 0.09302367 | 0.00387556 |
| blues.00001.1.wav | 66149 | 0.34550726 | 0.09103685 | 0.09465642 | 0.00149471 |

FIGURE 2. Part of GTZAN 3-sec Features

### B. *CNN Model*

GTZAN analyzed songs through by mean of roll-off, mean of zero crossing rate, and etc. For basic information of the song, each column of each song shows the length of the song, and all songs had same length. It is unnecessary information, so it removed. Similar to length information, I deleted the same and basic information (length, index, and label).

To build the model, it required to refer to the previous works for high accuracy model. For high accuracy model case, it mostly composition of CNN and LSTM or CNN and other deep learning algorithm. I focus on finding the similarity among the high accuracy result model. However, since the goal of the paper is evaluated CNN, the model should be built by CNN. Thus, building model process is based on this paper [2]. Model is built by CNN. It has 3 convolutional layers with 64 filters and relu activation. Output of the work is 10 for classifying 10 music genres (Table 1).

In the blow, Figure 3 shows the whole history of validation and train loss. In the graphs, there were no noticeable changes. It didn't go out of range. Even if the epochs number increases, the loss remains. It means the number of epochs is not deeply related to the accuracy.

The table in the below shows what parameters give an impact on the accuracy. I did 5 trials for each tuning parameter for reliability. Each table has a changing variable which are epochs and batch size.

TABLE I. CNN 1D MODEL SUMMARY

| Layer(type) | Output shape | Param # |
|---|---|---|
| Conv1d | (None, 49, 64) | 704 |
| Max_pooling1d | (None, 24, 64) | 0 |
| Conv1d_1 | (None, 15, 64) | 41024 |
| Max_pooling1d_1 | (None, 7, 64) | 0 |
| Conv1d_2 | (None, 3, 64) | 20544 |
| Max_pooling1d_2 | (None, 1, 64) | 0 |
| flatten | (None, 64) | 0 |
| Dense | (None,32) | 2080 |
| Dense_1 | (None, 20) | 660 |
| Dense_2 | (None, 10) | 210 |
| Total params: 65,222<br>Trainable params: 65,222<br>Nontrainable params: 0 | | |

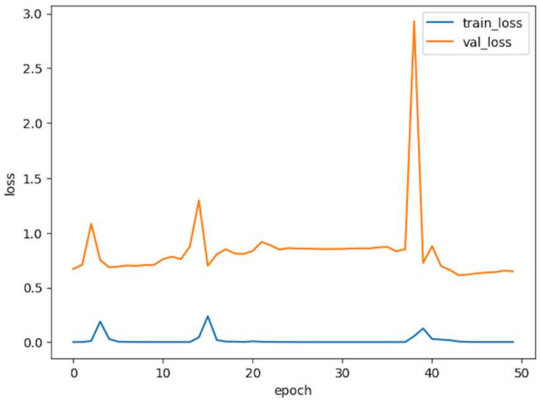


FIGURE 3. Validation loss (accuracy 83.20%)

TABLE II. TUNING PARAMETER (VALIDATION_SPLIT 0.5), EPOCHS, BATCH SIZE

| Independent parameter | Tuning parameter | Average accuracy |
|---|---|---|
| 3-layer, filter 32, relu activation, kernel size 3, dense 32,20,10 validation split = 0.5 | Epochs 10, batch size 10 | 70.98% |
| | Epochs 50, batch size 10 | 74.96% |
| | Epochs 10, batch size 50 | 75.69% |
| | Epochs 50, batch size 50 | 75.60% |

TABLE III. TUNING PARAMETER (VALIDATION_SPLIT 0.1), EPOCHS, BATCH SIZE

| Independent parameter | Tuning parameter | Average accuracy |
|---|---|---|
| 3-layer, filter 32, relu activation, kernel size 3, dense 32,20,10 validation split = 0.1 | Epochs 10, batch size 10 | 77.43% |
| | Epochs 50, batch size 10 | 77.88% |
| | Epoch 10, batch size 50 | 80.36% |
| | Epoch 50, batch size 50 | 80.30% |

TABLE IV. TUNING PARAMETER (VALIDATION_SPLIT 0.1, FILTER SIZE 64), EPOCHS, BATCH SIZE

| Independent parameter | Tuning parameter | Average accuracy |
|---|---|---|
| 3-layer, filter 64, relu activation, kernel size 3, dense 32,20,10 validation split = 0.1 | Epochs 10, batch size 10 | 82.35% |
| | Epochs 50, batch size 10 | 82.20% |
| | Epochs 10, batch size 50 | 83.41% |
| | Epochs 50, batch size 50 | 83.67% |

