**Music Classification and Statistical Analysis**

By Brandon Rasgaitis, Tyler Desharnais, Lucie Gaye, and Julien Toulon

**Abstract:**

In this project, we will study music data. We will try to train our artificial intelligence to classify songs by their specific genre. For this we will start by creating the artificial neural network. We will then look for a database containing twenty thousand songs and their spectrograms in order to train it to recognize and classify them. Once trained we will test our intelligence. Projects of this type already exist but we will try to improve them. Indeed there are many musical styles and some music cannot be classified in an absolute way. We can therefore try to improve the interface of analysis of the results. Thus, after training the dataset and configuring the Convolutional Neural Network model we will be using, our AI would be able to place the song’s spectrogram image on a graph or to give the influences drawn from each musical style and the nuances. The commercial benefit of this project could be for the artists to better classify their music and better fit into the current trends. It could also allow them to musically analyze the biggest commercial successes and understand what could lead them to success. By comparing the artist's music with the top of the charts, it could allow him to know if his music has a chance to appeal to the greatest number of people.

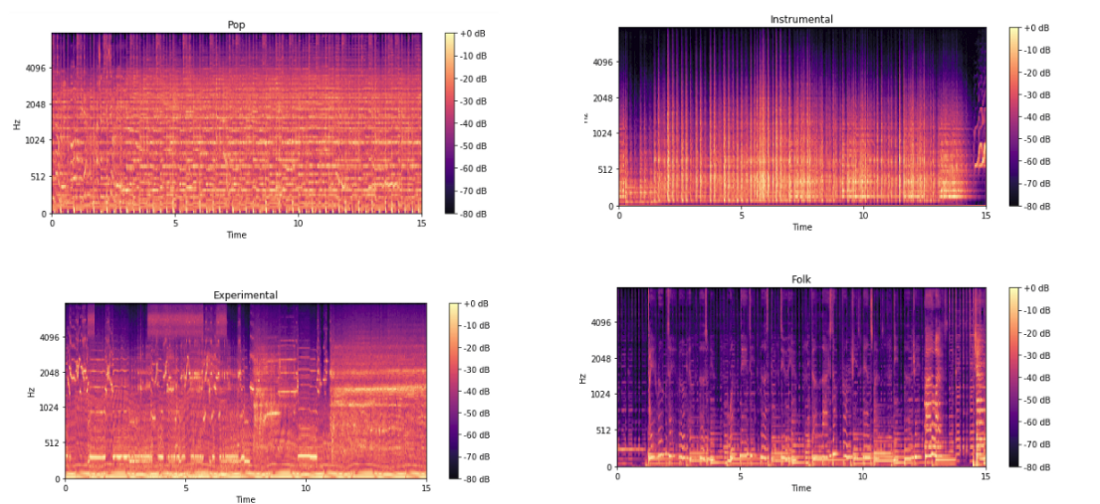
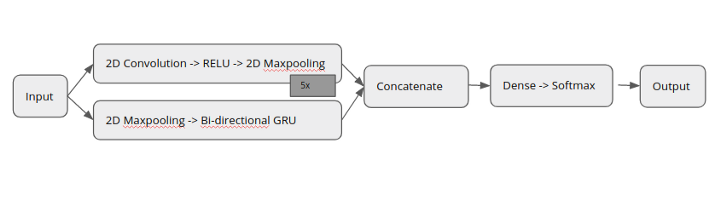
**Introduction:**

The overall problem being addressed in this project is the lack of nuance in music genre classification. Music is an artform that takes inspiration from many different technical aspects that make up the structure of a song. This makes the classification of particular songs to specific genres difficult as they can’t be classified in an absolute way. One of our solutions is to read specific data from a song’s spectrogram, which is an image detailing the specific aspects that design a song’s structure. The aspects that create the structure of said song include beat, tempo, decibel readings, length of time, danceability, frequency, etc. Improving the interface of genre classification to include statistical data of songs would allow artists and composers to better understand the technical aspects of how their song is designed, provide a tool to promote their music to a wider audience or better cater to a niche fan base, provide a tool to design and classify new genres by using specific data from different songs and their spectrograms, and finally it could aid in the commercial success of an artist and their music by detailing different statistics of a genre’s commercial success.

**Related Works:**

There are many other Music Genre Classification deep learning models that use the same techniques we use in order to classify and label each song. One such example, used by Arsh Chowdhry who posted their project on their blog post at [www.clairvoyant.ai](http://www.clairvoyant.ai) involves creating a Convolutional Neural Network model to classify music samples into different genres. Arsh also uses raw wave audio file images, spectrogram images, Spectral Rolloff graphs, chroma feature graphs, and zero crossing rate graphs to determine specific aspects that design the structure of a song. This data is then used for the creation of a music recommendation system that automatically classifies song genres for the user. This is where our project differs. We will be analyzing the results of the data that makes up the structure of a song and predict what genre it’s classified under. For example we take a rap song created by eminem and have our program predict the genre and form its spectrogram. From there our AI will look at a piece of data such as beat and predict how similar it is to other genre beats and will classify them and their accuracies. This will allow an artist to design their songs using different technical aspects from other genres and provide a more indepth way to classify their song. Instead of just labeling a song with the genre of rap, it will label it rap-metal for example as the beat has similarities with rock.

**Group Work Plan:**

* **Brandon:** 
  + Designing the Convolutional Neural Network Model. We will be using the Convolutional Neural Network algorithm as it provides the highest overall accuracy for projects similar to ours. We will use a batch size of 128, sparse\_categorical\_crossentorpy loss function, ‘adam’ as the optimizer, and 600 epochs. Our model will be a Sequential layer so that we can sequentially add layers because each layer will have one input and one output to form the entire network. Each layer will be a dense layer with a “relu” activation function, with the final layer utilizing the softmax activation function. We will also include four dropout layers of value 0.2 to mitigate overfitting of the model. Each person will have the responsibility of improving the accuracy of the model evaluation. Our goal is to have at least a ninety percent accuracy for our model evaluation.
* **Lucie:** 
  + Labeling the spectrogram and raw wav file￼ images with their appropriate song title and genre.
  + ex.)
  + This will require importing the final\_dataframe.csv data set into our CNN models that we created in hw3 to label images using the cifar100 dataset. The code will have to be modified to access and visualize the chromagram and spectrograms of various songs.
* **Julien:**
  + Creating the histogram and ROC-AUC graphs in order to test the accuracy of the training data.
  + Developing the Parallel CNN-RNN model. We need to use a RNN model in parallel with our CNN model because RNNs do a better job of determining time sensitive information which is useful as all songs have a time length. The parallel CNN-RNN model will pass the input spectrogram through both models and concatenate their output. It will then send the output through a series of dense layers with the final layer being a softmax activation function to perform the classification
  + ex.)  
      
      
      
      
      
      
      
    
* **Tai:**
  + Collection of data and the generation of spectrogram images. This involves reading in the file, final\_dataframe.csv, creating features, and changing dtypes. From there we need to split the data into training data, and test data. Explore the data to create charts, graphs, and images to visualize the data set. The data set will represent various traits of a song’s structure and popularity. The next step will be to check the distribution and make bins and then to check the bin counts and its variance. Then we can provide a statistical T-test for different variables of a song provided from the data set. The final steps are to process the entire data frame, extract it, and scale the data.