From Beethoven to Beyoncé: A Deep Learning Approach to Music Genre Classification

Stanford CS224N Custom Project

Dominic DeMarco
Department of Computer Science
Stanford University
demarcod@stanford.edu

Eric Martz
Department of Computer Science
Stanford University
emartz@stanford.edu

Regina T.H. Ta
Department of Computer Science
Stanford University
rta@stanford.edu

Abstract

This project explores music genre classification on feature vectors representing short segments of music, aiming to understand how humans and machines perceive music. We aim to improve upon a baseline multilayer perceptron (MLP) model by leveraging long short-term memory (LSTM) models and their use of sequential input data. We measure performance on three, six, and nine second samples, finding that there are additional benefits for shorter samples. We achieve over 90% test accuracy on our primary dataset, a marked improvement from the 60% average by our baseline MLP model and the source paper's 61% model using k-means clustering. By exploring different datasets and feature extraction techniques, our project seeks to bring more clarity into how models learn on music samples. Lastly, we interpret these results through a qualitative lens to determine how machine misclassifications compare to human error.

1 Key Information to include

- Mentor: Bessie Zhang
- External Collaborators (if you have any): None
- Sharing project: No
- Contributions: All team members contributed to LSTM implementation and the write-up. In addition, Dominic developed the baseline MLP model, while Regina and Eric generated data visualizations.

2 Introduction

From Spotify to Shazam, music genre classification comes into play as a challenging task that maps audio signals to the complex social and emotional constructs that are genres. Understanding how machines can perceive and classify music is not only interesting, but important for broader understanding of feature extraction and representations of music samples for deep learning models.

Current state-of-the-art methods succeed in achieving high accuracy with various model architectures, including LSTM and k-clustering, but do not use music samples shorter than 30 seconds (Yi et al., 2021). As a result, less is known about the performance of genre classification models on shorter time sequences. Our paper aims to provide clarity around these topics in order to increase

model interpretability.

We start by replicating our source paper's baseline accuracy using a simple MLP model. We try both 30-second and 3-second samples to determine whether the baseline accuracy of 60% could be replicated with an MLP model. Then, we implement an LSTM model that is run on 3-, 6-, and 9-second music samples to determine the extent to which sample length affects accuracy. Our model succeeds in improving upon the baseline model's accuracy, achieving a test accuracy of over 90% on the GTZAN dataset.

Lastly, we take a qualitative approach to better understand the patterns in classification that our models produce. For example, if there are overlaps and ambiguities between genres, what are the implications for human vs. machine music classification? How might qualitative analysis provide insights into the validity and justifiability of our model's classifications and misclassifications?

3 Related Work

Our project relies principally on two related works which serve as foundational papers in their field. The first, Musical Genre Classification of Audio Signals, provided insights into feature extraction of music audio as well as advanced classification techniques considering its publication in 2002. First, it developed novel features that are specifically designed to describe musical content and musical signals—including rhythm, pitch, and harmony. The paper's results show that it is possible to apply existing standard statistical pattern recognition classifiers – including established approaches like Gaussian classifiers and k-nearest neighbors classifiers – to the novel features extracted in order to successfully classify musical data into genres. In fact, the success rate of automatic musical genre classification is significantly better than random chance and even comparable to human performance on music classification tasks (Tzanetakis and Cook, 2002).

We use this paper's accuracy as a baseline upon which we try to improve, first by replicating its accuracy with an MLP model and then by leveraging an LSTM for increased accuracy. For guidance on using sequential music samples for an LSTM, we turn to our second related work, Music Genre Classification with LSTM based on Time and Frequency Domain Features.

