Music Genre classification and Song Recommendation System

Srinivas Yashvanth Valavala
Pranav Mahajan
Sri Venkata Sai Anoop Bulusu

***Abstract*—In this project, we developed a deep learning model using convolutional neural networks (CNNs) and transfer learning using residual CNN (RESNET 18) to classify the genre of musical audio samples. The model was trained on a dataset of musical audio samples, each labeled with a specific genre. Once the model was trained, we used it to classify the genres of a new set of musical audio samples. We then used the classification results to generate a custom playlist for the user, based on the distribution of genres they had been listening to. The results of our experiments showed that the CNN and residual CNN models were able to accurately classify the genres of musical audio samples, with a decent level of accuracy, demonstrating the effectiveness of using deep learning techniques, specifically CNNs and residual CNN, for genre classification of musical audio samples.**

I. Keywords

NumPy, Librosa, Pytorch, Mel Spectograms, Convolutional Neural Network (CNN), Resnet18, Residual networks, Deep learning, Music Recommendation system

II. Introduction

Music genre classification is an important task in the field of music information retrieval, which involves organizing musical works into categories based on their style, content, and other identifying characteristics. This allows for more efficient organization and retrieval of music and can help listeners and music industry professionals alike better understand and appreciate the vast array of musical genres and sub-genres that exist. In this project, we will explore various techniques for music genre classification and evaluate their effectiveness on a dataset of musical recordings. Our goal is to develop a model that can accurately classify a given piece of music into one of several predefined genres. By delving into the intricacies of music genre classification, we hope to gain a deeper understanding of this important task and contribute to the advancement of music information retrieval technology.

Convolutional neural networks (CNN) are a deep learning technique that is commonly used for image classification tasks and tasks that require pixel processing. In our case, we use it for the classification of music data by first converting them into an appropriate image format. CNNs are particularly well-suited for this task because they can automatically learn the important features of a piece of music, such as the tempo and the instruments used, and use this information to accurately classify the music into different genres. We also implemented transfer learning using ResNet18. ResNets solve the problem of the vanishing/exploding gradient by introducing Residual Blocks and can improve upon the accuracy of traditional CNNs.

By using these deep learning techniques, music genre classification can be performed more accurately and efficiently than with traditional methods. This allows for the generation of personalized playlists for users based on the distribution of genres they have been listening to. For example, if a user has been listening to a lot of rock music, a playlist consisting mostly of rock songs could be recommended to them. This provides a more tailored listening experience and can help users discover new music that they may enjoy.

III. Literature survey

In [1], Lin Feng et.al describes a proposed hybrid architecture for music genre classification using deep learning. The architecture consists of parallel convolutional neural network (CNN) and bidirectional recurrent neural network (Bi-RNN) blocks, which focus on spatial features and temporal frame orders, respectively. The outputs of the two blocks are fused into a single representation of musical signals and fed into a softmax function for classification. The authors argue that the paralleling network improves the robustness of feature extraction and that the addition of the Bi-RNN block is a supplement for CNNs. The experiments show that the proposed architecture improves the performance of music genre classification.

In [2], Shweta Koparde et.al describes a system for classifying music by genre using a deep learning technique called a convolutional neural network (CNN). The authors argue that music genre classification is a complex task in music information retrieval, but that machine learning models can solve these kinds of problems. They propose using a CNN to classify music into various genres and suggest that acoustic feature extraction is a crucial task in this process. The proposed system is trained on the GTZAN dataset. Overall, the abstract presents a system for classifying music by genre using a deep learning technique.

In [3], Michael Haggblade et.al describes a study investigating the use of various machine-learning algorithms for music genre classification. The algorithms studied include k-nearest neighbor (kNN), k-means, multi-class support vector machine (SVM), and neural networks. The study focuses on classifying four genres: classical, jazz, metal, and pop. The authors use Mel Frequency Cepstral Coefficients (MFCCs)

as features for the classification algorithms, as recommended by previous research in the field. The study also explores an extension of the classification task by mapping images to music genres using the Fourier-Mellin 2D transform and clustering the images with k-means. Overall, the study seeks to improve the performance of music genre classification using machine learning algorithms.

In [4], Qi He et.al describes a study that proposes a method for classifying digital music by genre using recurrent neural networks (RNNs) and attention. The method involves dividing the music into multiple local passages, extracting features from the passages, and using RNNs to learn temporal and semantic information about the music. The study collects 1920 MIDI files with genre labels from the internet to create a dataset for classification testing. The proposed method is evaluated in terms of its accuracy for classifying music by genre. Overall, the study seeks to develop a more effective method for classifying digital music by genre using RNNs and attention.

