mEtRoBoOm
Final Report

Predrag Muratovic and Armaan Lalani
December 6, 2020
ECE324
2000 Words

Introduction

The overarching goal of this project is to develop a neural network capable of classifying the genre of an inputted mp3 file. The inputted mp3 file will first have to be converted to mel-spectrogram which will then be passed through a neural network to predict the genre (hip-hop, pop, r&b, rock, latin, edm, and country). [1]

This project is extremely interesting because it essentially provides an additional method to classify songs because Spotify for example classifies based on features of the songs (eg. scales, energy, acoustics, etc.). [2] A neural network is appropriate for this task because it will be able to recognize distinct patterns in mel-spectrograms shared by songs of the same genre.

Illustration

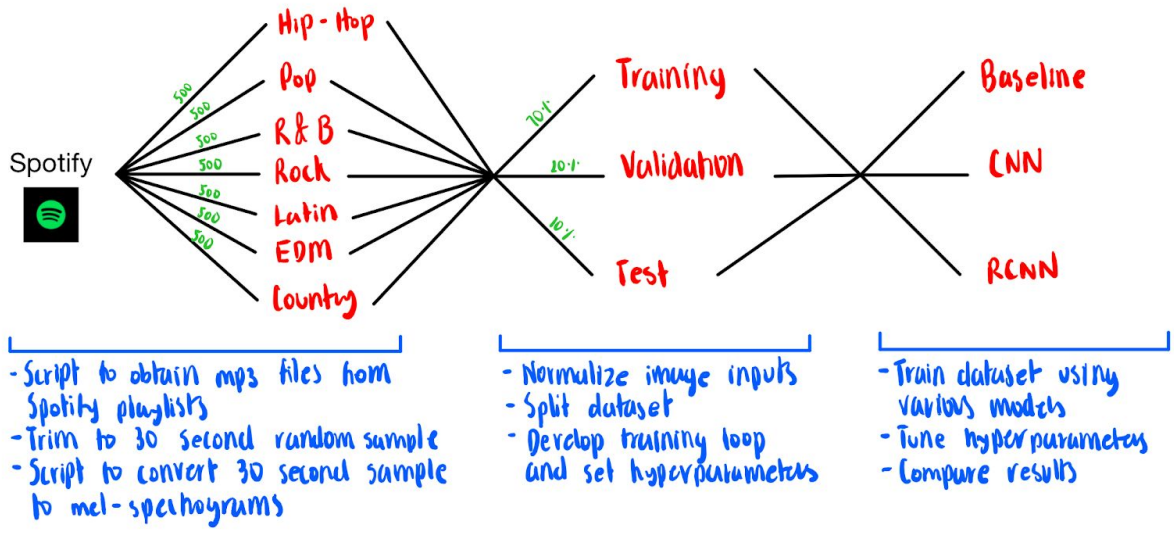

Figure 1: an overall representation of the project idea

Background & Related Work

There is similar work that has been undergone by both music streaming platforms as well as online research articles completing similar tasks. An article on towardsdatascience.com outlines the process of classifying music genres with the goal of creating playlists based on the information outputted from the model. Initially, audio files are used to create mel-spectrograms, which convert audio files to the frequency domain using Fourier transforms.

Mel-spectrograms are extremely important when conducting auditory analysis as it essentially extracts all of the most important features of an audio clip into a singular image, where each pixel represents something specific about the audio. This process is accomplished through a series of transforms including:

- A Fourier transform of the signal over numerous equally-sized windows

- Split the entire frequency spectrum into many evenly-distributed frequencies, where the frequencies '*sound'* equally distanced to one another based on human hearing.
- At this point one has a spectrogram; thus, to obtain a mel-spectrogram, one transforms on frequency spectrum into a mel spectrum.

Before finalizing this topic, one very important aspect of the background we had to consider is that music genre classification is extremely subjective; there is ambiguity in deciding exactly what genre some songs may belong to. Music genre subjectivity is extremely apparent in the genres of hip-hop, r&b, and pop which is something we expected to see in the results of the project.

