Music Genre Classification using Deep Learning

Nikita Patel

Abstract— **This paper analyzed a series of deep learning architectures to determine the most effective model when performing genre classification on audio files. The analysis used the GTZAN dataset to analyze clips 6 seconds in length into one of 10 genres. Using raw audio data pre-processed and transformed into MFCC data, the tested models included several variations of Multi-Layer Perceptron (MLP), Convolutional Neural Networks (CNN) and Relational Neural Networks (RNN) and It was identified that both CNNs as well as RNNs could identify audio genre at a similar level of accuracy, about 77%. The advantages that would typically be present in an RNN for time series data were not significant in this implementation due to the the audio data was pre-processed and fed into the models. The results of this analysis could further be leveraged in other types of data tagging as well as be combined with techniques like Natural Language Processing to increase overall accuracy.**

I. Introduction

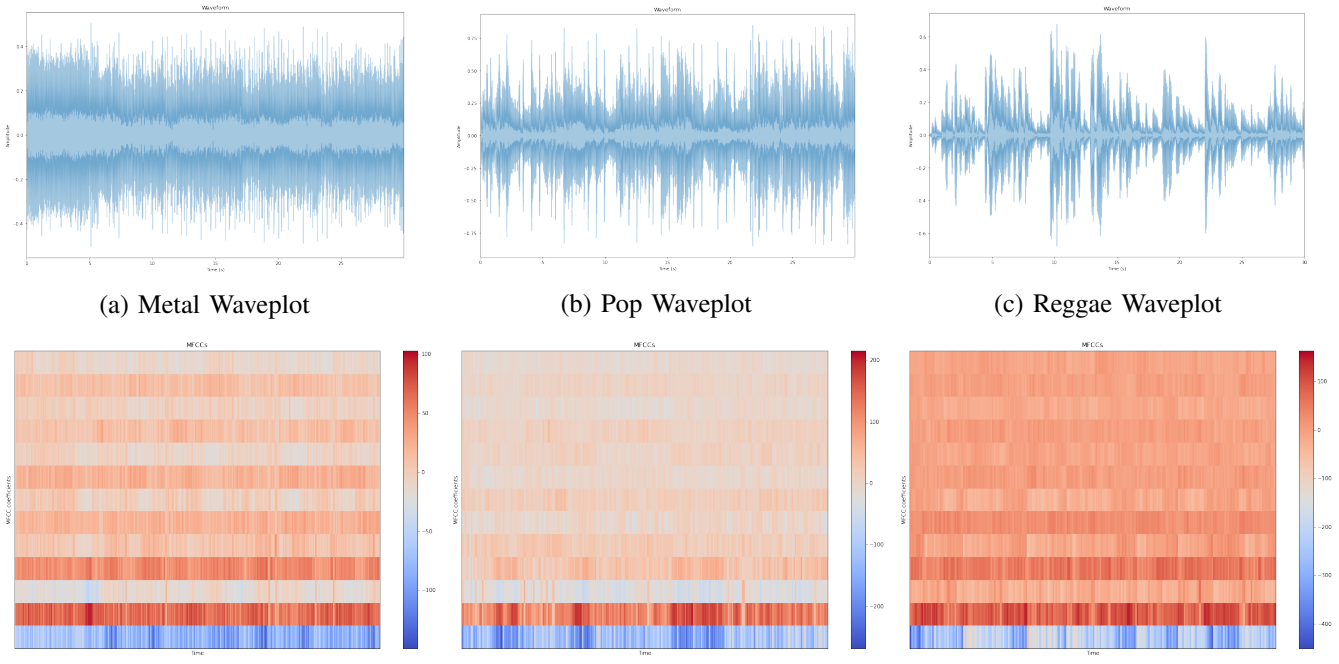
The use of learning techniques in the realm of audio classification is not novel or new, especially in an age of music streaming platforms like Spotify, Apple Music and YouTube. An average of 40,000 new songs are added to Spotify a day as of 2019, with the platform hosting around 50 million tracks in total [1]. With such enormous amounts of data, streaming services must have autonomous ways organizing and managing all the available audio. Beyond data organization, good streaming services also find ways for their users to discover new music based on their taste through recommendation systems. What was done manually by a radio station, is now done autonomously for individual listeners based on their tastes.

