Exploring Deep Learning Techniques using SciKit-Learn and TensorFlow for Music Genre Classification on the FMA Dataset

Ben Chapman-Kish
School of Engineering
University of Guelph
Guelph, Canada

November 30, 2022

***Abstract*—The field of music information retrieval has existed for as long as image classification, but remains hindered by a distinct lack of variety in the use of labelled datasets for music genres. There have been numerous recent advances in the technologies behind categorizing the music genre of audio signals through the use of deep learning techniques, but their benefits are understated when the datasets being used are poor representations of the actual music that people are listening to. In this project, I propose the application of modern machine learning techniques to an underrated and well-deserving dataset, the Free Music Archive.**

Keywords—Deep Learning, Neural Networks, MIR, MGR, FMA, GTZAN, Parallel Recurrent-Convolutional Networks, Audio Feature Engineering, Mel-Frequency Cepstral Coefficients

I. Introduction

The field of music information retrieval (MIR) has been of interest to computer science researchers since computing power has advanced enough to allow for complicated processing of audio signals. In a similar vein to applications of AI such as image classification and text analysis, audio classification remains the lesser-known but just-as-powerful cousin which powers multi-billion-dollar industries in the modern world. It is held back in the mainstream lens by the lack of immediate visual clarity in how features can be extracted and transformed to create convincing predictions, but the very same principles that can work on images and text can also work on audio.

Over the past two decades, countless researchers have contributed to the development of more modern machine learning systems that can classify the genre of songs faster and more reliably, to a greater degree of precision through multi-genre and hierarchal classifications, and with consideration of further contexts and social trends that are intimately associated with music genres. These works have been applied in large part to ancient datasets, originally for lack of a suitable alternative but recently as a mere benchmark or proof-of-concept not intended to be applied for real-world settings. In this project, we will cover the history of music genre recognition/classification (MGR) and attempt to reproduce model performance on a modern and high-quality music genre dataset.

II. Dataset

The first well-known case of MGR was in 2002 when Tzanetakis and Cook [1] proposed the famous GTZAN music dataset and created a simple classifier to predict music genres. Through the application of now-archaic ANNs they were able to achieve 65% accuracy on their dataset, and later attempts by other research teams only improved this. According to a survey by Sturm in 2013 [2], the GTZAN dataset had already been used in more than 100 papers and was considered the industry standard of the field. It comprised 10 genres of 100 songs each (limited to 30s samples), for a total of 1000 tracks – but it came with many shortcomings. Aside from the obvious issue of all the songs being over 20 years old, and thus the dataset not including music genres that hadn't been invented yet, Sturm pointed out that GTZAN had a problem of incorrect labels, heavy artist repetition, and many of the songs were copyrighted and could not be used for free in machine learning research.

This was observed by Defferrard, Benzi, Vandergheynst, and Bresson, who in 2016 created a new dataset called the Free Music Archive [3] which aimed to solve all of these problems while significantly increasing the number of songs and genres available for training. The complete FMA dataset includes a whopping 106,574 songs with over 50 features and labels, including an hierarchy of genres as agreed upon by human contributors. Unlike GTZAN, these songs are entirely royalty-free and they are offered in their full, uncropped glory. All signs indicate that this dataset would immediately take over the field of MGR and become the new de facto standard, but regrettably there are very few publications which leverage its power. Of our efforts to discover them, we found only one paper by Yuan, Zheng, Song, and Zhao [4] which made use of the FMA dataset – even then, no new architectures were proposed and a small 8,252-song subsample of the complete data set was used for training of the models. This is a regrettable state of events for both casual enthusiasts and dedicated researchers of MGR alike, and we will demonstrate in this project the value of switching to the FMA dataset for the industry.

III. Literature Review

There have been countless papers investigating the application of various machine learning techniques to MGR – much of the research up until 2010 has been summarized by Fu, Lu, Ting, and Zhang [5], noting that the most common choice of classifier was K-nearest neighbour (K-NN), support vector machine (SVM), and Gaussian mixture model (GMM). These traditional approaches fell to the wayside when deep learning became popular after the advent of AlexNet on image classification in 2012, and the cross-domain applications of similar structures became apparent. Researchers such as Sigtia and Dixon [6] were quick to successfully employ deep learning networks for MRG; in particular, convolutional neural networks (CNNs) seemed to be the most promising type of model.

