Music Genre Classification with Mel Spectrograms and CNN

Albert Pun
Stanford University
apun@stanford.edu

Kamilla Nazirkhanova
Stanford University
nazirk@stanford.edu

Abstract

Classifying music is often a subjective tasks performed by humans, which makes it a challenging task for algorithms to do automatically. In this study, we first explored traditional machine learning methods, such as SVM and Naive Bayes, to form a baseline on the GTZAN dataset. Our final model employed a neural network approach with both a CNN and FCC part that used extracted features along with Mel Spectrograms to attain a model with 13% higher accuracy than all our baselines.

1. INTRODUCTION

Music is one of the most popular forms of art consumed by people everywhere daily. For instance, in 2020, 25% of Americans reported using Spotify monthly [1]. These music streaming companies use music genre classification to improve their recommendations for the customers and to create playlists of the same genre. Generally, music recommendation is a more complex problem and genre classification is an important step to solve it.

While music genre serves an important purpose, it is often a subjective label given to a music piece by humans. The given genre can depend on their own perception of the music and their knowledge in that music style. This can often lead to conflicting opinions about a music's genre, especially when some music pieces contain elements from multiple genres. This makes automating the process of music genre classification a challenging task.

In this paper, we use the GTZAN dataset [2] to explore a variety of methods to classify songs into 10 genres. To build a baseline, we first implemented a couple traditional machine learning techniques, such as SVMs and logistic regression. To improve upon the baseline, we experimented with fully connected neural networks and convolutional neural networks.

2. RELATED WORKS

The earliest paper that uses the same GTZAN dataset, Tzanetakis et al [2] proposed the use of 9 dimension feature vector, that include beat-related features, pitch-related feature and timbral texture features for automatic music genre recognition using k-Nearest Neighbors (k-NN) classifier and Gaussian Mixture Model (GMM) classifier. Their model was able to achieve a 61% accuracy.

Benetos et al [3] used a tensor representation, where each recording is represented by a feature matrix over time that is created by concatenating feature matrices associated to the recordings. These features were also extracted from the GTZAN dataset and are comprised of various sound descriptions. They created a novel algorithm for non-negative tensor factorization (NTF), which employs the Frobenius norm between an n-dimensional raw feature tensor and its decomposition into a sum of elementary rank-1 tensors. Their model achieved a 75% accuracy on the dataset.

In recent years, neural networks have shown remarkable success in the area of audio data and other forms. However, representing audio in the time domains for inputs to a neural network is not straightforward due to high sampling rates of audio signals. Van Den Oord et al. [4] were able to address these challenges and train on tens of thousands of samples per second of audio for the audio generation tasks. Another representation of audio other than the raw data is the spectrogram of a signal, which captures both time and frequency information. Wyse [5] used a variety of spectral representations as input to a CNN for a music style transfer tasks.

3. DATASET AND FEATURES

GTZAN which was firstly proposed by G. Tzanetakis in [2] is one of the most popular dataset used for music signal processing. It contains 1,000 music with 30-second, 22050 Hz sampling frequency and 16 bits. Genres in the GTZAN are blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae and rock and all of these genres have 100 music pieces. The files were collected in 2000-2001 from a variety of sources including personal CDs, radio, mi-

Feature	Statistical Functions	# of subfeatures
Zero Crossing Rate	Mean, Median, Standard Deviation	3
Spectral Centroid		3
Spectral Contrast		3
Spectral Bandwidth		3
Spectral Rollof		3
MFCC (13 coeff)		39
MFCC Derivation		39
Total		93

Table 1. : GTZAN Feature Summary

crophone recordings, in order to represent a variety of recording conditions

The dataset also contains 7 main features, each with its own subset of features, such as mean, median, and standard deviation. In total, there are 93 features that can be used as input to our model. Additionally, each audio piece has a visual representation in the form of a Mel Spectrogram. The Mel Spectrogram is based off a linear cosine transform of a log power spectrum on a nonlinear Mel scale of frequencies. More importantly, it is able to convert an audio's time series data into a visual representation that entails frequency and time information.