In this paper, the GTZAN dataset is used similarly to our approach. It outlines a methodology involving the extraction of short-term features like zero crossing rate (ZCR) and mel-frequency spectral coefficients (MFCC) from digital music, which are then fed into the LSTM to generate deep features. Support vector machine (SVM) and k-nearest neighbors (KNN) algorithms are employed for classification based on these deep features. Experimental results indicate significant improvement in accuracy compared to traditional methods, demonstrating the efficacy of LSTM-based feature extraction for music genre classification. Due to their combined use of three techniques – SVM and KNN algorithms, and an LSTM model – the paper achieves very high accuracy of 98.9%. (Yi et al., 2021)

4 Approach

We begin by creating a baseline model to replicate and compare against the performance of our source paper's model which was based on k-means clustering. To achieve these results, we implemented a basic MLP model. The baseline model architecture consists of two linear layers with dropout and softmax applied. We used 13 hidden layers for the 30-second feature vectors and 64 hidden layers for the 3-second feature vectors. Our MLP achieved on average 60% accuracy in genre prediction. Our best accuracy for the 30-second samples was 74%, and our best accuracy for the 3-second samples was 65%, in comparison to 61% accuracy from the paper (Tzanetakis and Cook, 2002).

Next, we implemented an LSTM model which works with sequences of 3-, 6 -, and 9-second feature vectors that all sum to 30 seconds (except the 9 second feature vector which sum to 27 seconds). We implemented this model with two LSTM layers and a dense linear layer with a ReLu activation

function, each with 64 hidden parameters. Then, we added dropout and a final softmax layer to output a vector of size equal to the number of prediction categories, which is 10 for the GTZAN dataset.

5 Experiments

5.1 Data

Given audio files as input, our task is to predict the correct genre classification for each file. We primarily used the GTZAN dataset, which contains 1,000 human-labeled audio files across 10 genres (100 songs per genre), with each file being 30 seconds long. The dataset also contains pre-extracted feature data, one file with the feature data of all 30 seconds of music, and the other file with feature data in 3-second increments (10 feature vectors per audio file).

Each feature vector contains 58 features, including tempo, spectral centroid data (measuring "brightness"), mel-frequency cepstral coefficients (measuring the perceived power spectrum of the sound), chroma data (showing which pitches are most common), Root Mean Square data (measuring average loudness), and zero-crossing rate (detecting percussive sounds).

5.2 Evaluation method

First, we evaluated overall test accuracy, comparing the performance of our baseline MLP model against our LSTM. We further computed the precision, recall, and f1 scores for each genre. For the LSTM model, we tracked the training and validation accuracy along each epoch in order to evaluate whether our model was learning correctly.

Additionally, to interpret the resulting probability distributions, we formulated a "certainty" metric. Defined as the log of the ratio of the most-likely to second-most-likely genre, this gives an intuition into how much confidence the model has that its top prediction is actually the correct answer.

5.3 Experimental details

Our baseline MLP model was trained using 100 epochs with a batch size of 4. Since our dataset is relatively small, our training time is short: $15 - 30$ seconds. This model has two linear layers with 13 hidden features for the 30-second data and 64 hidden features for the 3-second data, and an Adam optimizer with a 0.01 learning rate. The final layer is put through a dropout layer and a softmax to return a probability distribution over each genre. Our baseline MLP model was implemented using PyTorch.

Our LSTM model was trained using 50 epochs with a batch size of 16, and we leveraged the following performance boosts in its architecture: two LSTM layers with 64 hidden features; a dense layer with 64 hidden features, a ReLu activation function, a dropout layer, and the final linear layer with a softmax to return the probability distribution. Our LSTM model had an Adam optimizer with a 0.001 learning rate. The training time was $45 - 60$ seconds for the GTZAN dataset. While we experimented with varying hyperparameters for our LSTM (e.g., the number of hidden parameters, L2 regularization, and dropout probability), the model architecture described above performed best. The input to the LSTM model is a sequence of 10 three-second feature vectors, since each audio sample is 30 seconds long. We also experimented with 6-second features (sequence length of 5) and 9-second features (sequence length of 3). Our LSTM model was implemented using TensorFlow's Keras API.

