Music Genre Classification

Jonah Rockey, Charlie May

Introduction:

With the evolution of technology, the ability to access music has evolved as well. Applications such as Spotify and Apple Music allow their users to have access to thousands of artists, genres, and songs allowing users to expand the musical preferences. But how exactly do these apps recommend music? Spotify, for example, creates playlists for their users such as daily mixes and discover weekly. These playlists are designed to contain songs, based off what a user has listened to, that Spotify believes the user would enjoy. However, these playlists don't always contain songs that resonate with the user's preferences, usually recommending songs that are in genres and by artists that the user has never listened to. This can happen for a number of reasons but usually the main one is the genre of music that is listened to by the user contains artists that they have never heard of. Music genres are categories of music that follow a certain set of conventions from the sounds to style or place/time of origin. This is what causes the playlists that are made to not be as accurate when it comes to recommending songs for the user. So, what if there was a better way to recommend songs for users that they would actually enjoy and listen to. This is where having a machine with this ability becomes useful. Having a machine that has the ability to learn what kind of music a specific user listens to the most and then accurately recommend songs, artists, and genres of music would become a more efficient way for people to be exposed to music that they would actually enjoy and open them up to new artists and genres.

Problem Description:

The problem that will be looked at is being able to define and classify specific songs, artists, and genres of music based on what a person listens to the most.

Data:

The data that was used for this project was from the GTZAN database. The data contained 1000 samples of songs. Each entry in the dataset contained a 30-second-long sample of the song with accompanying statistics and a visual representation that was called a Mel Spectrogram. The genres that the songs were taken from were blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock.

Methodology:

The methodology of this project was based around analyzing the Mel Spectrogram of each sample. The Mel Spectrogram is a visual representation that measures the various frequencies present in a song and at what volumes they are being detected. It utilizes the Mel Scale which is a logarithmic formula used to measure frequencies in a scale that is more comparable to how the human ear perceives sounds. Below is an example of a Mel Spectrogram from our dataset that represents a song sample from the metal genre.

