Exploring Machine Learning Techniques Employed in Music Genre Classification

1. Abstract

Music genre classification has become an increasingly useful tool with the advent of streaming services such as Spotify and Apple Music. These music streaming services often employ machine learning techniques in order to categorize music into genres and recommend users new music based on their listening history. The goal of this project is to investigate the machine learning techniques that are often used in the categorization of music into genres. I use the GTZAN dataset to train 8 different models, based on two different modalities—text-based audio features and image-based audio features. Through my investigation of these machine learning techniques, I also seek to learn about the features that can be extracted from an audio signal, how they can be visualized and understood, and improve on existing music genre classification models.

2. Introduction

With the recent rise in the accessibility of music through music streaming platforms, the need for techniques to classify music into genres has also increased. Since most streaming services offer thousands of genres to users around the world, music genre classification has become important for both music lovers and streaming platforms—lovers of music often seek music that is similar in sound to the music they currently listen to, while streaming platforms use these classification techniques to provide users with brand new recommendations on a daily basis.

Additionally, the advancement of programming techniques has opened the door for the exploration of data features across countless modalities—audio feature extraction is no exception to this evolution. With libraries and packages for audio data processing and feature extraction available in most programming languages, implementing techniques for the task of music genre classification has become easier for those interested in investigating this data. At this point, the only real limits to music genre classification: the first is the size of the datasets, and the other is in music copyright laws—a necessary, but inconvenient aspect of music data.

3. Related Work

Tzanetakis and Cook (2002) laid the groundwork for music genre classification when they proposed the use of machine learning methods, such as Gaussian classifiers, Gaussian mixture models, and k-nearest neighbor, to assist in the task of genre classification. In their paper, they discuss features that can be extracted from audio, such as mel-frequency cepstral coefficients, spectral contract, spectral centroids, and many other features that will later be discussed in detail. Furthermore, Alías, Socoró, and Savillano (2016) provide an in-depth analysis of feature extraction of speech, music, and environmental sounds. Their work provides even further insight into the features of audio that are important for music genre classification and other music information retrieval tasks.

Bahuleyan (2018) investigated music genre classification using machine learning techniques. They studied a total of 8 models for 2 modes of data—VGG-16 transfer learning, VGG-16 fine-tuning, and feed-forward neural network using spectrograms as image data; logistic regression, random forest, support vector machines, and extreme gradient boosting using textual audio feature data; and an ensemble classifier using VGG-16 and extreme gradient boosting. The research in this paper was a good starting point for the research in my project. The dataset of 7 genres with a total of 40,504 10-second song clips. Much of the code used for this research was adapted to the dataset employed in this project (GTZAN) which will be discussed in more detail in the dataset section. The results of this article exemplify the effectiveness of the VGG-16 model over the text-based models in music genre classification, achieving a high accuracy of 63% on VGG-16 transfer learning alone. Similarly, research by Haggblade, Hong, and Kao (2011) investigated different machine learning algorithms in the task of music genre classification. The machine learning algorithms they used were k-Nearest Neighbor, k-means, multi-class SVM, and neural networks; the dataset they used was the GTZAN dataset. While the GTZAN dataset has a total of 10 genres, Haggblade et al. only used 4 of the 10 genres, reducing the dataset by 60%. Their models performed with strong accuracy, with the lowest averaging accuracy at 61% for jazz. The result for their neural network achieved an accuracy of 91% in classifying the 4 different genres.

Zhang, Lei, Xu, and Xing (2016) also investigated the use of convolutional neural networks for music genre classification. Their results reinforce the previous research method, showing that convolutional neural networks perform exceptionally well for this task. They achieve a high accuracy of 87.4% using Fourier transform to convert the audio signal into a

visual representation of the frequencies over time (i.e., a spectrogram). They used the GTZAN dataset for training and evaluation of the different networks, with an 0.8/0.1/0.1 train-validation-test split.

4. Dataset

The dataset that I used for my research is the GTZAN dataset, compiled by George Tzanetakis in 2001. This dataset is a compilation of 10 genres, with 100, 30-second song clips per genre, for a total of 1000 30-second song clips. This dataset is often employed in researching music information retrieval tasks such as music genre classification, as could be interpreted from my prior work section. The advantage of using this dataset is that it differs from the dataset employed by Bahuleyan (2018) whose code I have implemented for the text-based models in this project. In that project, a dataset of 40,540 10-second clips of sound with 7 genres sourced from YouTube is employed for the investigation of music genre classification techniques. Both the difference in clip-length and number of genres are beneficial in comparing how many audio features are needed to get good performance in genre classification.