Data and Data Processing

The process of data collection and processing can be explained based on the following steps:

1. We chose some of the most popular Spotify playlists belonging to the 7 genres of interest in order to develop a dataset of 350 samples per genre (this was later increased to 500 samples). Although songs can be found in different playlists, duplicates were removed since folders were merged together.
2. We utilized the Github repository 'Spotify to to MP3 - Python' by JayChen35 [3] which required the unique playlist uri in order to download the associated mp3 files. This was accomplished by using the youtube_dl package. (Note: we ensured the mp3 files being downloaded were the audio forms of the song rather than the music videos since music videos can have cinematic elements to them)
3. A script was written which chose a random 30 second sample of the mp3 files and saved the new versions.
4. The 30 second samples were then converted to mel-spectrograms using the Librosa package (see above section for details on spectrograms). The mel-spectrograms were normalized prior to training [4].
5. The training/validation/test split used was 70%, 20%, and 10% respectively.

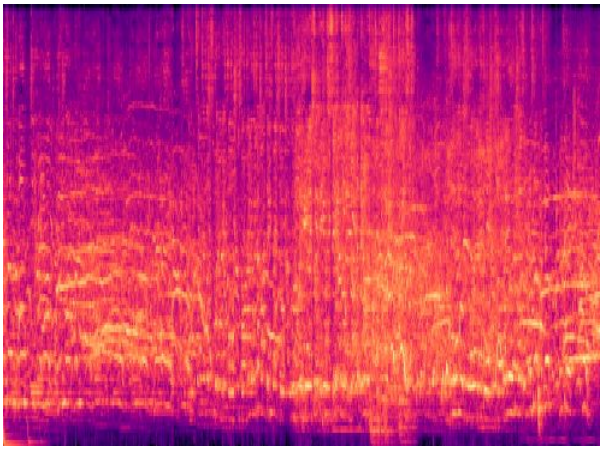

Figure 2: an example of a hip-hop mel-spectrogram

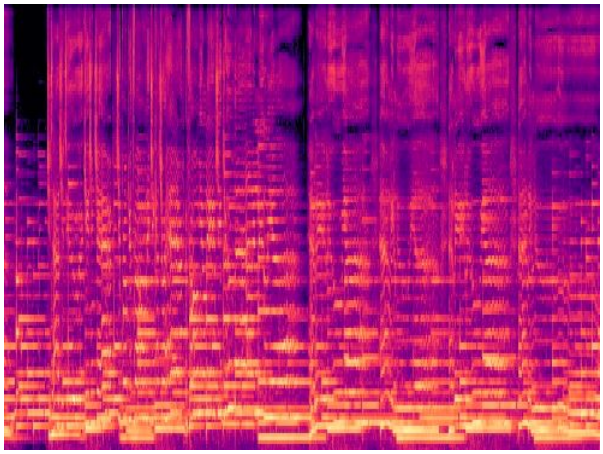

Figure 3: an example of a rock mel-spectrogram

Architecture

There were two primary model architectures that were used in addition to the baseline model which is further described below. The first model we used was a CNN model and the second model we used was a Convolutional Recurrent Neural Network (CRNN) which combines architecture of both RNN's and CNN's. Using CRNN's for a variety of music and audio related machine learning problems is a very common practice as it does very well in generalizing musical styles. [5] The basis of the model architectures is described and visualized below:

CNN: Input Image → 16 3x3 kernels → 32 3x3 kernels → 32 3x3 kernels → 32 3x3 kernels → linear layer (19488 → 4000) → linear layer (4000 → 500) → linear layer (500 → 7 class neurons)

CRNN: Input Image → 16 3x3 kernels → 32 3x3 kernels → 32 3x3 kernels → 32 3x3 kernels → GRU (input size: 23, hidden size: 500) → average along input size → linear layer (500 → 7 class neurons)

Note: each convolutional layer was followed by batch normalization and a 2x2 max pool for all three models

