Music Genre Classification using CNN

Simen Fredheim Aronsen

ABSTRACT

Music Genre classification is an inportaint feld in the music streaming industry where the user expects the application to choose and sort the songs in the different playlists by their respective genres. To achieve this there can be used multiple methods and the most common one to use machine learning (ML) methods that can be trained to classify the right genre and place them in the respective folders and playlists. Some of the common methods for acchieving this is by using common ML algorithms like *k*-NN (K-Nearest Neighbors and SVM (Support Vector Machine) but recent papers and reseach have concluded that Convolutional Neural Network (CNN) models have been able to acchieve good results.

1 INTRODUCTION AND RELATED WORK

Being able to sort different genres can be a hard task for any person, especially considering all the metadata that consists of a music streaming platform database. Reviewing hundreds of songs to find the one the user wants to listen to and with no knowledge of what genre the song belongs to could make a bad user experience. The music streaming platform has to manage different playlists that the user set up for example and in these libraries of playlists, there could be a number of different genres that the system has to keep track of. In our project, we use a CNN model and input our data to train the model and test its accuracy. We use a dataset that contains data features extracted from waveform files, it also contains image files that can be used to input the model. In a project, [6] they try a different amount of classification algorithms ranging from RBF kernel support vector machine, *K*-NN, and CNN. They used data from the same dataset that we are using and they experimented with the raw amplitude data and the image files consisting of Mel Spectrograms of the raw data. They experienced better accuracy in the CNN model using the image files as input and they even state that the accuracy surpasses human accuracy when it comes to classifying genres. In a project[5], they show how to analyze the different aspects of an audio file using different python libraries and this is a good starting point for understanding the nature of the data we are using for this project.

2 METHODOLOGY

The way we go forward to develop our CNN in this project is to use the feature dataset that comes in CSV format and use this data as input. The model is an interpretation of the AlexNet [1] framework in CNN modeling. The model consists of an input layer that uses the Relu activation function and has a filter size of 512. It passes through 3 hidden convolution layers with 256, 128, and 64 in filter size respectively. Then an output layer that consists of the softmax activation function normalizes the output and makes a probability distribution over the predicted 10 output classes. We use Tensorflow with Keras to implement the model. The loss is calculated with the Sparse Categorical Crossentropy function to measure how accurate the model is during training. We also use dropout to prevent overfitting. Overfitting happens when the model performs worse on previously unseen data. This is tested on the test dataset.

3 EXPERIMENTS AND DISCUSSION

3.1 Dataset

The dataset we use in the project is a public dataset. The GTZAN Dataset [2] is a dataset that contains samples of audio files from 10 genres with 100 audio files each and image data of Mel Spectrograms generated from the audio files. It also holds two .csv files containing raw data of the features extracted from the audio files. The audio has been collected from a variety of sources like CDs, radio, and microphone recordings among others. One file with each audio track being 30 seconds long and the other file with audio tracks 3 seconds long. According to the creators of the dataset, the inspiration for making it was to find out what an audio file is and how it looks. The purpose was also to find out if you can perform EDA(Exploratory Data Analysis) on the dataset from extracted features. Many projects have been conducted with the use of this dataset and some of them are the inspiration for this project.

3.2 Experimental Setup

We start by implementing the dataset and exploring it to understand its nature. We use the "feature3sec.CSV" dataset. When viewing it we can see that it holds 60 columns with a variety of data extracted from the audio files like mean and filename. The dataset has 9990 rows and the nature of datatypes is mostly numeric float64 that are good for machine learning models. However, we can see that some of the data are objects like the "filename" column. We remove this column for simplicity since we don't need it. We have to do further adjustments to the dataset. We want to focus on the "label" column. The nature of this data is an object or categorical data. We use the Label Encoder class from Python to convert this into numeric data for the model. We want to explore some of the audio files and choose to analyze a file using the Librosa python library for analyzing audio [3]. We use the IPython library [4] for playing a song from the dataset and after that, we visualize it in the waveform format. Figure 1 shows an audio file displayed in the waveform format. Figure 2 shows the same file displayed in the spectrogram format which is a different form of visualizing the audio file. The x-axis is the clip represented in time and the y-axis shows the frequencies from 0 to 10kHZ. After the preprocessing of the data is complete, we split the data into training and testing datasets. The number of epochs that are used is set to 600 and the optimizer used is the "adam" optimizer.

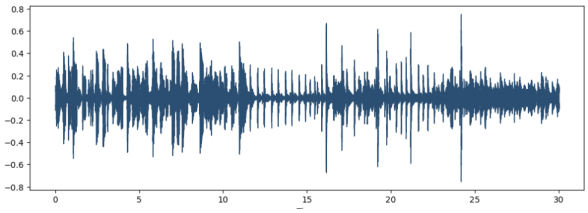

Figure 1: Waveform

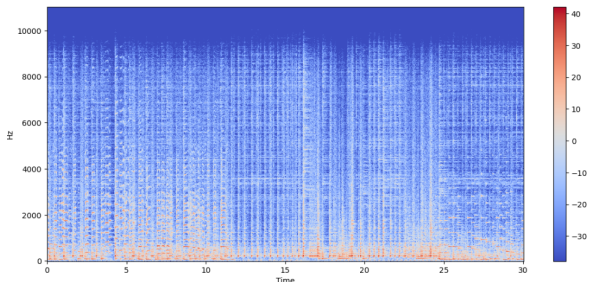

Figure 2: Spectrogram

3.3 Discussion

When we evaluate if the model is ready for prediction, we want to look at how it performed during the training process. We test this using our test data with a batch size of 128 and get the results that are represented in table 1.

test loss	66.4
test accuracy	92

Table 1: Model evaluation

From these results, we can conclude that the model has good accuracy in predicting the various music genres

4 CONCLUSION

We have learned how to implement a CNN for music genre classification in this project. We now know how to make a simple CNN using either the LeNet or AlexNet structure. Using Tensorflow with Keras was a good way of implementing the model and can be expanded on due to it being an intuitive way of implementing the model. The initial goal was to make a model that could predict music genres and to make the possibility for the model to recognize if two songs were of similar genres or not. This feature of the model was not implemented in the project for simplicity but the possibility is there to implement this in a future project. Neural networks are a good way to implement a machine learning algorithm for this kind of problem and if we look at what companies and applications use today, we understand that this approach is the most recommended approach if we should recommend an approach for implementing such a system. The possibilities are many and the company can develop systems like recommendation systems and the like from the knowledge gained from this approach. We learned much about what an audio file is by conducting this study. We learned about how to visualize an audio file and how to explore the audio files using premade tools in Python.

REFERENCES

[1] [n. d.]. [1803.01164] The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches. https://arxiv.org/abs/1803.01164. (Accessed on 11/20/2022).

[2] [n. d.]. GTZAN Dataset - Music Genre Classification | Kaggle. https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification. (Accessed on 11/18/2022).

[3] [n. d.]. librosa — librosa 0.9.2 documentation. https://librosa.org/doc/latest/index.html. (Accessed on 11/20/2022).

[4] [n. d.]. Module: display — IPython 3.2.1 documentation. https://ipython.org/ipython-doc/stable/api/generated/IPython.display.html#IPython.display.Audio. (Accessed on 11/20/2022).

[5] [n. d.]. Music Genre Classification with Python | by Parul Pandey | Towards Data Science. https://towardsdatascience.com/music-genre-classification-with-python-c714d032f0d8. (Accessed on 11/09/2022).

[6] Derek A Huang, Arianna A Serafini, and Eli J Pugh. 2018. Music Genre Classification. *CS229 Stanford* (2018).