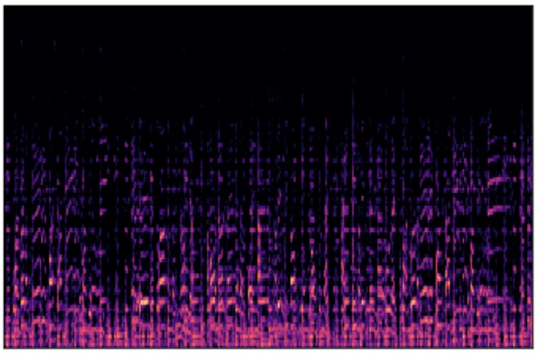

Fig. 1: Mel Spectrogram of a Blues piece

4. METHODS & EXPERIMENTS

We first created a baseline and then experimented with different neural network architectures to achieve better results.

4.1 Baseline (Logistic Regression, SVM, Naive Bayes)

We wanted to form a baseline using traditional machine learning techniques, including logistic regression, support vector machines, and naive bayes. The input to all 3 of these models were the 93 extracted features in the dataset.

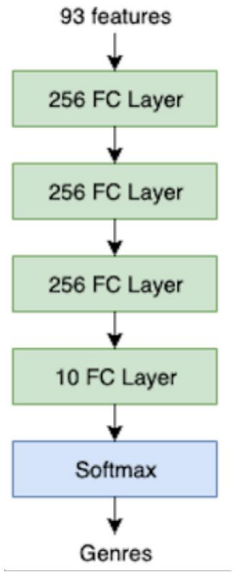

Fig. 2: Fully connected network architecture

Our logistic regression model consisted of 10 different classifiers: one for each genre. To predict the genre of a music piece, the features would be inputted to all 10 classifiers and the sample would be assigned to the genre with highest output probability.

Similar to our logistic regression model, we also had a different classifier for each genre, which totaled to 10 support vector machine classifiers. For our kernel function, we tried polynomial, radial based, and sigmoid functions. RBF attained the highest validation accuracy.
Naive bayes naturally generalizes to multi-classification problems, so we only used 1 model trained on all of the extracted features for the 10 genre classes.

4.2 Fully Connected Network

We wanted to explore neural networks to possibly improve on our previous models that used more traditional machine learning algorithms. Our first approach used a simple fully connected network with four layers and a softmax output layer that is shown in figure 2. Each layer used dropout and a ReLU activation function. We used a cross-entropy loss function with L2 regularization to reduce overfitting.

4.3 Convolutional and Fully Connected Network

To further improve on our neural network approach, we wanted to make use of the Mel Spectrogram through a CNN while also building off of our previous FCC model that used extracted features. To do this, we designed a network that takes both the spectrogram and features as input and has a softmax layer for the genres.

Each block in the architecture, shown in figure 3, is as described below

—Convolution: This block involves a 3x3 filter size with stride 1 over the input. We use ”same” padding on every convolution block, so that each block produces an output with the same height and width dimensions as the input. Each block has its own number of output filters, and has increasingly more number of output filters as the depth increases. We also have a skip connection similar to ResNet's architecture [7] to improve gradient flow. There was also a ReLU activation function after each convolution operation.

—Pooling: The pooling layer allows us to reduce the dimension of the feature map obtained from the convolution step. Every pooling layer had a window size of 2x2 and a stride of 2, which results in an output with half the width and height as the input

—FC Layer: Each fully connected layer used dropout to reduce overfitting and had a ReLU activation layer after.

5. RESULTS

Table 2 summarizes the accuracies for each model.

Model	Validation Accuracy
Logistic Regression	61%
Support Vector Machine (RBF)	65%
Naive Bayes	64%
FCN	68%
CNN + FCN	81%

Table 2. : Validation accuracies of all models

6. DISCUSSION

We noticed that the Mel Spectrograms of the audio pieces contained unique, recognizable patterns for each genre. Additionally, we want to understand our two neural network models and why one performed significantly better than the other. Lastly, we made important design decisions for our final CNN+FCC network based off of our domain knowledge while also leveraging patterns from previous architectures.

6.1 Spectrogram Analysis

As shown in figure 4, the spectograms of each genre showed distinguishable characteristics in its patterns and visual appearance. For example, in the Blues' spectrograms, we can see a checkered pattern, which is distinctly different from the Jazz musics' wavy lines. By inputting this data into our model, it can learn these distinct patterns and produce better results.