5.4 GTZAN Results

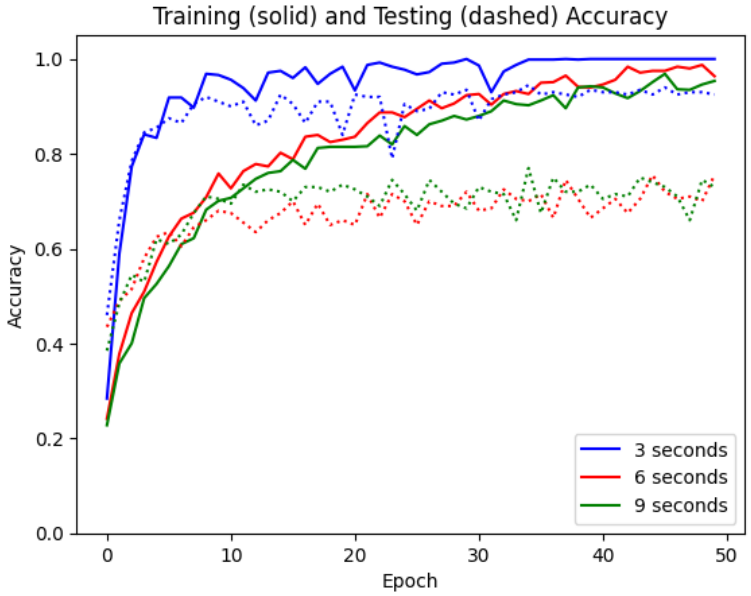
	MLP (30 seconds)	**MLP** (3 seconds)	**LSTM** (3 seconds)	**LSTM** (6 seconds)	**LSTM** (9 seconds)
Overall Test Data Accuracy	74%	66%	93%	76%	73%
Precision per genre					
Blues	87%	69%	83%	57%	65%
Classical	95%	62%	100%	85%	88%
Country	55%	62%	100%	72%	57%
Disco	51%	83%	83%	61%	47%
Hiphop	56%	57%	80%	62%	86%
Jazz	94%	57%	92%	89%	71%
Metal	93%	92%	100%	96%	100%
Pop	82%	90%	94%	93%	92%
Reggae	57%	55%	89%	65%	74%
Rock	62%	67%	95%	72%	60%
Recall per genre					
Blues	46%	62%	100%	87%	73%
Classical	100%	100%	100%	100%	96%
Country	61%	56%	78%	72%	67%
Disco	85%	42%	95%	70%	75%
Hiphop	69%	75%	92%	62%	46%
Jazz	73%	70%	96%	74%	87%
Metal	96%	76%	97%	80%	93%
Pop	95%	65%	75%	65%	60%
Reggae	55%	77%	89%	68%	74%
Rock	27%	43%	100%	68%	32%
F1-score per genre					
Blues	61%	66%	94%	68%	69%
Classical	98%	77%	100%	92%	92%
Country	58%	59%	88%	72%	62%
Disco	64%	56%	88%	65%	58%
Hiphop	62%	65%	86%	62%	60%
Jazz	83%	63%	94%	81%	78%
Metal	95%	83%	98%	87%	97%
Pop	88%	76%	83%	76%	73%
Reggae	56%	64%	89%	67%	74%
Rock	38%	52%	97%	70%	41%

As seen in the table, our best-performing baseline MLP model achieved a 74% accuracy when trained over the 1,000 30-second feature vectors, and 66% when trained over the 10,000 3-second feature vectors. These baselines outperformed the k-means baseline from Tzanetakis and Cook's foundational paper (Tzanetakis and Cook, 2002).