Especially in the contemporary era since the creation of the FMA dataset, there continue to be countless advances made with new network architectures and training methods within the field of MIR. In 2016, Choi, Fazekas, Sandler, and Cho [7] proposed a hybrid model of CNNs and recurrent neural networks (RNNs) which they called CRNNs, a structure well-suited to music feature extraction and feature summarisation. The next year, Feng, Liu, and Yao [8] took this hybrid approach to the next level by having both the CNN and RNN components operate in parallel, an architecture that they named PRCNN – and yet all of these publications used the GTZAN dataset, or in some cases, the Million Songs Dataset (MSD).

Very recently, in 2021, a case study on these parallel hybrid structures for MGR was published by Yuan, Zheng, Song, and Zhao [4], in which they used the PRCNN framework described above but trained it on the FMA dataset, achieving an overall accuracy of 88% – a respectable improvement considering the challenges of recognizing such a broad number of genres. In the same year, Chaudary, Aziz, Khan, and Gretschmann [9] had success using SVMs after extracting features with Empirical Mode Decomposition (EMD). Also in the same year, Puppala, Muvva, Chinige, and Rajendran [10] made strides in the feature extraction part of MGR using Mel-frequency Cepstral coefficients (MFCCs) and fast Fourier transforms (FFTs), which was also explored later that year by Khasgiwala and Tailor [11] on transformer-based models in addition to CNNs.

The most recent advancement came with the publication of a paper by Chaudhury, Karami, and Ghazanfar several months ago [13], exploring the use of Apache Spark as a parallelized distributed machine learning system, enabling much faster computations on large datasets. They used the GTZAN dataset and found that random forest classifiers had the best performance. The literature review in that paper explicitly addresses many of the techniques that I've listed above, and I consider it the best overview of recent improvements in deep learning solutions for MGR as it pertains to this project. However, due to technical limitations of reproducing these networks on a personal notebook, the distributed approaches will be admired in theory and not put to the test in this project.

IV. Feature Engineering

Before going any further with the descriptions of modern solutions to MGR, it's essential to explain the features that these networks are actually training on and using to distinguish between genres. As Tzanetakis and Cook [1] discovered while creating the first music genre classifier, these are the most essential differentiating qualities of audio signals that can provide a boundary between music genres:

1. Rhythmic content (beats)
2. Pitch content (melodies)
3. Timbral texture (instrumentation)

There are a variety of ways to extract these features from audio signals, but by far the most common and practical approach is the use of Mel-frequency cepstral coefficients (MFCCs), which provide a multi-dimensional summary of these components to a high degree of precision, allowing for minimization of bias and variance in the prediction of genres. The principle behind MFCCs is familiar to all electrical and computer engineers, sound engineers, and signal/system engineers alike: relations and conversions between the time and frequency domains. A Fourier analysis of an audio spectrum can be mapped onto a special acoustic scale called the Mel scale – Mel is short for melody, and you may recognize this scale from the definition of music note A4 as being 440Hz. The exact process by which one would calculate the MFCCs of an audio signal is as follows:

1. Calculate the short-term Fourier transform (STFT) of an audio wave, parameterized by the length of the FFT window and the gap between slides
2. Map the STFT onto the Mel scale
3. Take the logarithm of the powers at each Mel frequency
4. Calculate the discrete cosine transform (DCT) of the log
5. Extract the amplitudes of the DCT into a feature matrix

This process will be processed in advance for each song during the training phase and will need to be computed on the fly for prediction of unseen songs.

V. Methods

The FMA dataset is provided out of the box with a train-test split of 80%-20%, which is perfect for our purposes. Furthermore, due to computational and storage limitations, a small subset of the FMA dataset (released by the FMA team under the alias FMA_small) is used, as it comprises exactly 8,000 songs in total spanning 8 different genres.

For the purposes of this project, a reproducibility study on old-fashioned classifiers on the FMA dataset is the first step. An exhaustive grid search has been conducted over the following classifiers, provided by SciKit-Learn:

- Support Vector Machines
- Logistic Regression
- Fully-connected Multi-layer Perceptron
- Gaussian Naïve Bayes
- K-Nearest Neighbours
- Ada Boosting
- Decision Trees
- Random Forests
- Gradient Boosting
- XGBoosting
- XGBoosting Random Forests

Identify applicable funding agency here. If none, delete this text box.

However, this grid search took several days of nonstop computations to perform, and the exact initial results were accidentally discarded after the Python kernel of the Jupyter Notebook crashed – the only memorable results were that the worst classifiers were ones designed for linearly-separable problems (i.e. SVMs) and the only classifiers to achieve greater than 80% accuracy were the XGBoost family.