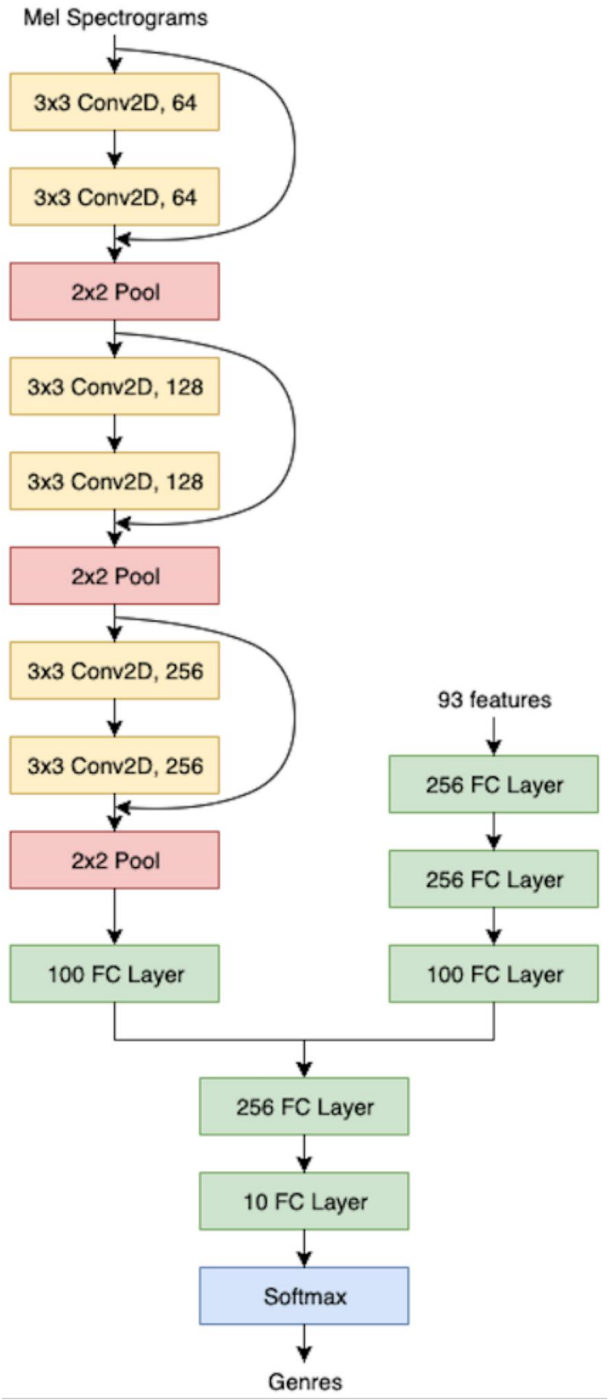

Fig. 3: Convolutional and fully connected network architecture

6.2 Baseline Comparison

Our first non-baseline approach was a simple 4 layered fully connected network performed only 4% better than our baseline models. We believe that this is the case because the 93 input features do not contain enough information about the music samples to accurately