For our LSTM model, we achieved a very high test rating with 93% on the 3-second, 76% on the 6-second, and 73% on the 9-second time slices. When training, we found that our models converged in 20-40 epochs. Given the observed gap between training and testing accuracy, there may be evidence of possible overfitting in the 6-second and 9-second data, potentially due to those time seequences having a smaller number of discrete data points (there are 10,000 3-second vectors, 5,000 6-second vectors, and 3,000 9-second vectors). The training and testing histories are graphed in Figure 1.

Ultimately, achieving 90%+ accuracy on a 10-class genre classification problem where some genre boundaries are ambiguous is quite an accomplishment. For comparison, Tzanetakis and Cook reference an experiment that shows how humans perform at about a 70% accuracy when they classify music into 10 genres after listening to 3-second samples (Tzanetakis and Cook, 2002). The relative success of our LSTM model suggests that NLP-inspired architectures can apply well to audio-processing problems, on the precondition that audio is properly vectorized.

Figure 1: Training accuracy (solid) and testing accuracy (dashed) of the GTZAN data on the LSTM model



6 Analysis

6.1 Time Slice Performances

One interesting result from our LSTM model was that the performance of the model changed when it used data extracted from 3-second, 6-second, and 9-second features. Our original hypothesis was that a 3-second time frame might be too short for the model to infer genre information. However, our results indicate that training on 3-second features significantly improves the performance of the classification task. We think this may be because the 3-second features have longer sequences – 10 segments as compared to 5 for the 6-second features and only three for the 9-second features – which provides more granularity and sequential data for the LSTM to leverage. In the future, we can experiment on finer-grained intervals to determine if and when accuracy deteriorates when increasing the timescale resolution.

Even on our basic MLP model, training on 3-second time slices performed surprisingly well. The correct genre was predicted 51% of the time, and the correct genre was in the top 3 most probable genres returned by the model 78% of the time. While the accuracy difference between the LSTM and MLP model shows the importance of the additional context on improving accuracy, it is nevertheless impressive how even our basic model can effectively infer genre with only 3 second segments of input data. Perhaps the most interesting discrepancy between our baseline MLP model's performance on 30-second vs 3-second features was the drastic decline in precision for the classical category. This is likely due to short instrumental segments within non-classical music - in context, it's clear that these instrumental sections belong to another genre, but without the context of the rest of the piece, the model (and humans) may mistakenly identify a 3-second window as classical. This is part of the reason why our LSTM model, which can take context into account while also examining 3-second time-slices of audio features, outperforms the basic MLP model.

6.2 GTZAN Certainty

In addition to testing the accuracy of our classification models, we examined which genres were easier for the models to differentiate. This information comes from the resulting softmax distribution - a probability distribution over the 10 possible classes. We devised a "certainty" metric, which compares the log ratio between the top predicted genre and the second-most likely genre, defined mathematically

as *certainty*$(x) = \log_{10}(\frac{\max(x)}{\max(x \setminus \max(x))})$. When running our analysis on our top-performing model, here are the resulting certainty scores (paired with recall scores for comparison):

Genre	Confidence	Recall
Blues	1.50	46%
Classical	9.70	100%
Country	2.06	61%
Disco	1.89	85%
HipHop	3.08	69%
Jazz	2.58	73%
Metal	5.79	93%
Pop	4.53	82%
Reggae	2.00	55%
Rock	1.31	27%

From this table, we can see that the model's certainty score generally correlates with recall. It is most "certain" about Classical, Metal, and Pop, and those are among the most accurately predicted genres. Similarly, it has lowest certainty on Blues and Rock, genres which also have low recall. An outlier here is in the Disco genre, which has low certainty but high recall. This is likely because the model, when unsure between Disco and a similar genre (think: Rock or Blues), defaults to Disco. This results in high recall, but low precision (51%).

There are a few reasons why the model is better at discriminating between certain genres over others. Most notably, classical music has a very different audio signature than the other (mainly lyrical) genres of music here. This translates to many distinct audio features, one example of which is the spectral centroid. The spectral centroid roughly correlates with the perceived "brightness" of the sound - and the instrumentation of classical music is generally much brighter than the other genres listed here. In the opposite direction, the metal genre has a characteristically "dark" sound, reflected once again in the spectral centroid, and is therefore also easy for the model to differentiate.