Genres are one way in which these platforms can identify the musical taste of an individual and subsequently recommend music. While genre is an intuitive and logical way to group music, how genres are identified is quite complex. The genre of song can be derived from a variety of factors including the musical techniques used, the cultural context of the music, the content and spirit of the themes in the music, as well as the geographical origin of the music [2]. Another misconception is that classification is simple because there is a finite number of genres for classification, however, there about 1,300 genres in the world as identified by Spotify in 2017 [3]. This paper focuses on identifying audio as one of ten musical genres: blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae and rock.

In literature, there are methods through which researchers have approached the classification and tagging of audio. The variations exist both in the type of data used, how that data was pre-processed, as well as the techniques implemented to extract valuable features from the data. This paper aims to analyze a variety of techniques and identify which are best for genre classification given raw audio data all pre-processed in the same way to create a basis for comparison.

II. Background Work

Most research in music tagging is focused on two types of classification: mood-based classification and genre-based classification. In their paper [4] authors Bischoff et al., explored mood and theme classification of music using combinations of audio and its corresponding social annotations. A single mood class would be a combination of several mood tags (e.g. MM1 mood was defined as a combination of Passionate, Rousing, Confident, Boisterous, and Rowdy) and themes (e.g. wedding songs or night driving). Exploring a variety of machine learning algorithms such as K-Nearest Neighbors, Logistic Regression, and others, the authors compared the classification abilities of these algorithms using recall, precision, F1, and accuracy values. Interestingly, the authors experimented with using audio data and tag data separately as well as combining the two types of data through linear combination. In their results,



(a) Metal Waveplot (b) Pop Waveplot (c) Reggae Waveplot

(d) Metal MFCC (e) Pop MFCC (f) Reggae MFCC

Fig. 1: Waveplots and MFCC plots for genres of metal, pop and reggae

it was found that using a linear combination of a Support Vector Machine for audio data and Naive Bayes Multinomial for social annotations achieved their best results with a reported 56.9% accuracy for mood classification and 62.5% accuracy for theme classification. This analysis aims to achieve a similar or better accuracy for genre classification through using only raw audio data and the implementation of deep learning techniques.

The author of [4] approached mood classification from a different perspective than most. Using data drawn from the Spotify API, the author used the derived features of danceability, acousticness, energy, instrumentalness, liveness, valence, loudness, speechiness, and tempo (as defined and provided through the Spotify API) instead of raw audio data to classify songs as calm, energetic, happy or sad. The use of derived data as well as the modern music available through Spotify set this approach apart from others. Using a neural network with a single hidden layer, implementing a ReLu activation function, and an output layer, using a softmax activation function, they were able to achieve 76% accuracy in mood classification. While this level of accuracy was impressive for a single-layer perceptron, the chosen classes were not extensive enough to offer valuable information to an algorithm or a user. Also, the use of derived features in place of raw audio data causes a loss of relevant nuance that could be relevant to a tagging algorithm.

Genre classification, explored by [5], used the GTZAN dataset and audio files were classified as 1 of 10 genres. Audio data formatted as Mel Frequency Cepstral Coefficients (MFCC) was used as an input and passed into a K-Nearest Neighbors (KNN) classifier. KNNs are fairly simple models that require no training and leverages given inputs proximity to k other classified data samples with similar features to derive a classification. With a K value of 5, they were able to achieve an accuracy of 69.4%. The authors used the same dataset and pre-processing techniques presented in this analysis allowing their results to be directly comparable. This analysis aims to see if deep learning techniques can offer improvements to the accuracy achieved using KNN.