TABLE V. TUNING PARAMETER (VALIDATION_SPLIT 0.1, FILTER SIZE 64, KERNEL SIZE 10), EPOCHS, BATCH SIZE

| Independent parameter | Tuning parameter | Average accuracy |
|---|---|---|
| 3-layer, filter 64, relu activation, kernel size 10, dense 32,20,10 validation split = 0.1 | Epochs 10, batch size 10 | 83.02% |
| | Epochs 50, batch size 10 | 82.32% |
| | Epochs 10, batch size 50 | 84.51% |
| | Epochs 50, batch size 50 | 84.41% |

TABLE VI. TUNING PARAMETER (VALIDATION_SPLIT 0.5, FILTER SIZE 64, KERNEL SIZE 3), EPOCHS, BATCH SIZE

| Independent parameter | Tuning parameter | Average accuracy |
|---|---|---|
| 3-layer, filter 64, relu activation, kernel size 3, dense 32,20,10 validation split = 0.5 | Epochs 10, batch size 10 | 76.31% |
| | Epochs 50, batch size 10 | 76.76% |
| | Epochs 100, batch size 10 | 77.40% |
| | Epochs 10, batch size 50 | 78.25% |
| | Epochs 50, batch size 50 | 78.16% |
| | Epochs 100, batch size 50 | 78.04% |

The 1D CNN model's result is 85.6 percent. It is not on the table, cause the tables show average. 85.6% is maximum value.

Based on the tables, one of the factors of accuracy is estimated as a validation split. When the validation split changed to 0.1 from 0.5, accuracy showed a big gap. Another factor of accuracy is presumed filter size. It improved the portion of accuracy. It did not have an impact like validation split but it gave some effect on the result when validation split is changed 0.1 from 0.5. Epochs did not affect the accuracy rate. When average accuracy is under 75%, it seems like it most impacts the result. However, following table number 6, epochs did not influence the average accuracy. Among epochs 10, 50 and 100, there was no big difference. The biggest difference was 0.64. Though, after the accuracy increased, epochs could not give ideal change. Overall, I could observe, when batch size is 50, it shows higher accuracy compared to when batch size was another number. Nevertheless these results, that is tuning parameters for CNN indeed about the music(sound) learning. This result cannot be convinced these parameter conditions work for every deep learning algorithm even if it is CNN.

### C. *Reflection*

The limitation of the model is it is not available in real life. To get results, the model has to learn the data. It learns by 3 second features, if people want to use this model, they need a similar data type that the model learns for high accuracy. On the other hand, to analyze data like GTZAN analysis, it is a professional music expert's field. In addition, I do not have much professional knowledge of music. Thus, it is hard to use this model right now. If the model did not use 3 features of GTZAN, it would be available in real life. If a model learns the wave frequency of a song to classify the music genre, then it can be used as a music genre classification model. Later work, I wish to experiment again using another dataset.

## IV. CONCLUSION AND FUTURE WORKS

In this paper, I looked up CNN genre classification model training by GTZAN dataset. It focused on discovering reasonable parameter settings for high accuracy. Compared with other papers, I am also able to come up with the amount of data that can be the most crucial factor of accuracy.

Next step can be about music generation. It can be fundamental learning of the model that shows the percentage that the song is written by AI or Human. Thus, it can contribute to the decline of the AI abuse problem and raise awareness of copyright. Furthermore, the future work will be a comparison between the songs. In the next step, it can build another model that can compare the same genre song and determine the similarity between two songs. I will increase the accuracy based on the tuning result.

## REFERENCES

[1] Yoonhee.Jang, Analysis features of famous k-pop song by Librosa, Korea Computer Congress 2023

[2] Derek A.Huang, Arianna A.Serafini, Eli J.Pugh, Music Genre Classification, Cs229 Final report, Stanford University, pp.1-4.

[3] Ndiatenda Ndou, Ritesh Ajoodha, Ashwini Jadhav, Music Genre Classification: A Review Of Deep-Learning and Traditional Machine-Learning Approaches, IEEE 2021, pp.2-5.

[4] Jessica Dias, Hrutvik Deshmukh, Vaishak Pillai, Ashok Shah, Music Genre Classification Recommendation System using CNN, SSRN, 2022

[5] Gabriel Gessle and Simon Åkesson, A comparative analysis of CNN and LSTM for music genre classification, Degree project in project in Technoloy, first cycle, 15 credits stokholm, Swedem 2019, pp.1-10.

[6] GTZAN.https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification