CS 577 – Spring 2020

Music Analysis and Genre Classification using Deep Learning

Manoj Narayan Bisarahalli – A20452726
Vaishnavi Manjunath – A20446043

1. Problem Statement

Genre classification is an important task with many real-world applications. Being able to instantly classify songs in any given playlist or library by genre is an important functionality for any music streaming/purchasing service, and the capacity for statistical analysis that correctly and complete labeling of music and audio files is essentially limitless.

2. Abstract

Free Music Archive (FMA), an open and easily accessible dataset suitable for evaluating several tasks in MIR, a field concerned with browsing, searching, and organizing large music collections. It provides full-length and high-quality audio, pre-computed features, together with track and user-level metadata, tags and free-form. In this project, we use CRNN (Convolution Recurrent Neural Network) architecture for music tagging used to classify genre of the mentioned FMA music metadata. CRNNs take advantage of CNNs for local feature extraction and RNNs for temporal summarization of the extracted features. We see how to process audio files, train, test a CRNN network and examine the results.

3. Introduction

A general definition - “music genre is a conventional category that identifies pieces of music as belonging to a shared tradition or set of conventions” . The term “genre” is a subject to interpretation, and it is often the case that genres may be very fuzzy in their definition.

Currently, genre classification is performed manually by humans applying their personal understanding of music. This task has poor performance by conventional algorithmic approaches since the distinctions between music genres are relatively subjective and ill-defined. Given enough audio data, of which large amounts can be easily harvested from freely available music like FMA dataset, machine learning can observe and make predictions using these music patterns.

This project aims at content-based classification, focusing on information within the audio rather than extraneously appended information. The traditional deep learning approach for classification is used - find suitable features of data, train classifiers on feature data, make predictions of the genre.

4. Dataset

The dataset we chose for this project is FMA dataset [1] which is an open and easily accessible dataset and contains 106,574 tracks from 16,341 artists and 14,854 albums, arranged in a hierarchical taxonomy of 161 genres. The dataset has the benefits like it is metadata, pre-computed features, and copyright-free audio. The full dataset is divided into three subsets: small, medium and large.

4.1 Small Dataset – This is composed of 8000 30s audio clips from 8 top genres, balanced with 1000 clips per genre, 1 root genre per clip.

4.2 Medium Dataset – This is composed of 25000 30s audio clips, genre unbalanced with 21 – 7103 clips per top genre, but only one of the 16 top genres per clip.

4.3 Large Dataset – This is the full dataset with audio limited to 30s clips extracted from the middle of tracks.

We have chosen small and medium dataset in this project and 4 classes i.e., 4 genres:

- Electronic
- Folk
- Rock
- Hip-Hop

5. Proposed Methodology

The proposed methodology is done in two steps:

- Pre-process audio files
  - Convert raw audio (30 secs) to mel spectrograms
- Use CRNN network configuration
  - Convolution blocks - extract features from mel spectrograms
  - Recurrent blocks - Learn temporal summarization of the features
  - Fully connected block - Classify based on inputs

6. Implementation

6.1 Audio Pre-Processing

Audio pre-processing involves each audio file being converted into a mel-spectrogram which is a visual representation of spectrum of frequencies over time. A mel-spectrogram is a detailed view of audio, able to represent time, frequency, and amplitude all on one graph. Mel-frequency spacing better approximates the hearing scale for human ears where lower frequencies are emphasized, and higher frequencies are compressed.

We used LIBROSA library to produce mel-spectrograms for each audio track. Raw audio of approximately 30 secs are used. Shorter clips are padded with zeros and larger ones are clipped. Frequency of the resulting mel spectrogram is converted to db (decibels). Furthermore, the Mel-spectrograms produced by Librosa were scaled by a log function. This maps the sound data to the normal logarithmic scale used to determine

loudness in decibels (dB) as it relates to the human-perceived pitch. As a result of this transformation each audio signal gets converted to a mel-spectrogram of shape – 96, 1366

Figure 1 below shows the raw audio signals and their corresponding Fourier Transforms.

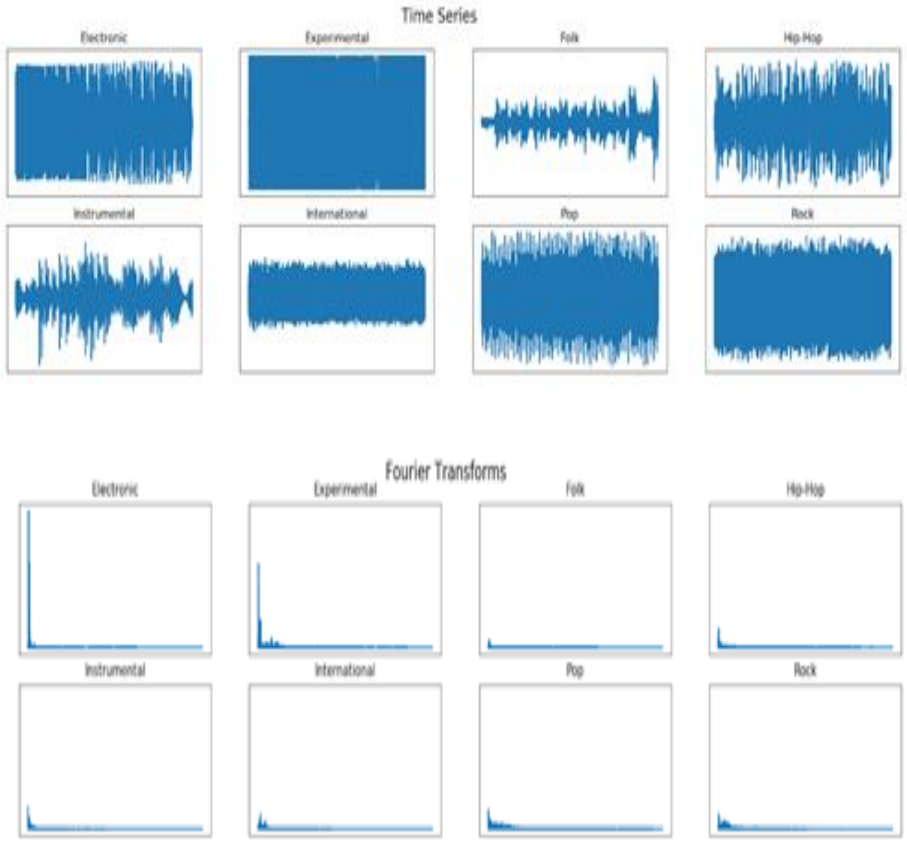

Figure 1: Raw Audio Signals and Corresponding Fourier Transforms

Figure 2 below shows the mel-spectrograms of four genres. As shown in the below figure, for each genre the spectrogram is distinct. This spectrogram is made compatible for genre classification in the CRNN model.

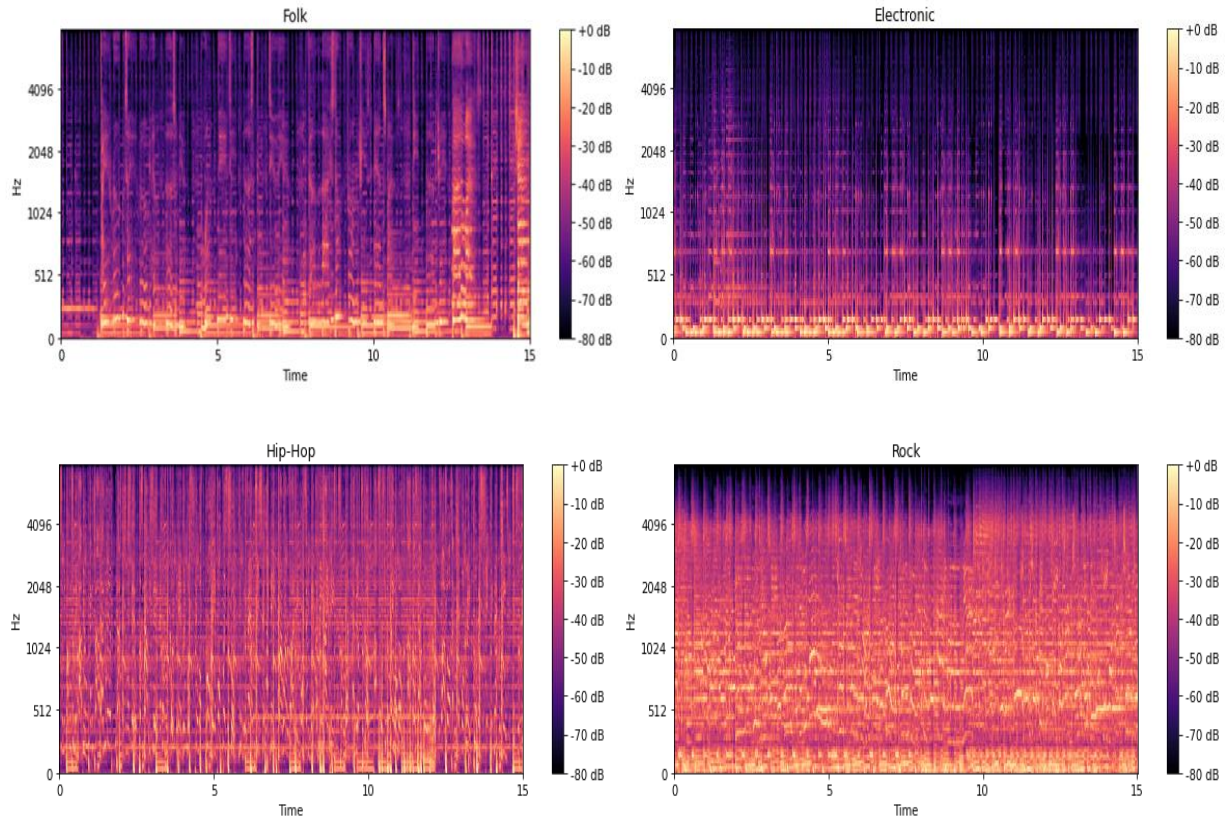

Figure 2: Spectrogram – Folk (BR), Electronic (BL), Hip-Hop (BR) and Rock (TL)

6.2 CRNN Model

A Convolution Neural Network or CNN is a special kind of neural network for processing data that has a known, grid-like topology. CNN are mostly used in image recognition tasks. Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers.

A Recurrent Neural Network or RNN are the class of neural networks that allow previous outputs to be used as inputs while having hidden states. RNN are mostly used in speech recognition tasks. Recurrent networks are very powerful networks because of their distributed hidden state that allows them to store a lot of information about the past efficiently and non-linear dynamics that allows them to update their hidden state in complicated ways.

Description of our model:

Three CNN blocks and one RNN block (two GRU layers) are used. Each CNN block is made up of Convolution 2D layer, Batch Normalization 2D, Max Pool 2D and an Alpha Dropout layer. The CNN layer extracts necessary features from the mel-spectrograms whereas the RNN block summarizes the temporal information in the dataset which is important in the classification. The output from the convolution layer is fed into the Recurrent Neural Network. RNN block consists of two GRU (Gated Recurrent Unit) layers. Figure 3 below shows the network architecture of our model.

6.3 Network Configuration

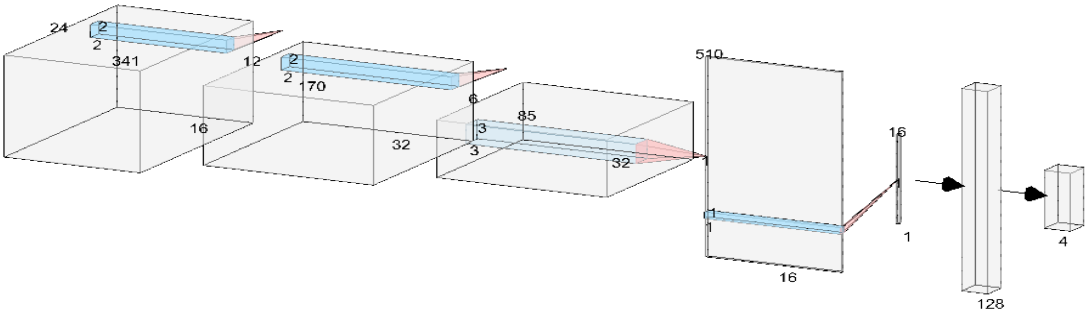

Figure 3: CRNN Architecture with one Dense and output layer

Each 2D layer extracts features from a small slice of spectrogram as shown in Figure 4 below.

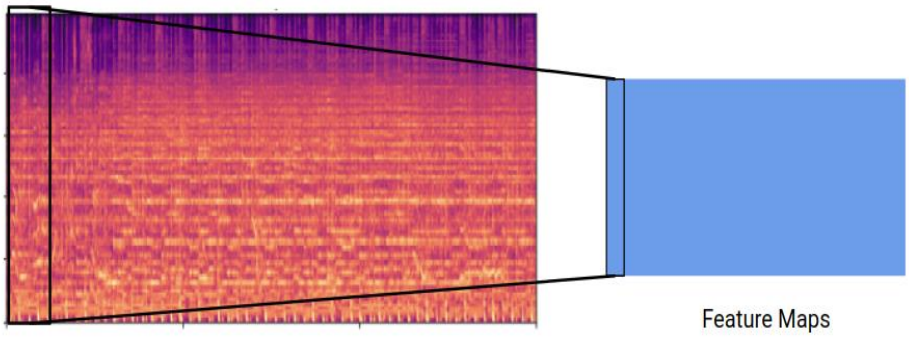

Figure 4: 2D Convolution done by the CRNN model

7. Experimental Results

7.1 FMA Small

7.1.1 Network Architecture

- 3 Convolution Blocks (Conv2D + BN + MP + Dropout)
  - **a.** 16 , 32, 32 filters in each block respectively
- Permute and Reshape layers
- 1 RNN block - GRU with 8 units each
- 1 Dense layer with 128 units

7.1.2 Parameters

- Trained for 20 epochs, Adam optimizer, lr = 0.0001
- loss = categorical cross entropy, batch size = 128

With the above configuration the achieved test accuracy was **50%.**

Tuning the network involved changing the number of RNN units and CNN filters along with early stopping to prevent over fitting. More data is needed to train the network to improve accuracy.

7.2 FMA Medium

An important task in tuning a network to learning from the existing dataset and generalize well is to choose the right network configuration. Fig 5 shows the varying parameters that improved the testing accuracy.

CNN Blocks	RNN Blocks	Dense Layer	Activation	Test Accuracy %
3 16 F (2,2) 32 F (2,2) 32 F (3,3)	1 - GRU 16	1 - 128 units	relu	**60**
3 16 F (2,2) 32 F (2,2) 64 F (3,3)	2 - GRU 8 units each	1 - 128 units	relu Alpha Dropout	**70**
3 16 F (2,2) 32 F (2,2) 64 F (3,3)	2 - GRU 16 units each	1 - 128 units	relu Alpha Dropout	**72**

Fig 5: Table describing experiments on the medium dataset

8. Analysis

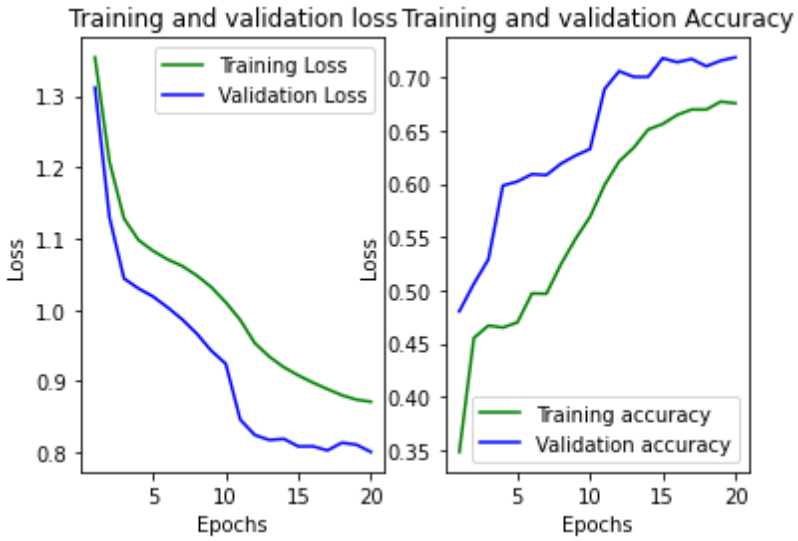

Figure 6: Training, Validation loss vs accuracy graph

The final configuration shown in the FMA medium section yields the results shown in Fig 6. We can see that the network does not overfit since the validation accuracy is always higher than the training accuracy and it increases gradually. Further, loss also behaves in a similar fashion. Plotting the confusion matrix is often a good metric of performance for a multi class classification problem like this one.

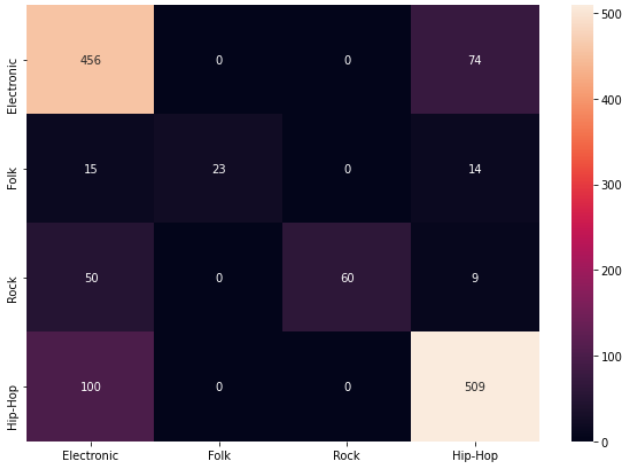

Figure 7: Confusion matrix

9. Visualization

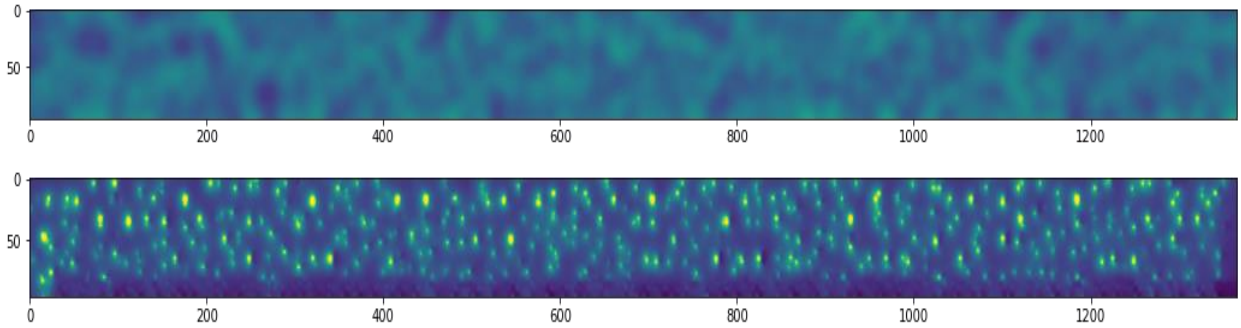

Fig 6: (a) top : Activation of Conv1 Filter 1, (b) below: Activation of Conv3 Filter 16

Visualizing can help in understanding how our network is learning. As we can see in Fig 6(a) the activations are subtle with respect to Fig 6(b) where several parts of the input image contribute to the activation of the filter.

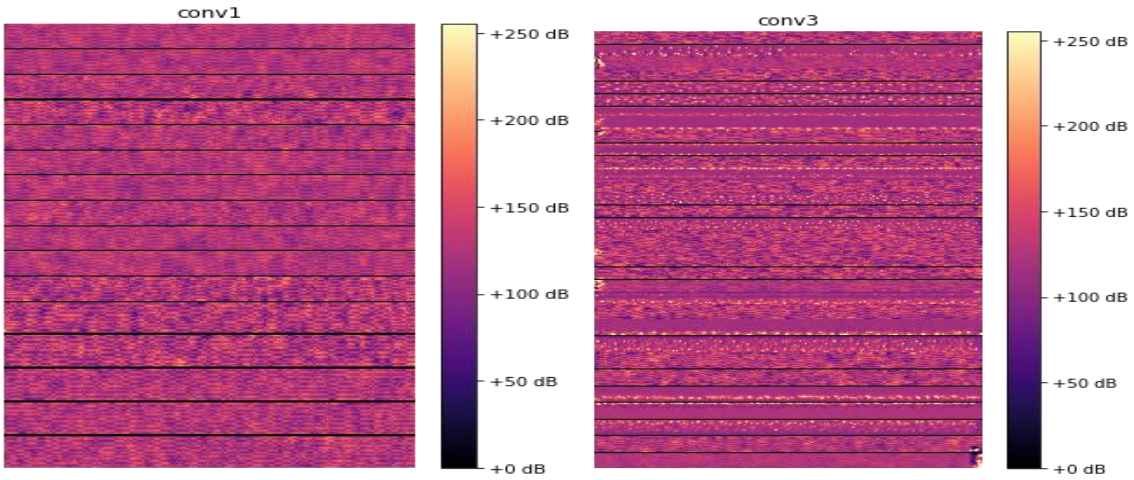

Fig 7: (a) Filter activations for Conv1 (b) Filter activations for Conv3

We can observe that filters in conv1 (Fig 7a) look at smaller areas of size 2 x 2 and can learn smaller patterns when compared to Conv3 (Fig 7b) that can pick up bigger patterns in the image.

10. Conclusion

After examining several choices of datasets, pre-processing methods, neural network structures, and other factors, we found the optimal combination to be a convolutional recurrent neural network using Mel-spectrograms of thirty seconds long samples of audio. A bigger dataset improves our final accuracy and f1 score but further tuning and class balancing techniques can further improve our results. Our final best test accuracy turned out to be 72%. Although it was inferior to the state-of-art accuracy for music genre classification, it outperformed other attempts to solve this challenge with convolutional neural networks. We also discovered that the classification accuracy was highly genre-dependent, which could have impeded the overall performance. It also showed genre selection's great impact on the classification difficulty.

11. References

[1] Michal Defferrard, Kirell Benzi, Pierre Vandergheynst, Xavier Bresson. FMA: A Dataset For Music Analysis. arXiv:1612.01840

[2] Keras Library API Documentation - https://keras.io/

[3] Scikit Learn - https://scikit-learn.org/

[4] Tzanetakis, G. and Cook, P. (2002). "Musical genre classification of audio signals." IEEE Transactions on Speech and Audio Processing, 10(5), pp.293-302

[5] Ioffe, Sergey and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." ICML (2015)

[6] ybayle/awesome-deep-learning-music: List of articles related to deep learning applied to music