Baseline Model

The baseline model that was utilized was a relatively straightforward CNN consisting of 4 convolutional layers and 3 linear layers. The model architecture can be described as follows:

Baseline CNN: Input Image → 10 3x3 kernels → 10 3x3 kernels → 10 3x3 kernels → 10 3x3 kernels → linear layer (6090 → 1000) → linear layer (1000 → 200) → linear layer (200 → 7 class neurons)

Quantitative Results

Overall, the model performed reasonably well, however, we were initially hoping for results that were slightly better. Based on our analysis, the CRNN model was the best in predicting the genre of the test set with a peak test accuracy of approximately 79% compared to the CNN which achieved a peak test accuracy of approximately 73%. The test accuracy is the best measure of results because it represents how the model is able to generalize to data samples it has never seen before. Generally speaking, the higher the testing accuracy, the better the overall performance of the model and its application to real-world issues. The performance of the models is displayed in the graphs that follow.

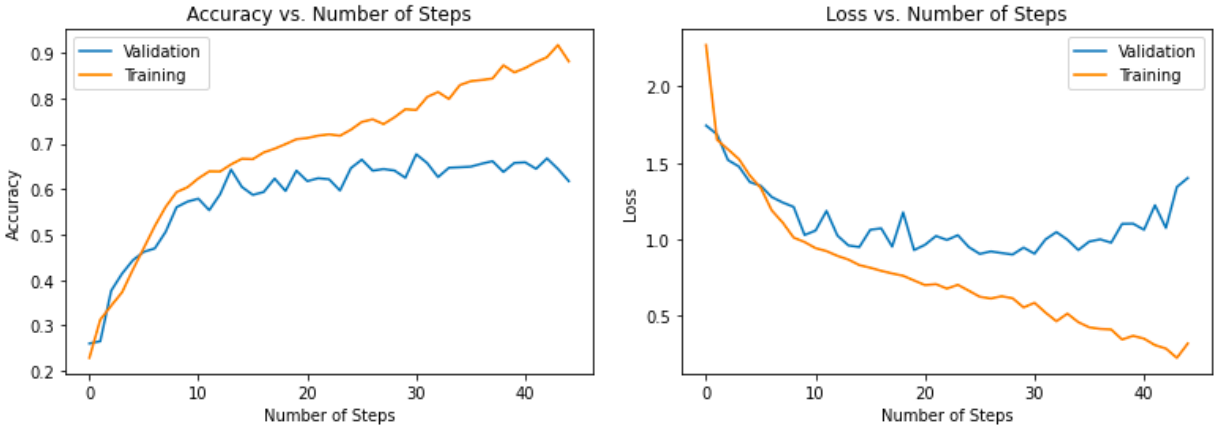

Figure 4: accuracy and loss plots for the CNN model using sigmoid activation on the convolutional layers and ReLu on the linear layers