For the genres that our models struggle to differentiate, they tend to be genres that have many stylistic similarities and overlap. In particular, rock is a very broad genre, and many rock songs contain the musical signatures of other genres. Perhaps a Bruce Springstein rock song might sound a bit like country, Def Leppard a bit like metal, Chicago a bit like jazz, or KC and the Sunshine Band a bit like disco. Given that humans may disagree on how to label these songs, it's no surprise that our model also struggles to do the same. After all, these genre labels are human constructs and reflect societal and historical trends in music, not some underlying ground truth. Ultimately, our model may struggle to differentiate genres that many humans would struggle with, too.

6.3 Examining Genre Ambiguity

To see a concrete example of this genre ambiguity, we looked at three of the audio files that the system misclassified. First, there was "reggae.00099", which the model thought was likely metal, rock, or jazz. Listening to the audio, I would have thought this to be jazz, due to the characteristic horn hits. Additionally, this track has a very muted tone, as if it was recorded a long time ago. This darkens the overall sound, which is why the model may have thought this to be metal. Next, there was "rock.00059", an excerpt of Sting's song "Children's Crusade", misclassified as reggae. This classification seems rather reasonable to me, it's a low-tempo ballad with heavy emphasis on rhythmic syncopation. It's certainly a far-cry from typical rock-and-roll songs. Children's Crusade exemplifies why the model had difficulty accurately predicting Rock music - this genre simply encompasses too many diverse styles. Finally, "pop.00079" (Kate Bush - Cloudbusting), was confused with rock, country, and classical, all of which make sense given the 30-second excerpt we hear. The percussion and guitar stabs are typical to rock, the vocal style and sparse arrangement are typical to country, and in this 30-second excerpt there is a lengthy violin solo, which would be more typical of classical music. In general, when examining where the model makes mistakes, we can see that they are either mistakes a human could make or mistakes that make sense in the context of the 30 seconds of sample audio they are fed.

7 Conclusion

Our project has delved into the realm of music genre classification using deep learning and NLP techniques. Through our exploration, we aimed to enhance understanding of how machines perceive and classify music, a task crucial for various applications in the digital music industry. Our efforts culminated in the development of an LSTM-based model that achieved significant improvements over baseline accuracy, reaching up to 93% test accuracy on the GTZAN dataset. We discovered the benefits of using shorter time-slices so as to take advantage of our LSTM's ability to capture dependencies across sequential data. This achievement highlights the potential of applying sequential input and advanced neural network architectures for music genre classification tasks.

Our analysis revealed limitations in the performance of our model on certain genres, indicating the inherent ambiguity and complexity present in music classification tasks. Despite these challenges, our project provides valuable insights into the capabilities and limitations of current deep learning approaches to music genre classification.

Looking ahead, there are several avenues for future work in this field. One promising direction is the exploration of finer-grained temporal resolutions to better understand how accuracy varies with different time scales and uncover the minimum time slice needed to extract genre information. Another would be to implement other NLP-inspried architectures, such as feed-forward networks, transformers, and attention schemes. Additionally, further research could focus on refining feature extraction techniques to improve model performance across a broader range of music genres. Overall, our project contributes to advancing the understanding and development of deep learning models for music genre classification, paving the way for future innovations in the domain.

References

George Tzanetakis and Perry Cook. 2002. Musical genre classification of audio signals. In *IEEE Transactions on Speech and Audio Processing, Volume 10 Issue 5*, pages 293 – 302, Online. Institute of Electric and Electronic Engineers.

Yinhui Yi, Xiaohui Zhu, Yong Yue, and Wei Wang. 2021. Music genre classification with lstm based on time and frequency domain features. In *2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS)*, pages 678–682.