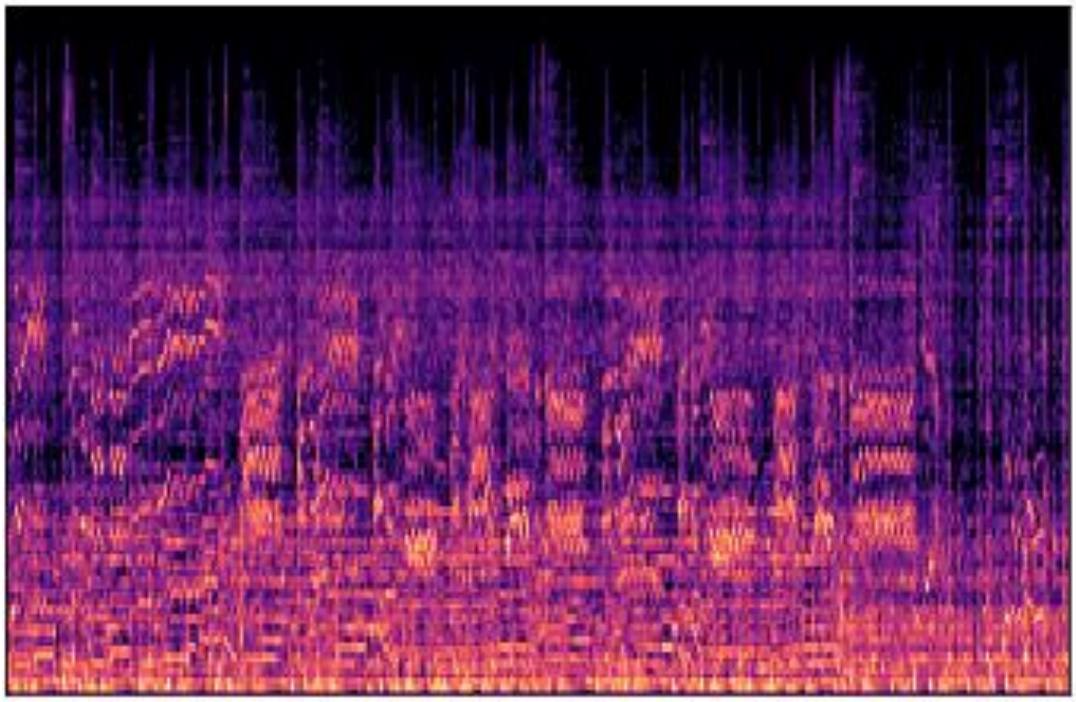

Figure 1: Mel Spectrogram

The x-axis in this visualization represents time, while the y-axis is frequency in Hz on a logarithmic scale. Additionally, the brightness of the graph represents the volume of that specific frequency. Essentially, this can be thought of as a continuous heat map showing the volume of various frequencies as the song progresses. In this project, we are using this image data to classify songs into genre. We chose to use Mel Spectrograms because they are one of the best ways to visually represent a song. We feel that the differences between different genres of music is evident when looking at the Mel Spectrographs. Ideally, the model can identify these differences and classify the images into genres with a high degree of accuracy.

The first step in building the model was data selection of the inputs. Each sample in our dataset is 30 seconds long, and there are 1000 samples total. After testing various sizes of inputs in our model, we decided to split each data entry in three parts. Therefore, each input into the model represents a 10-second sample of music. We did this because it was extremely beneficial for the model to have more data to train on. We felt that splitting the data into 10-second intervals did that sufficiently while still maintaining the connectivity of the image data, so patterns could still be detected by the model.

Next, we built a CNN model to classify images into a predicted genre. This model went through many different iterations with different aspects. Below is the final CNN model used in the project.

```
model = Sequential()
model.add(Conv2D(64, (3, 3), activation = "relu", input_shape = input_shape))
model.add(MaxPool2D((3, 3), strides=(2, 2), padding="same"))
model.add(BatchNormalization())

model.add(Conv2D(32, (3, 3), activation = "relu"))
model.add(MaxPool2D((3, 3), strides=(2, 2), padding="same"))
model.add(BatchNormalization())

model.add(Conv2D(32, (2, 2), activation = "relu"))
model.add(MaxPool2D((2, 2), strides=(2, 2), padding="same"))
model.add(BatchNormalization())

model.add(Conv2D(16, (1, 1), activation = "relu"))
model.add(MaxPool2D((1, 1), strides=(2, 2), padding="same"))
model.add(BatchNormalization())

model.add(Flatten())
model.add(Dense(128, activation="relu"))
model.add(Dropout(0.3))
model.add(Dense(64, activation="relu"))
model.add(Dropout(0.3))
model.add(Dense(10, activation="softmax"))
```

Figure 2: CNN model

This is a sequential model that contains three convolutional layers. Each layer utilizes max pooling and batch normalization. This was done in order to decrease overfitting and optimize training time of the model. Additionally, the CNN model has 3 dense layers and utilizes dropout for each of these. Finally, the model uses the softmax activation function to predict what genre the image most likely belongs to.

Results:

Overall, the model performed somewhat well on the data, but not exceedingly so. Below is a graph showing the model loss and accuracy of both the training and validation set.