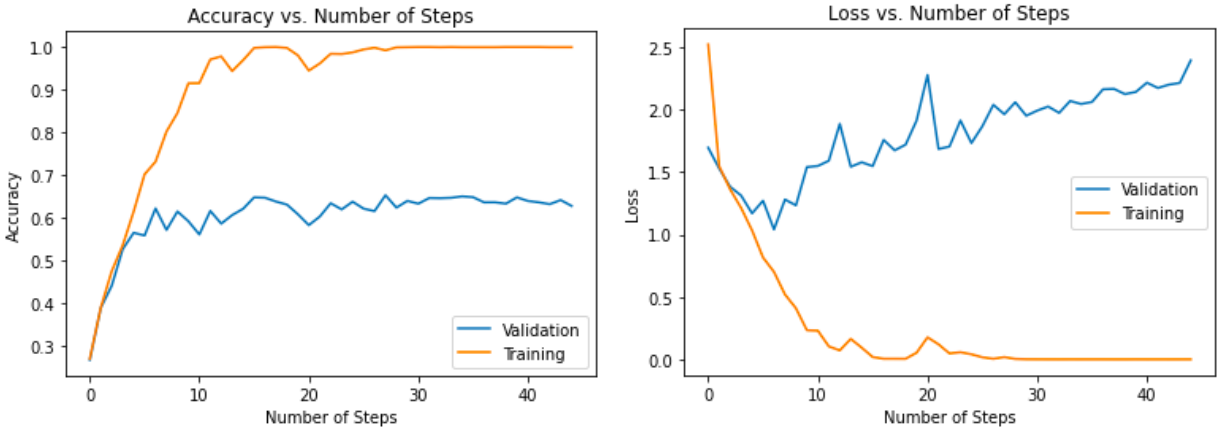

Figure 5: accuracy and loss plots for the CNN model using ReLu activation on both convolutional and linear layers

The hyperparameters for the above two graphs except for learning rate are as follows:

- Epochs: 45
- Learning Rate: 0.0001
- Batch Size: 32
- Loss Function: cross-entropy loss
- Optimizer: Adam

The test accuracies for the above two graphs were 73% and 70% respectively. As shown in the graphs, the first few epochs perform relatively well; the validation accuracy tends to follow the test accuracy within a reasonable range. However, as time moves on, the training accuracy begins to separate itself and the model fails to generalize well to both the validation and test data. Therefore, we hoped that the CRNN would do a better job in generalizing to the validation and testing dataset. The CRNN, while not significantly better, was more effective in generalizing to data it was not trained on as shown by the graphs below.

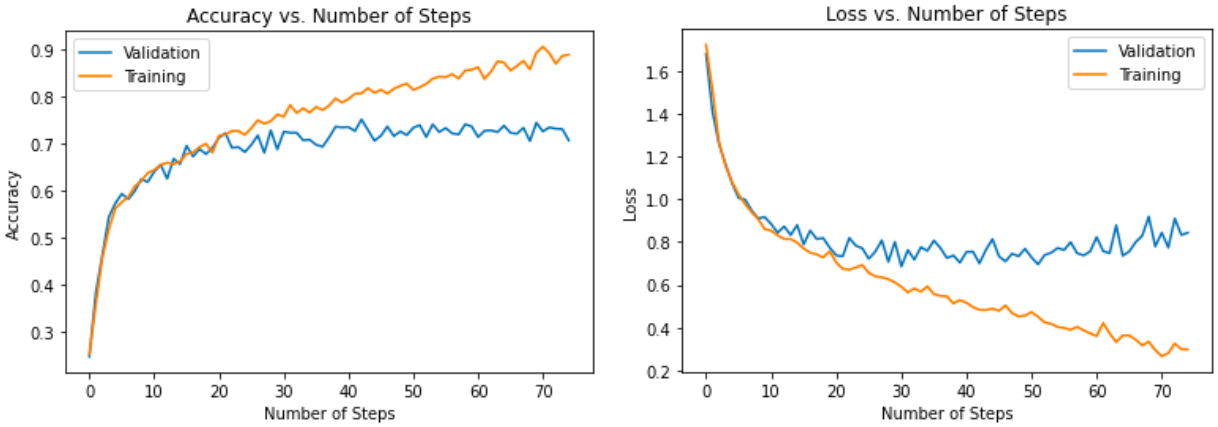

Figure 6: accuracy and loss plots for the CRNN model using a learning rate of 0.001