Approaching the problem from a deep learning perspective, Choi et al. [6] compared the use of three different convolutional neural networks (CNN), varying in kernel and convolution dimensions, to one that appended recurrent neural net-

works (RNN) to a CNN. The authors leveraged CNNs for local feature extraction and used the RNNs for "temporal summarization of extracted features" creating what they called a convolutional recurrent neural network (CRNN). They compared each model's ability to tag audio correctly according to mood, genre, instruments, and era. Using Area Under Curve (AUC) as their measure, the authors found that a 2-dimensional convolution with a 2-dimensional kernel performed similarly to their CRNN and was slightly more advantageous due to their speed in performing the classification.

III. Methodology

While mood-based classification was considered for this analysis, due to the context-based nature of mood, the high degree of subjectivity involved with classifying it, as well as the inability to define a finite number of mood classes, a genre-based classifier was implemented. Also, there is a need for natural language processing within mood classification. While music in audio could be classified as calming, the language of its lyrics could distinguish it from chill or depressing, thus, introducing another level of complexity. Overall genre-tagged data is also more readily available, and genres are more easily identifiable.

Genre classification was conducted using a variety of deep learning architectures. Using accuracy as the primary value for comparison, the genre tagged data was inputted into a series of multi-level perceptrons, convolutional neural networks, and recurrent neural networks that varied in layers and dropout percentages. Every implemented architecture implemented an Adam optimizer.

A. Dataset

Similar to [5] the GTZAN dataset [7] was used for genre classification. It consists of 1000 audio samples, each 30s in length for each of 10 genres. The genres include blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock. To create more data for training and testing, each 30s audio file was segmented into five segments of 6 seconds each, creating a total of 500,000 audio files for training, testing, and validation.

B. Pre-processing

In audio classification, pre-processing is a crucial part of the learning process. The representation of audio data determines the features a model attempts to analyze. There are several ways this data often is represented.

Overall, visualizing sound is a tricky and math-intensive task. At its core, the sound is a sequence of vibrations varying in pressure strengths. Waveplots are an initial representation of a piece of audio representing amplitude over time. However, the amplitude is not a sufficient representation of a sound bite since it only relays sound intensity. Therefore, Fast Fourier Transforms are usually implemented at fixed time intervals in the sample to create a spectrogram. A spectrogram is a visual representation of frequencies as they vary through time. Spectrograms display as multi-colored images where each pixel color represents amplitude, and its position represents the frequency at a given time [8]. Using the Mel Scale, the linearity of a spectrogram is converted into data that better represents the way humans hear a sound and a Mel Spectrogram can be created.

Audio clips were converted into a compressed version of a Mel Spectrogram, also known as a Mel-frequency Cepstral Coefficients (MFCC), the same as used in [5]. Since MFCCs are designed to represent how humans hear a sound, they are more biologically inspired and are generally a better representation of audio. The ability to visualize an MFCC as an image with multi-colored; makes methods like convolutions easily performed.

Table I shows the parameters used in the MFCC plots of 6s audio files. Conventional values for each parameter were used as explained in [9].

TABLE I: MFCC Parameters

Sample Rate	# MFCCs	# FFTs	Hop Length
22050	13	2048	512

Figure 1 displays both waveplots and MFCC diagrams for audio clips in the reggae, pop and metal genres for visualization.

C. Multi Layer Perceptrons

Across research, some classification models were more implemented than others. However, no