5. Methodology

5.1. Feature Extraction

All of the audio feature extraction performed in this project was completed using the Librosa Python package. Librosa is a Python library that is used specifically for music and audio analysis. For music genre classification, many different functions are used in order to extract different features from an audio source. Features extracted by Librosa can be classified into 3 main categories: temporal features, which represent how the features of an audio signal evolve over time; spectral features, which represent the characteristics of a frequency based signal; and chroma features,

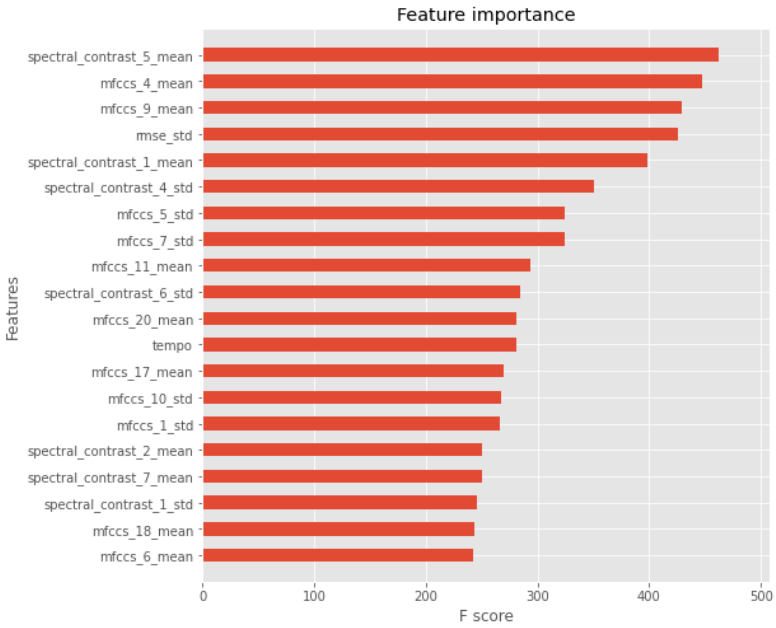

Figure 1: Feature importance based on the XGBoost Music Genre Classification Model. From this graph, we can see that the three most important features for the XGBoost model are spectral contrast, Mel-frequency cepstral coefficients (MFCCs), and root mean square energy (RMSE)

which represent harmonic and melodic characteristics of music. The temporal and spectral features are the main features that I have used in this project. Figure 1 shows the feature importance based on the XGBoost model employed in this project.

5.1.1. Temporal Features

The first feature extraction method employed in my project is Fourier transform. Fourier transform is a mathematical function that converts a waveform (the waveform of music would be soundwaves) into a new representation, often characterized by sine and cosines (Alías et al., 2016). Interestingly, Fourier transform does not apply only to the domain of sound and music. Any type of waveform can be represented in a new form using Fourier transform.

In music genre classification, Fourier transform is employed to visualize how frequencies of sound evolve over a period of time. Different methods can be employed in calculating the Fourier transform of a signal. The most common are linear, logarithmic, and power logarithmic. The linear Fourier transform is able to better represent higher peaks in frequency, at the cost of sparse low-frequency information. The logarithmic Fourier transform accounts for this issue, and is better suited at representing sound as humans perceive it. This is due to the fact that amplitude is typically represented in logarithmic units called decibels, the standard measurement of the intensity of sound (Alías et al., 2016). The power logarithmic function is similar to the logarithmic function, except the input of the frequency is raised to some power before the transform is applied.

The next feature of discussion is CQT transform, which is similar in concept and application to the Fourier transform, especially to the logarithmic Fourier transform (Schörkhuber & Klapuri, 2010). CQT differs from a Fourier transform in that it accounts for the differences between higher and lower frequencies as humans perceive them—that is, lower frequencies have a better spectral resolution (i.e., finer wavelength intervals) than higher frequencies. To make this more clear, it is helpful to understand how frequencies and harmonic characteristics interact. On a standard piano, the lowest note's (A0) frequency is typically around 27.5 Hz. The note immediately following the lowest note (A#0) has a frequency around 29.14 Hz. In contrast, the highest note on a standard piano (C8) has a frequency around 4186 Hz, while the second highest note (B7) has a frequency around 3951 Hz. The difference at the high end of sound frequencies, in my example this is a difference of 235 Hz, is much larger than the

difference at the low end of sound frequencies, which is a 1.64 Hz difference in the prior example. CQT is a valuable resource in representing audio accurately as humans perceive it.