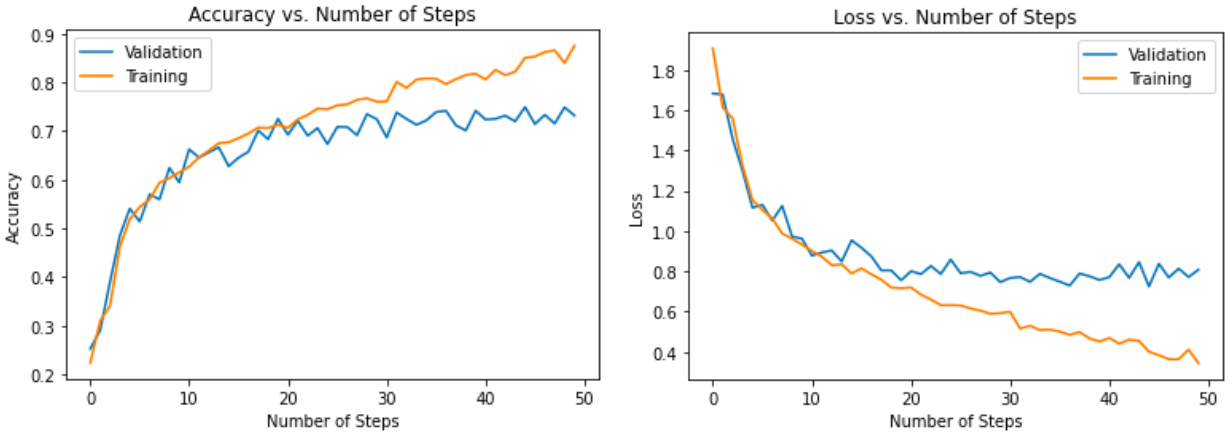

Figure 7: accuracy and loss plots for the CRNN model using a learning rate of 0.01

The hyperparameters for the above two graphs except for learning rate are as follows:

- Epochs: 75, 50
- Batch Size: 32
- Loss Function: cross-entropy loss
- Optimizer: Adam
- ReLu on convolutional layers

The test accuracies for the above two models were 79% and 76% respectively. As shown in the graph, the visual performance of the CRNN compared to the CNN is not significantly different; however, as evident by the test accuracy the CRNN does a stronger job of generalizing to data samples it was not trained on by utilizing its memory. The following displays the confusion matrix from figure 6.

		Predicted Class						
		Country	EDM	Hip-Hop	Latin	Pop	R&B	Rock
True Class	Country	44	1	0	2	0	2	2
	EDM	0	40	1	1	5	0	4
	Hip-Hop	0	2	32	1	2	13	1
	Latin	3	0	0	39	5	0	4
	Pop	0	7	0	7	39	1	1
	R&B	3	2	6	0	2	38	0
	Rock	3	2	0	1	0	0	45

Figure 8: confusion matrix of CRNN model

The use of a confusion matrix is another very important quantitative result for this particular project for the purpose of identifying the most common classes that were mistaken for each other. Prior to training a model, there were a number of classes which we had expected to be confused for each other and this should be reflected in the confusion matrix, which is further discussed in ‘Discussion and Learnings’.

Qualitative Results

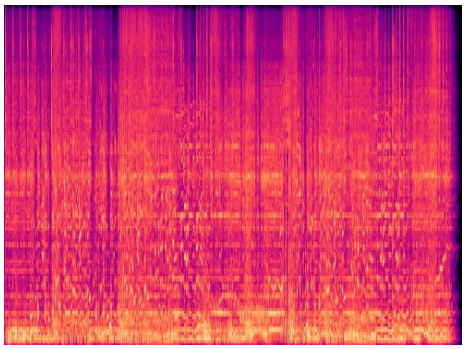

Figure 9: Mel-spectrogram of ‘Born to be Wild’ of class ‘Rock’

The above figure represents the spectrogram of a particular song that belongs to the Rock class. After running this particular example through the model, the associated output was a list of

probabilities, with the largest being located in position 6, which corresponds to the class Rock. This particular example represents the model working well.

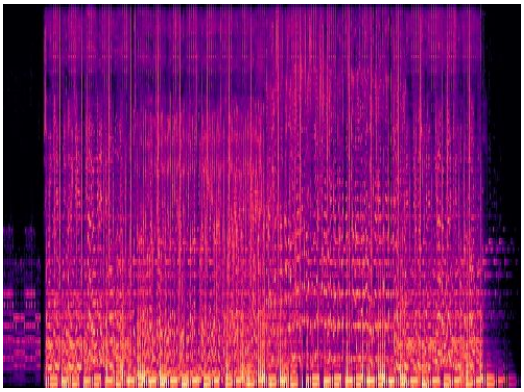

Figure 10: Mel-spectrogram of ‘Crew’ of class ‘R&B’

The above figure represents an ambiguous case described above; a song classified as R&B that can easily be classified as Rap. Crew in this case is listed under an R&B playlist, however, if you google Crew by GoldLink, it is listed under Hip-Hop/Rap. In this particular instance, the song was classified by the model as Hip-Hop which shows the interchangeability between the two classes.