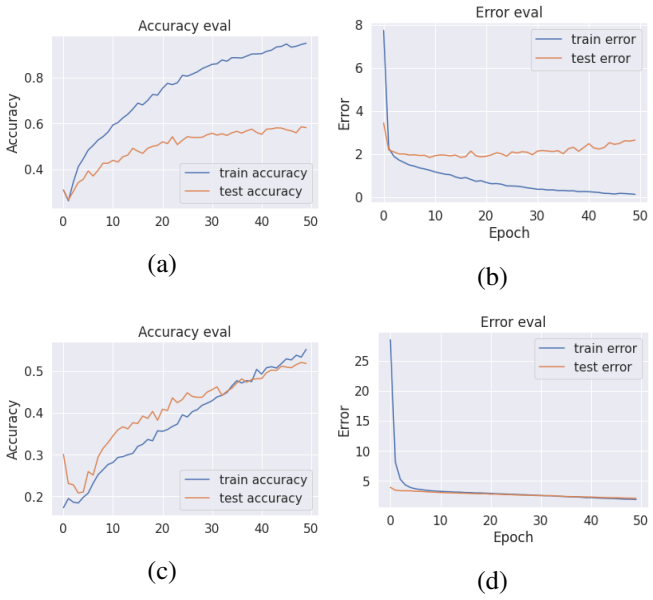


Fig. 2: (a) and (b) show accuracy and error plotted against epoch for a 3 layer MLP while (c) and (d) show the same plots with dropout of 0.3 added to each hidden layer

single method presented superior results to any other. Similar to what [4] implemented, multi-layer perceptrons (MLP) are explored, but with raw audio data rather than derived data. Every implementation of MPLs contained hidden layers using ReLu activation and an output layer with softmax activation. The architecture output was a probability distribution across ten nodes, representing the probability of given audio fitting each of the ten genres. The cross-entropy objective function was chosen to evaluate loss.

The initial MLP implementation contained three hidden layers and an output layer. After 50 epochs, the model achieved an accuracy of 58%. However, overfitting was detected, as is visible in Figures 3a where the accuracy of the test and training sets diverges within five epochs and varies by almost 40% after 50 iterations. Figure 3b displays the test loss increasing within the first ten epochs. Both dropout and regularization are incorporated to mitigate overfitting to the training data. In the same MLP architecture, a regularization constant of 0.001 is used with varying dropout rates ranging from 0 to 0.5 in 0.1 increments. Dropout layers are after each hidden layer. Taking the average across three runs, a dropout of 0.3 outperformed other values resulting in an average accuracy of 50.5% The reduction of overfitting provided by the dropout layer is evident in Figures 3c and 3d

In deep learning, there is often the misconception that the more layers a model has, the better it will be at learning any given item. Along with the MLP architecture discussed above, two other similar architectures are explored. Using a single hidden layer with ReLu and an output layer with Softmax, an average accuracy of 51.5% was achieved. Implementing that same structure but with 5 ReLu hidden layers and a Softmax output, an accuracy of 41.9% is achieved. Both variations used the same regularization value and a variety of dropout rates.

Accuracy and error values for each version of the MLP implementation are visible in Table II below.

TABLE II: Multi-Layer Perceptron Architectures

# Layers	Dropout %	Accuracy	Loss
1	0.0	0.5685228308	9.354848226
	0.1	0.5773035487	3.166623433
	0.2	0.5754140417	2.69190073
	0.3	0.564743797	2.363642136
	0.4	0.5462932189	2.1477681
	0.5	0.5135044853	2.161022902
3	0.2	0.394575	2.279722
	0.3	0.505168	2.093088
	0.4	0.426475	2.279688
	0.5	0.19751	2.762096
5	0.3	0.489718	2.343419

D. Convolutional Neural Networks

CNNs were the most popular classification models used in audio tagging or classification among literature. Some research aimed to assess the trade-offs between one and two-dimensional convolutions, including , two-dimensional convolutions were deemed more effective.

The CNN model consists of three two-dimensional convolution layers with ReLu activation, followed by max-pooling. The first convolutional layer leverages batch normalization. After flattening, the model uses a hidden Relu layer and a softmax output layer. Once again, a cross-entropy objective function is implemented for calculating loss and performing backpropagation. On the initial run, overfitting was detected so a dropout layer was implemented in between the hidden and output

layers. Dropout rates ranging from 0.1 to 0.8 in 0.1 increments were used in the CNN model, and Table III displays the results

Since literature did not show any significant improvements in modifying the number of convolutional layers implemented, such was not done as part of this experiment.