In [5], Leland Roberts talks about how Human perception of frequency is not linear. We are better at detecting differences in lower frequencies than in higher frequencies. For instance, we can easily tell the difference between 500 and 1000 Hz, but we would have a hard time noticing a difference between 10,000 and 10,500 Hz, even though the distance between the two pairs is the same. In 1937, Stevens, Volkmann, and Newmann proposed a unit of pitch that would make equal distances in pitch sound equally distant to the listener. This is called the Mel scale. To convert frequencies to the Mel scale, we perform a mathematical operation on them.

IV. Data Description

In this project, we will be using the GITZAN dataset, which contains 1000 musical recordings of 30 seconds each. The goal of this project is to develop a model that can accurately classify a given piece of music into one of these predefined genres.

A. About the data:

- The GTZAN dataset is the most commonly used public dataset for evaluating machine listening research in the area of music genre recognition (MGR). The files in the dataset were collected in 2000-2001 from various sources, including personal CDs and radio recordings, and they were chosen to represent a wide range of recording conditions.

- The dataset is evenly distributed among 10 different genres, including blues, classical, country, disco, hip hop, jazz, reggae, rock, metal, and pop.

The audio waveforms for one musical recording from each genre looks like the following:

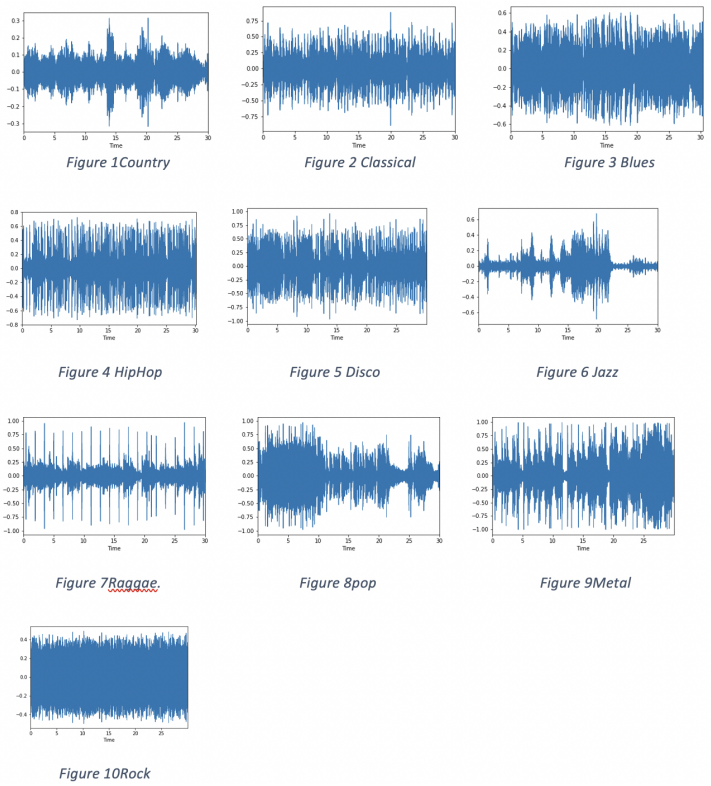

Fig 1: Audio waveforms for one sample in each genre

The dataset can be found on Kaggle[6]

V. Data Prepossessing

Before feeding the audio waveforms as input to the models, we have applied some pre-processing techniques and converted them into spectrograms. We have employed two different techniques of representing the audio signals in spectrograms. The two techniques we have used are:

A. Mel Spectogram

- The Mel Scale is a non-linear scale used to represent the perceived pitch of a sound.

- Studies have shown that humans are better at detecting differences in lower frequencies than higher frequencies on this scale.

- The Mel Spectrogram is a type of spectrogram (a visual representation of the spectrum of frequencies in a sound) that uses the Mel Scale to measure frequencies.

- Despite the complex concepts behind it, the Mel Spectrogram can be easily implemented in a few lines of code.

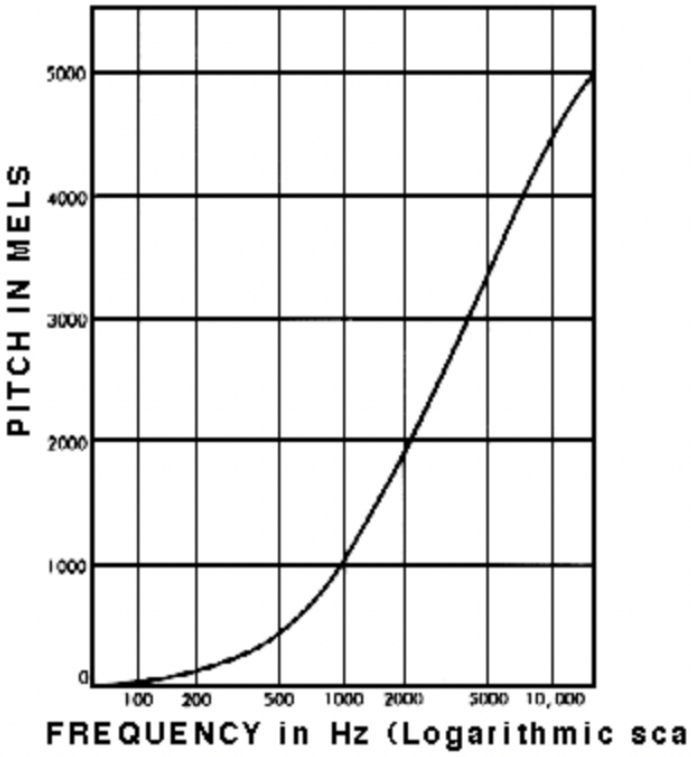

Fig 2: Frequency to Mel conversion

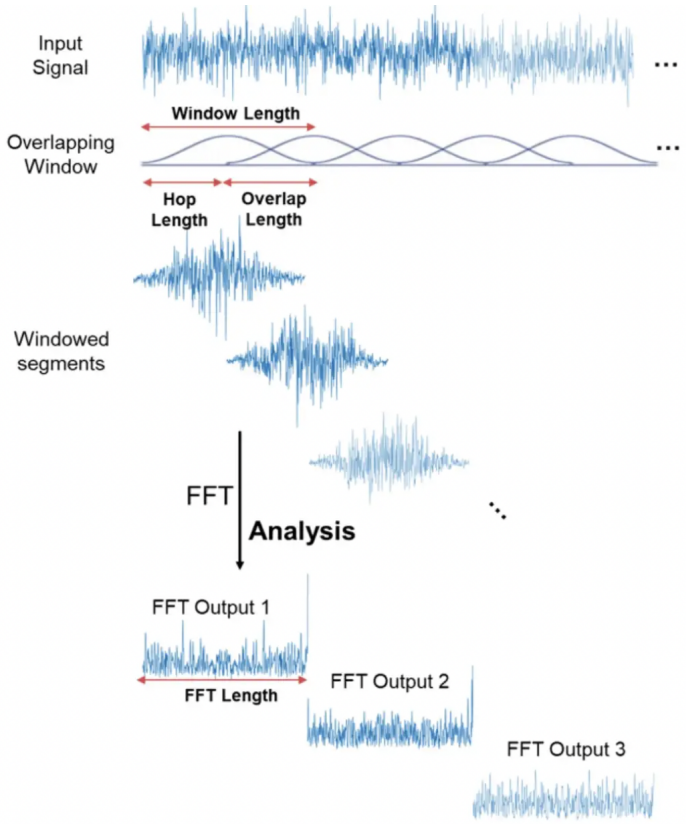

Fig 3: STFT

B. short-time Fourier transform(STFT)

- The fast Fourier transform (FFT) is a commonly used technique for analyzing the frequency content of signals, but it is not well suited to signals whose frequency content varies over time. These signals, known as non-periodic signals, are common in audio signals such as music and speech.
- To analyze the spectrum of non-periodic signals, we can use the short-time Fourier transform (STFT), which is simply the FFT computed on overlapping windowed segments of the signal.
- The resulting spectrogram is a visual representation of the signal's amplitude at different frequencies over time.
- To make the spectrogram more intuitive for human perception, the y-axis is typically converted to a log scale and the color dimension is converted to decibels.
- This allows us to easily see the most important features of the signal in terms of frequency and amplitude.

VI. Experiments

To classify the audio sample based on its genre, we have implemented two different techniques. The techniques we have employed are:

A. Convolutional Neural Network (CNN)

- The architecture we have used is:

```
ConvNet(
  (conv1): Conv2d(1, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (pool1): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (conv2): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (pool2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (conv3): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (pool3): MaxPool2d(kernel_size=4, stride=4, padding=0, dilation=1, ceil_mode=False)
  (conv4): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (pool4): MaxPool2d(kernel_size=4, stride=4, padding=0, dilation=1, ceil_mode=False)
  (dropout): Dropout(p=0.5, inplace=False)
  (fc1): Linear(in_features=77824, out_features=1024, bias=True)
  (fc2): Linear(in_features=1024, out_features=256, bias=True)
  (fc3): Linear(in_features=256, out_features=10, bias=True)
)
```

Fig 4: CNN Architecture

- After tuning some hyper parameters and training the model for few numbers of times, we were able to achieve an accuracy around 60
- The train and test loss curve for the model is :

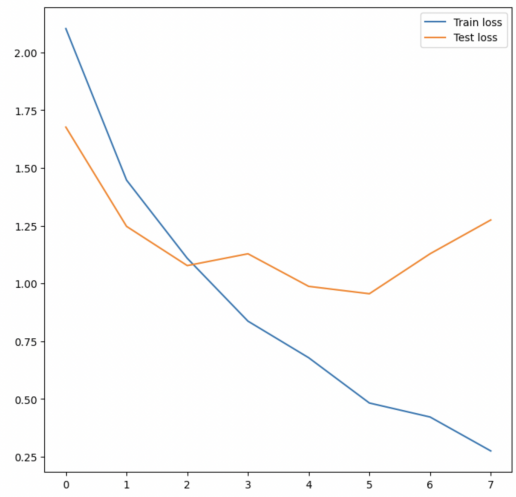

Fig 5: The train and test loss for CNN

- The CNN model is not performing as good as RESNET model as the model is over-fitting after certain number of epochs

B. RESNET 18 Model

- The architecture we have used is:

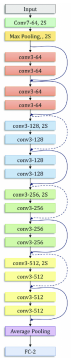

Fig 6: RESNET Architecture

- After tuning some hyper parameters and retrained the last fully connected layer.
- We were able to achieve an accuracy around 71
- The train and test loss curve for the model is :

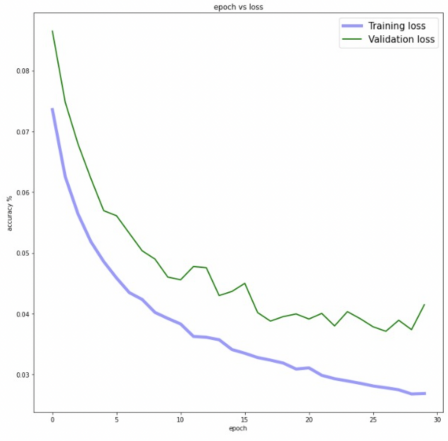

Fig 7: The train and test loss for RESNET

- The CNN model is not performing as good as RESNET model as the model is over-fitting after certain number of epochs

VII. Results

- Our use case is recommending users a playlist of music depending on the music they were already listening to, which doesn't require very high levels of accuracy to actually work. Instead, it only requires that similar music is recommended.
- Also, some inaccuracy might be rather beneficial in our case to provide a wider variation of music to the user.
- The two approaches used by us are a regular CNN and a pre-trained ResNet18 model. The CNN achieved an accuracy of about 65 percent while the pre-trained ResNet model was able to achieve a test accuracy of about 71 percent. Overall, both models are able to learn the patterns in the audios.

VIII. Conclusion and Future Work

Both of our models achieve decent accuracy for our use case. As discussed earlier this use case can actually benefit from being not fully accurate.

- This thought process gives rise to one future implementation that is controlling a variable that helps us bring more variety to what is being recommended to the user. In other words, we are able to control how close to the original songs are the songs that are recommended.
- Another future implementation would be using more data and a bigger data set involving a higher number of genres for the training.
- Using data augmentation we can increase our data set by using the following transformations:
  - time stretching the audio sample
  - making changes in tempo
  - making changes in harmonic component
  - making changes in percussive component
  - adding beat into the audio sample
  - remixing the audio sample
- All of the above transformations can be performed using Librosa

References

1) Music Genre Classification with Paralleling Recurrent Convolutional Neural Network
2) A Survey on Music Genre Classification using Machine Learning
3) Music Genre Classification
4) A Music Genre Classification Method Based on Deep Learning
5) Understanding Mel Spectogram
6) GTZAN Dataset - Music Genre Classification