Discussion

Overall, based on the results shown above, it can be concluded that the CRNN is the best performing model in comparison to the CNN and does a reasonable job of predicting music class genres. While 79% is not a very high test accuracy, there are reasonable explanations for this performance based on our quantitative results.

Firstly, the two classes that were confused with each other most often were R&B and Hip-Hop which is an extremely predictable result. It can be argued that various songs placed in either category can easily be swapped for each other depending on the listener. Therefore, the fact that roughly 20 test samples were incorrectly classified simply from R&B and Hip-Hop suggests the model performed well (thos 20 samples account for 6% for the entire test dataset alone). Secondly, another interesting aspect of the results is the behaviour of the Latin class. Latin music on Spotify is essentially classified as the lyrics being Spanish, however, the music, scales, acoustics, etc. from the MFCC might suggest another music genre. For example, several popular Latin songs on the Spotify playlists can easily also be classified as rap songs ‘delivered in Latin’ or pop songs ‘delivered in Latin’ for example.

Based on the above observations, and considering the ambiguity in classifying music genres to begin with, it can be concluded that the model performed well in classification. If another project was done similar to this, something we would do differently involves the method of data collection. In music, it is extremely common to have features, where the feature might be

a completely different genre from the song label itself (eg. a pop song with a rap feature). When selecting a random 30 second sample from the overall song, we could easily have selected a rap feature from a pop song without knowing. If a similar project was to be done, a more sophisticated method of clipping tracks would need to be established to avoid this.

Ethical Framework

To begin, our group is one of the important stakeholders in this project as we are the individuals who oversaw the project, gathered and processed data, as well as developed models to solve the problem at hand. Beneficence is one of the core principles that correspond to us as stakeholders because by doing this project we have developed numerous beneficial skills; for instance, we have learned how to write well-documented code, obtain and process data, as well as develop and evaluate a machine learning model for a real-world problem. These skills are very beneficial to us because they provide us with experience that will aid us in future scholastic and industry related projects. Furthermore, another relevant stakeholder and reflexive principal pair is the teaching team: Professor Rose and TA Shashank Saurav, and autonomy. The teaching team is tied to respect for autonomy because they enabled us to have full control of the project; thus, enabling us to have the freedom to make our own design decisions throughout the project. Additionally, the teaching team exhibits respect for autonomy because they were very supportive of our goals throughout the project and ensured we were keeping up with the related deadlines.

References

[1] "Visualized: Can we Quantify the Most Popular Music?," *Displayr*, 20-Nov-2020. [Online]. Available: https://www.displayr.com/most-popular-music/. [Accessed: 06-Dec-2020].

[2] N. Patch, "Meet the man classifying every genre of music on Spotify - all 1,387 of them," *thestar.com*, 14-Jan-2016. [Online]. Available: https://www.thestar.com/entertainment/2016/01/14/meet-the-man-classifying-every-genre-of-music-on-spotify-all-1387-of-them.html. [Accessed: 06-Dec-2020].

[3] JayChen35, "JayChen35/spotify-to-mp3-python," *GitHub*. [Online]. Available: https://github.com/JayChen35/spotify-to-mp3-python. [Accessed: 06-Dec-2020].

[4] McFee, Brain, et al. (December 2020), Librosa (Version 0.8.0), https://librosa.org

[5] Z. Nasrullah and Y. Zhao, NA, Toronto, ON, rep., 2019. (Music Artist Classification with Convolutional Recurrent Neural Networks)

Permissions

Post Video: No
Post Final Report: No
Post Final Source Code: No