An interesting aspect of all of the transform functions is that the frequency characteristics (the spectral features) are obtained after a transform function has been applied to a given signal (Tzanetakis & Cook, 2002). This gives transform functions the interesting characteristic of converting a signal from the temporal domain into the spectral domain. This makes transform functions vital to the task of music genre classification.

Another important temporal feature that Librosa is able to extract is Zero Crossing Rate (ZCR). ZCR is the rate at which a frequency changes sign (Alías et al., 2016). ZCR is important in distinguishing between voiced, unvoiced, and silent sections of an audio signal. While ZCR is most prominently known for speech recognition and other tasks involving voice, it is also often employed in identifying percussive sounds in music or sections of music where there is a singing voice.

The last feature from the temporal domain is root mean square energy, which is conceptually very similar to the common root mean square error function. Root mean square energy averages the squares of magnitude of the audio frames (Bahuleyan, 2018). In sound-based signals the magnitude is defined in terms of the loudness of a sound, meaning the root mean square energy calculates the average loudness of a signal for each frame.

5.1.2. Spectral Features

The first spectral feature of note is the spectral centroid. The spectral centroid determines the frequency at which the energy of the spectrum is centered at for each frame of a signal (Tzanetakis & Cook, 2002). It essentially works as a weighted mean for each frame in the signal. Spectral bandwidth, another audio feature, uses the spectral centroid of a frame in determining the p'th-order spectral bandwidth of a signal (Alías et al., 2016). Spectral contrast determines the difference between the spectral peaks and valleys of an audio signal at each sub-band of the signal (Alías et al., 2016). Spectral contrast is one of the most important features for the models that I have employed for this project. Spectral contrast varies greatly between different genres of music—this is due to different styles of music using different instruments, tempo, loudness, and other characteristics that define musical genres. Spectral roll-off determines the frequency at which a given percentage (typically 85%) of the total spectral energy is concentrated (Tzanetakis & Cook, 2002).

The second most important spectral feature after spectral contrast is Mel-frequency cepstral coefficients (MFCCs). In order to understand the concept of MFCCs, it is important to know what the Mel scale is. The Mel scale is defined as a perceptual scale of pitches judged by a listener to be equally distanced from one another (Alías et al., 2016). For music enthusiasts, an easy way to understand the Mel scale is thinking about whole steps and half steps in music. Half steps (semitones) are essentially the equally distanced pitch unit—going up a half step from the music note E would bring you to an F, which is 'equally distant' to going from another arbitrary note, say an A to an A#. MFCCs are important for music genre classification as different genres typically follow different guidelines when considering keys and musical theory. Jazz, for example, has much less strict 'rules' than pop music; jazz typically encourages players to improvise, under the guise that no note is a wrong note, even if it is not a part of the melodic scale that the song employs. Pop, on the other hand, is much more formulaic, where typically the song structures in different pop songs are similar or identical, and the notes played follow the rules of the melodic scale employed. MFCCs are especially useful in distinguishing between genres from around the world—in Western music, the Mel scale is interpreted in half-steps, whereas in other parts of the world, quarter-steps are used (quarter-steps are very uncommon for the most part in Western-based music).

The last spectral based feature of major importance is the Spectrogram. Spectrograms visualize the frequencies of a signal as they evolve over time (Alías et al., 2016). A spectrogram can be produced from a transform function or from spectral features. Not all spectrograms represent the same features of audio—creating a spectrogram based on spectral contrast would look different than a spectrogram based on Root Mean Square Energy, as can be seen by comparing Figures 2 and 3. Again, this is due to spectrograms being able to visualize a multitude of audio features. Spectrograms are important in music genre