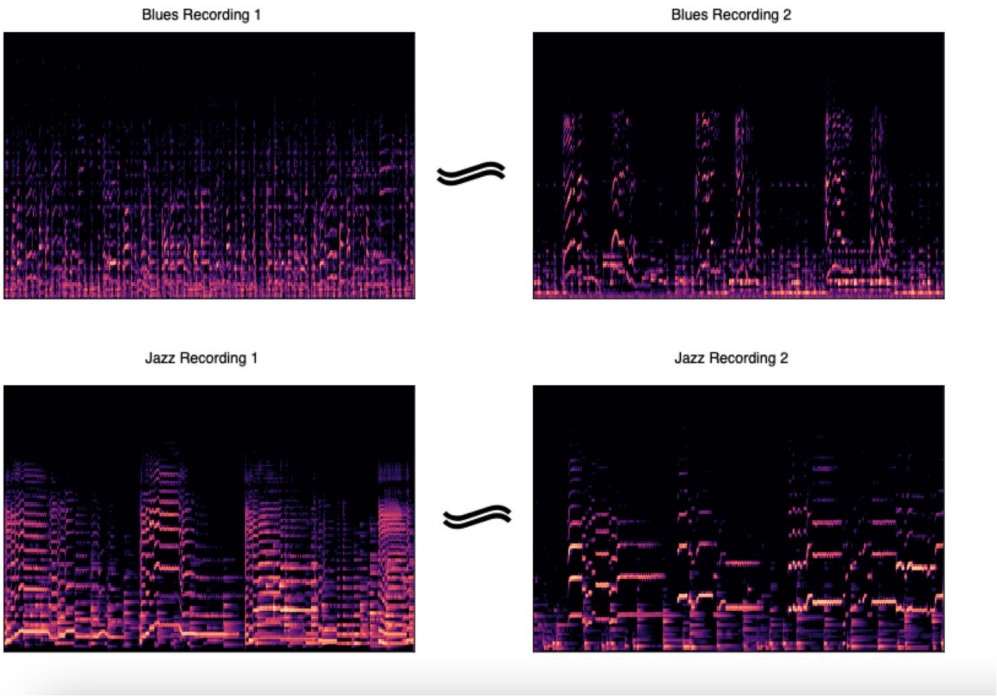

Fig. 4: Mel Spectrograms

classify pieces that are similar to each other. In particular, we found that 31.4% of the wrongly classified pieces were pop and rock genres, which have been historically known to be difficult to classify due to their similar features [6].

The CNN model attained a 13% higher accuracy than all previous models because it was able to use the extracted features along with the spectrogram data. The extracted features provided information about important statistics over the entire music piece, such as timbre and chroma. The spectrogram data contained more detailed information about the frequencies at each timestep. By creating a model that combining both of these important information, we could attain a significant improvement over our baseline models.

6.3 CNN+FCC Model Architecture

We designed our own architecture that uses a CNN and FCC to take both the spectrogram and features as input. For the CNN, we decided to limit the number of layers to reduce the number of parameters and the chances of overfitting. Additionally, it was only important for the CNN to recognize patterns in the spectrogram data and did not need to have a large receptive field. We added skip connections similar to ResNet [7] to help propograte gradients and prevent vanishing the gradient problem. We followed VGG16's model design pattern [8] of doubling the number of output filters after every pooling layer.

The fully connected network part followed a similar architecture to our simple feed forward network design in methods 4.2. The output of the FCC and CNN both had an equal amount of 100 output features so that there was no architectural bias for the spectrograms or the extracted features. Both of these output vectors were concatenated together and inputted into another hidden layer to learn weights before the final output layer.

7. CONCLUSION & FUTURE WORKS

In this work, we explored different methods to classify music into 10 genres based on 93 extracted features and Mel Spectrograms of the audio pieces. We first used traditional machine learning techniques, such as SVM and Naive Bayes, that only used the extract features to form a baseline of 64%. To improve upon this, we used a fully connected network that was able to achieve a 68% accuracy. Our final model was a combination of a fully connected network for the extracted features and a convolutional neural network for the spectrogram. This model achieved an 81% accuracy and outperformed all baselines by 13%.

For future works, using different types of spectrograms altogether rather than just the Mel Spectrogram may produce better results. In this case, we can concatenate all spectrograms together for the input to the CNN and provide even more information to our model than our current approach.

8. REFERENCES

[1] Share of Spotify users in the United States 2013-2020, Statista Research Department, https://www.statista.com/statistics/294640/spotify-listenership-in-the-us/

[2] Tzanetakis, George and Cook, Perry *Musical Genre Classification of Audio Signals* in IEEE Transactions on Speech and Audio Processing, vol.10:293 - 302, 2002

[3] Benetos, Emmanouil and Kotropoulos, C. *A tensor-based approach for automatic music genre classification*, 2008

[4] Aron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu *WaveNet: A Generative Model for Raw Audio, 2016*

[5] Lonce Wyse, *Audio spectrogram representations for processing with Convolutional Neural Networks* in the Proceedings of the First International Workshop on Deep Learning and Music joint with IJCNN, 2017

[6] Nagamanoj Karunakaran and Arti Arya, *A Scalable Hybrid Classifier for Music Genre Classification using Machine Learning Concepts and Spark* at International Conference on Intelligent Autonomous Systems, 2018

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun *Deep Residual Learning for Image Recognition*, 2015

[8] Karen Simonyan and Andrew Zisserman *Very Deep Convolutional Networks for Large-Scale Image Recognition*, 2015