MUSIC GENRE CLASSIFICATION

Julie Karam, Sasha Liu,
Nathan DePiero, and Gavin Sabalewski

1| Introduction

We re-implemented an existing paper titled, **"Music Genre Classification using Machine Learning Techniques"** (Bahuleyan 2018). The paper's objective was to create a music classification model based on genre for the purpose of automatic organization of music libraries - "Being able to automatically classify and provide tags to the music present in a user's library, based on genre, would be beneficial for audio streaming services such as Spotify and iTunes" (Bahuleyan 2018).
The study utilized the Audio set data set, and they report an AUC value of 0.894 and a testing accuracy of 0.65 for an ensemble classifier which combines the two proposed approaches. We had a goal of implementing this model on another audio dataset and achieving similar performance.

Research paper link: Music Genre Classification using Machine Learning Techniques

2| Data

This dataset we chose to use is the **GTZAN dataset**, which is considered the "MNIST of sounds." The dataset contains a collection of **10 genres with 100 audio files each**. Additionally there are visual representations (spectrograms) for each audio file. The dataset contains a CSV file, that has a mean and variance computed over multiple features for each **30 second song.**

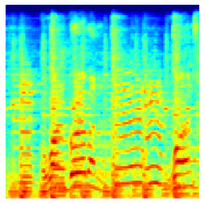
Genre Categories:

blues	jazz
classical	metal
country	pop
disco	reggae
hip hop	rock

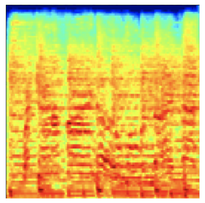
3| Preprocessing

We used **mel spectrograms** as inputs to the model. Mel spectrograms depict audio files in a 2-dimensional format that replicates how humans hear differences in frequencies, unlike a regular spectrogram. In order to process the 1000 30-second .wav files into mel spectrograms of 3 second snippets we created a script that used the **librosa library**. This resulted in a 2-d array which we repeated into 3 dimensions to input as a "color" image into the VGG-16 model. We chose to do this instead of converting the mel spectrogram to a .png image with a color to represent the different decibels values across the spectrogram because converting to an image would distort the data being represented by converting decibel strengths to colors.

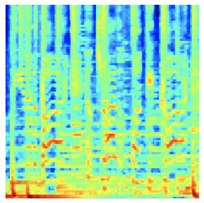
Spectrogram Image Examples:



blues spectrogram



metal spectrogram



pop spectrogram

4| Methodology

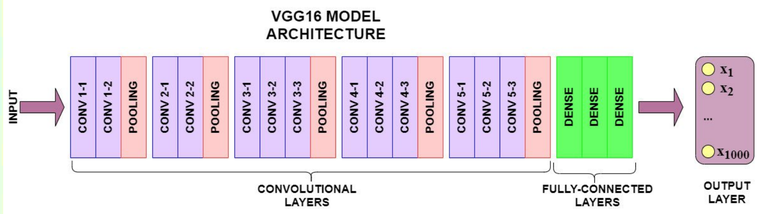


Figure 1. Model Architecture

The 3-d versions of the 2-d mel spectrograms are fed into a **VGG-16 transfer learning model** with pre-trained weights. This is a model (depicted above) that has several groups of convolution layers separated by max-pooling layers and the output of these is run through dense layers for classification. It is designed for image classification where pre-trained weights for the convolution layers that are trained on generic image data can be imported, and using these allows for faster training for different image datasets because the model starts with some feature recognition already learned. By converting the mel spectrogram into a 3-d tensor, it was able to be inputted into the VGG-16 model with default pre-trained weights. We modified the output of the VGG-16 from 1000 classes to 10 classes. This output uses a softmax activation. Thus, the index of the largest probability in this array is the classification of the song. We used **Cross Entropy** for our loss function and **Adam** with a **learning rate of 0.001** and a **weight decay of 0.0001** for our optimizer. The model was built using Pytorch and trained using Google Colab GPUs.

5| Metrics

We used two different metrics to measure the performance of our model: **Classification Cross-Entropy Loss and accuracy.**

6| Results

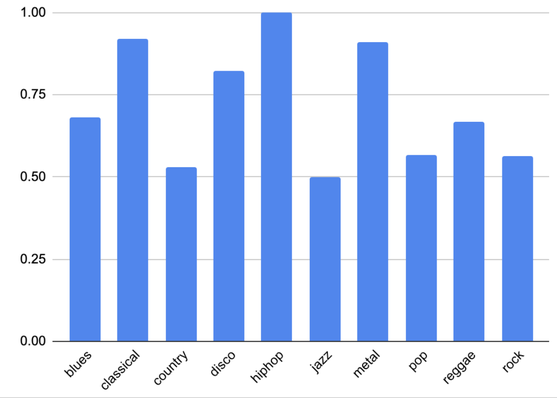


Figure 2. Percent correct by genre for one test batch

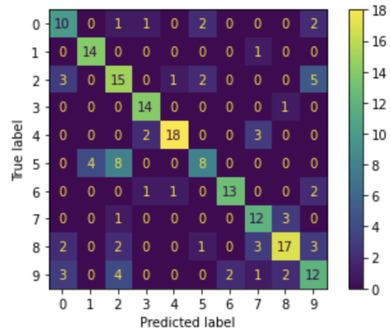


Figure 3. Confusion matrix for one test batch

After training and testing our model we were able to obtain average **accuracy of 0.75 and loss of 0.70**. The accuracy per genre can be seen in the figures above. This is comparable to the paper which obtained an accuracy of 0.64.

7| Discussion

A limitation of this model is that the dataset we used does not include all existing genres.. More robust genre classification may be necessary as music genres become more niche and music streaming platforms compete to give the best recommendations to their users. Future work may look to classify songs into more specific sub genres and may even use lyrics in conjunction with spectrograms for this purpose!

We encountered several challenges while working on our project. Our first challenge was becoming familiar with this new dataset. Audio files are something that we have not worked with before, so we spent a lot of time becoming familiar with this new data type. Additionally, we implemented our project in Pytorch, which none of us had prior experience with. We had to familiarize ourselves with this package, and apply our knowledge and skills from class to successfully implement this model in Pytorch.