TABLE III: Convolutional Neural Network Architecture Dropout Experimentation

Dropout %	Accuracy	Loss
0.1	0.741096437	1.172744075
0.2	0.7531012495	0.9861798088
0.3	0.7509670655	0.9542256395
0.4	0.7549686633	0.8701910567
0.5	0.7579031567	0.8851159033
0.6	0.7608376741	0.7860955199
0.7	0.7466986567	0.78643294
0.8	0.7029478567	0.8983779133

E. Recurrent Neural Networks

Audio data inherently is time-series data. A single note provides more information when analyzed with a series of proceeding notes than it does alone. Due to these traits, it is assumed that recurrent neural networks would outperform other deep learning architectures since they have feedback loops allowing them to maintain "memory" over time.

A recurrent neural network is implemented consisting of two long-short-term memory layers (LSTM), a dense layer with ReLu activation, a dropout layer, and an output layer using softmax activation. Once again, as in all the previously implemented architectures, cross-entropy was the chosen objective function.

LSTMs were specifically identified for this implementation due to their ability to learn long-term patterns and their prevalence in audio classification [10]. Table IV shows the different dropout values that were experimented with along with their corresponding accuracy and loss values. The dropout layer is appended after the hidden layer to see if accuracy could be increased through the removal of any potential overfitting. A weak dropout of 0.1 was identified as being optimal for the given architecture and data.

TABLE IV: Recurrent Neural Network Architecture Dropout Experimentation

Dropout %	Accuracy	Loss
0.1	0.7773776054	0.7912333333
0.2	0.7568360766	0.8552333333
0.3	0.7561691205	0.8291666667
0.4	0.7480325467	0.8664253967
0.5	0.7484326959	0.8539088967

IV. Evaluation

Based on the literature available in music classification and tagging, the GTZAN [7] dataset was pre-processed and fed through a series of different deep learning architectures to create a level field for comparison. Doing so allowed for the identification of a method or architecture best suited for genre classification.

MLPs were perhaps the least used deep learning method in the available literature. However, being that they are fundamental deep learning models leveraged in both CNNs and RNNs, they are implemented in this experiment. Varying the dropout rate implemented after the single hidden layer of the model produced ambiguous results. While increasing the dropout caused the error or loss to reduce, it also caused the accuracy to reduce. Therefore, the highest accuracy value is not associated with the lowest loss value. This is potentially indicative of the correct genre having a high probability associated with it but another genre having an even higher probability resulting in incorrect classification. Additional steps may have to be taken to ensure accuracy increases as error and overfitting are minimized. For the purposed of this analysis, a dropout of 0.2 will be considered optimal since it maintains a higher accuracy while still minimizing overfitting as seen in Figure 3.

This architecture is similar to that of [4] but with dropout. While the implementations are similar, and the desired output of both models is similar enough to be comparable, the pre-processing of the data was completely different. Therefore, the assumption cannot directly be made that the pre-processing performed by [4] was superior based on accuracy since the goals of the systems were slightly different.

In both the CNN and RNN implementations,

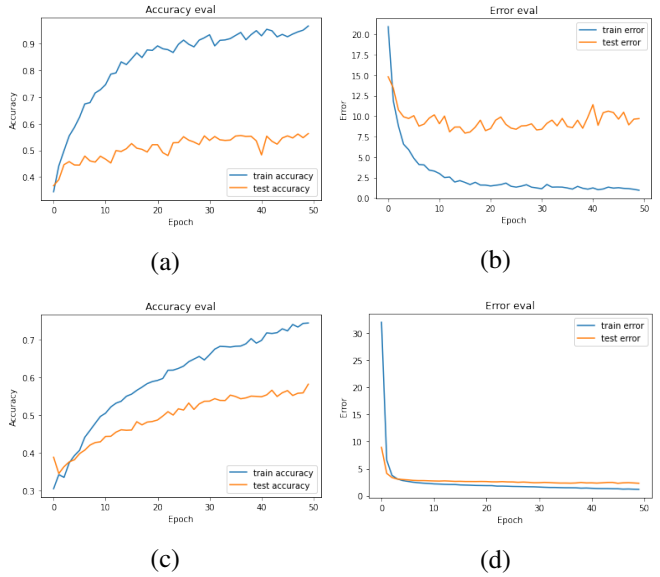


Fig. 3: (a) and (b) show accuracy and error plotted against epoch for a single layer perceptron while (c) and (d) show the same plots with dropout of 0.2 added to the hidden layer