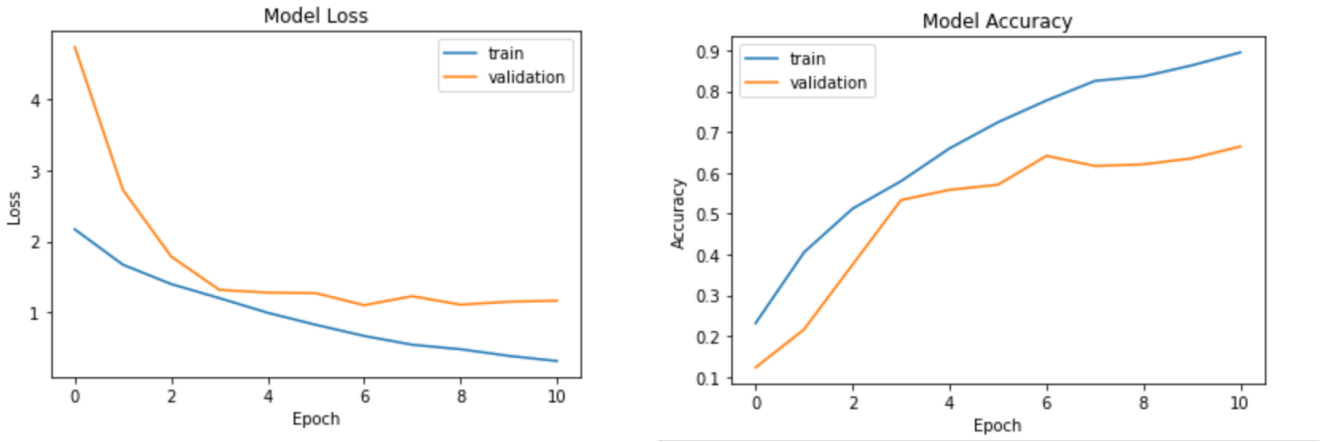

Figure 3: Model loss and accuracy

On the training data, the model loss steadily declined to near zero while the accuracy rose to greater than 0.9. For the validation set, the loss becomes somewhat stagnant at four epochs. The loss is about 1.3 after ten epochs. Additionally, the accuracy of the validation set was around 0.65 after 10 epochs. After training the model, we then used the test set to see how accurate its predictions of genre were. The figure below shows the overall accuracy as well as the class-wise accuracies of the model.

```
Overall Accuracy: 0.635
Class-wise accuracies for

blues samples: 0.5254
classical samples: 0.8519
country samples: 0.3692
disco samples: 0.4783
hiphop samples: 0.8429
jazz samples: 0.8085
metal samples: 0.9492
pop samples: 0.5484
reggae samples: 0.614
rock samples: 0.431
```

Figure 4: Accuracy of model

The CNN model achieved an overall accuracy of 0.635. This is not an exceedingly high accuracy, but it demonstrates that the model is still very capable of predicting the correct genre from the ten classes more than half the time. When we look at the class-wise accuracies, we see that the model does very well with some genres and badly with others. The model performed best when predicting samples of metal, classical, and hip-hop. We believe that these genres performed well because they are very distinct compared to the others. For example, classical music is a very different type of music than the other lyrical genres. These differences are also evident in the Mel Spectrograph. Therefore, the model was able to distinguish these differences and predict this genre better than others. On the other hand, the model performed worst on the genres of country, rock, and disco. We believe the reason for the low accuracy is that many of these genres are very similar and interact with each other in music today. A great deal of music exists on a spectrum in which techniques from many different genres are used. For example, it is often hard to draw the line between what is a pop song versus a rock song. Therefore, it can also be hard to differentiate between the Mel Spectrograms.

In conclusion, our project set out to analyze a visual representation of music samples in order to predict the music genre. This is a way of identifying the genre of music based solely on audio. We believe that this has interesting applications in today's environment of music streaming. Overall, our model achieved greater than a 60% accuracy, but struggled to differentiate between certain genres of music.

References:

Olteanu, Andrada. “GTZAN Dataset - Music Genre Classification.” *Kaggle*, 24 Mar. 2020, https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification.

Roberts, Leland. “Understanding the Mel Spectrogram.” *Medium*, Analytics Vidhya, 17 Aug. 2022, https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53.

Appendix:

Jonah: Worked on building the model, wrote methodology and results sections of report

Charles: Helped to build the model, wrote introduction, problem, and data sections of report