Aside from these, the advanced network architectures of the CNN-RNN combinations were implemented using TensorFlow and Keras, all of which involved rounds of 5-fold cross-validation with a 30-epoch early stopping criterion based on validation accuracy. The parameters of these networks have only been slightly adapted from their original papers to adapt to the input size of the network used for this project – an input size of 20x646, based on the computation of MFCCs. The performance metrics achieved by these networks by my own calculations appear to be skewed by an unknown factor, as they should be even higher than the XGBoosting accuracy in both theory and according to existing implementations, but it lies outside of the scope of this course to discern the underlying problem – a likely factor is the random shuffling of training data, or a misalignment of feature dimensions in a stage of the pipeline.

The serial CRNN had a performance of 66% accuracy and the parallel PR-CNN achieved 71% accuracy, which is still not the worst considering the dataset had only 6,400 songs to train on but it's firmly believed by us that an accuracy of 90% could be attained by proper implementation of these models. The disappointing results are only further inspiration for carrying on this investigation in the future, though.

Acknowledgments

This project was done under the guidance of [redacted] and the ENGG*6500 course staff at the University of Guelph. Code references for CNN and RNN implementations of the music genre classifiers were based on the work of P. Dwivedi [13] and U. Anupam [14].

References

[1] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," in *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293-302, July 2002, doi: 10.1109/TSA.2002.800560.

[2] B. Sturm, "The GTZAN dataset: its contents, its faults, their effects on evaluation, and its future use," 2013, *arXiv preprint*, arXiv:1306.1461.

[3] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "FMA: A Dataset for Music Analysis," in *18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017, arXiv:1612.01840v3.

[4] H. Yuan, W. Zheng, Y. Song and Y. Zhao, "Parallel Deep Neural Networks for Musical Genre Classification: A Case Study," in *2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)*, 2021, pp. 1032-1035, doi: 10.1109/COMPSAC51774.2021.00140.

[5] Z. Fu, G. Lu, K. M. Ting and D. Zhang, "A Survey of Audio-Based Music Classification and Annotation," in *IEEE Transactions on Multimedia*, vol. 13, no. 2, pp. 303-319, section III, April 2011, doi: 10.1109/TMM.2010.2098858.

[6] S. Sigtia and S. Dixon, "Improved music feature learning with deep neural networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 6959-6963, doi: 10.1109/ICASSP.2014.6854949.

[7] K. Choi, G. Fazekas, M. Sandler and K. Cho, "Convolutional recurrent neural networks for music classification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 2392-2396, doi: 10.1109/ICASSP.2017.7952585.

[8] L. Feng, S. Liu, and J. Yao, "Music Genre Classification with Paralleling Recurrent Convolutional Neural Network," 2017, *arXiv preprint*, arXiv:1712.08370.

[9] E. Chaudary, S. Aziz, M. U. Khan and P. Gretschmann, "Music Genre Classification using Support Vector Machine and Empirical Mode Decomposition," in *2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC)*, 2021, pp. 1-5, doi: 10.1109/MAJICC53071.2021.9526251.

[10] L. K. Puppala, S. S. R. Muvva, S. R. Chinige and P. S. Rajendran, "A Novel Music Genre Classification Using Convolutional Neural Network," in *2021 6th International Conference on Communication and Electronics Systems (ICCES)*, 2021, pp. 1246-1249, doi: 10.1109/ICCES51350.2021.9489022.

[11] Y. Khasgiwala and J. Tailor, "Vision Transformer for Music Genre Classification using Mel-frequency Cepstrum Coefficient," in *2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON)*, 2021, pp. 1-5, doi: 10.1109/GUCON50781.2021.9573568.

[12] M. Chaudhury, A. Karami, and M. A. Ghazanfar, "Large-Scale Music Genre Analysis and Classification Using Machine Learning with Apache Spark," in *Electronics*, vol. 11, no. 16, p. 2567, Aug. 2022, doi: 10.3390/electronics11162567.

[13] P. Dwivedi. "Using CNNs and RNNs for Music Genre Recognition", *Towards Data Science*, December 13, 2018. Retrieved November 11, 2022. Online: https://towardsdatascience.com/using-cnns-and-rnns-for-music-genre-recognition-2435fb2ed6af

[14] U. Anupam. "Music Genre Classification (Python, GTZAN Dataset)", *Kaggle*, May 14, 2021. Retrieved November 11, 2022. Online: https://www.kaggle.com/code/dapy15/music-genre-classification