overfitting is visible. However, the CNN architecture has higher accuracy and lower loss at a higher dropout of 0.6, while RNNs work best with a weak dropout of 0.1. This is attributed to the nature of each model since the final accuracy of each was similar as seen in V. The hypothesis was that RNN would offer better accuracy than CNN due to the time series nature of audio data, however, it did not do so beyond a reasonable doubt. This could be due to the way the data was pre-processed. MFCCs are created using both a sample length and a hop rate which means each sliver represented in the diagram is taking previous samples into account. Therefore, the simple relational data acquired by performing convolutions is sufficient for the analysis of such data. These results are similar to what was observed by [6]

TABLE V: Audio Classification using Deep Learning Architectures

Model %	Accuracy	Loss
MLP	0.575414	2.691900
CNN	0.760838	0.786095
RNN	0.777378	0.791233

As mentioned previously, genre classification is far from a perfect science, and therefore, there is a threshold to the accuracy achievable from using strictly raw audio data. The maximum accuracy achieved was 77.7% through the use of RNN-LSTM which, could be further improved through the further adjustment of hyperparameters. The analysis, with confidence, was able to prove the strengths of pre-processing audio data as an MFCC. It showed the need for a model that encompasses a data point's relation to others due to the pattern-based and time-series nature of music.

V. Future Work

Within the field of music classification, there is much that can be done. While the particular implementation explored in this paper is the identification of genre, the analysis's primary focus was to determine the best methods in audio classification when audio is represented as an MFCC. The implemented methods can be extended in various ways to provide insight into the realm of audio data tagging.

As explained previously, the genre is not a finite classification, nor is it mutually exclusive. The first and most simple improvement that can be incorporated into the existing work is the usage of multiple music tags rather than a single final classification. Unfortunately, this will require a divergence from the used GTZAN dataset to one that incorporates multiple genre tags to every audio file.

Natural Language Processing should also be incorporated into the existing CNN or RNN architecture. Similar to the work done in [4], genre classification should be examined using strictly NLP, then compared with the work done in this paper, and then combined to see which offers optimal accuracy. Experimenting in this way could also offer further insight into how humans define a genre, whether it has a heavier basis on lyrical or musical content.

For final tuning of the existing and future architectures, it will be imperative to understand where confusion in tagging occurs most. Confusion matrices should be analyzed and examined to determine where mislabeling could be occurring, and hyperparameters should be adjusted accordingly.

VI. Conclusion

This publication focused on the classification of audio from the GTZAN dataset into ten different genre categories. Genres are inherently a complicated classification mechanism, and therefore perfect accuracy will never be possible without the use of contextual metadata to go with raw audio. Comparing three different deep learning models, it is seen that CNNs and RNN-LSTMs perform similarly on data pre-processed and converted into an MFCC representation of the audio. Future research will be focused on the incorporation of natural language processing as well as tagging, as opposed to one hot-encoding classification, to expose the nuances of a genre.

References

[1] Tim Ingham. Nearly 40,000 tracks are now being added to spotify every single day, Apr 2019.

[2] Music genre, Mar 2021.

[3] Ivan L. How many music genres are there?, Apr 2017.

[4] Kerstin Bischoff, Claudiu S Firan, Raluca Paiu, Wolfgang Nejdl, Cyril Laurier, and Mohamed Sordo. (pdf) music mood and theme classification - a hybrid approach.

[5] Python project - music genre classification.

[6] Keunwoo Choi, Gyorgy Fazekas, Mark Sandler, and Kyunghyun Cho. Convolutional recurrent neural networks for music classification. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[7] George Tzanetakis, Georg Essl, and Perry Cook. Automatic musical genre classification of audio signals, 2001.

[8] Dalya Gartzman. Getting to know the mel spectrogram, May 2020.

[9] Valerio Velardo, Feb 2020.

[10] Mar 2020.