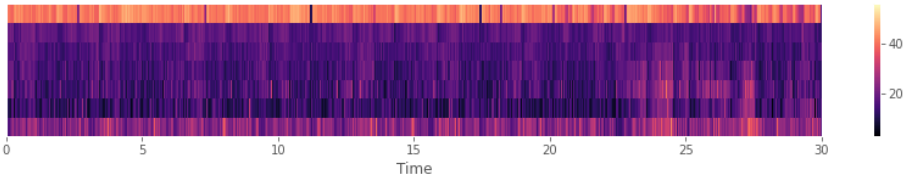

Figure 2:Spectrogram Produced from Spectral Contrast Features

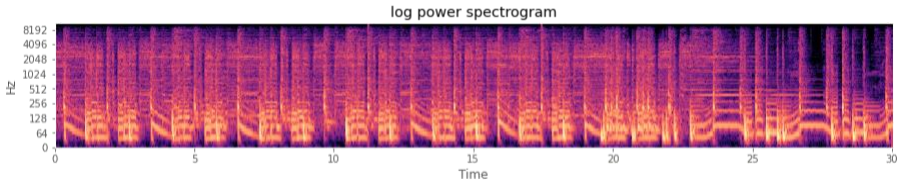

Figure 3: Spectrogram Produced from Root Mean Square Energy (RMSE)

classification because they can be fed to a convolutional neural network (or any computer vision-related classifier) to learn the audio features from the spectrogram.

5.2. Machine Learning Models

In my research for this project, I found that there are two main methods employed for the task of music genre classification—textual-feature based models, which take multiple forms of audio features extracted using the Librosa library in a text based format to train and predict genres; and image-feature based models, which take spectrograms produced by visualizing a type of audio feature as input to extract the visual data in order to learn and predict genres. I employed a total of 8 models in my investigation, 4 in the textual feature domain and 4 in the image feature domain.

5.2.1. Models Based on Textual Audio Features

The models that I used for the text-based audio features are directly from Bahuleyan (2018). I believed the ensemble of models from this paper were a great representation of machine learning techniques that were taught in class. I explored further machine learning models in the spectrogram based models, as will be discussed in further detail in the next section.

The first model based on textual audio features is a simple logistic regression model. The model comes from the Scikit Learn Python package. This logistic regression model employs a one-versus-rest scheme for multiclass classification problems. For my experiments, I used the newton-cg solver with an L2 penalty.

The next model based on textual audio features is a random forest model. Again this model comes from the Python package sci-kit learn. Random forest is a bagging method that employs multiple, independent decision trees in order to make decisions using a majority-decision vote. For my project, I used a model with 500 estimators, and a minimum of 5 samples split.

Following the previous model is XGBoost, which stands for eXtreme Gradient Boosting, and this model comes from the XGBoost Python package. XGBoost is a boosting method that employs multiple decision trees to make a decision, similar to the previous model, random forest. Unlike random forest, the decision trees are trained in a dependent manner—the learners consider previous models and determine areas where the prior models underperformed and adjust accordingly. Again, I used a model with 500 estimators, but with a max depth of 5.

Finally, the last model trained on textual audio features is an ensemble of Support Vector Machines, using the SVC model from the sci-kit learn Python package. The SVMs are employed similar to the logistic regression model, using a one-versus-rest scheme for the multiclass label problem. The kernel used for this task is the radial based kernel since audio waves as interpreted by humans are on a logarithmic scale.

5.2.2. Models Based on Spectrogram Image Features

For the spectrogram image feature based models, I employed 4 different convolutional neural networks that are implemented in the Keras Python library. Convolutional neural networks are most often used in the domain of computer vision, where they are typically employed for image classification tasks. CNNs work well for music genre classification because the spectrograms for each genre are typically quite distinct, as can be seen in Figure 4. Another advantage of using convolutional neural networks is that they can work directly with time series data (as music inherently is), even for low-dimensional inputs.

The first of the three basic CNN models that I have employed is a 1-dimensional CNN that takes in a 1-dimensional sequence of data. The spectrograms are reduced to a single dimensional array of data in which the network reads and interprets the different time series data that the spectrogram produces. The advantage of the 1-dimensional CNN is its quick execution time, which will be discussed further in the experimental results section.

The second basic CNN model that I employed was a 2-dimensional CNN that takes in 2-dimensional sequences of data. Similar to the 1D CNN, the spectrograms are reduced to 2-dimensional arrays of data that are passed to the CNN, which will interpret the time series features from the spectrograms.

The last basic CNN model I employed was a convolutional recurrent neural network

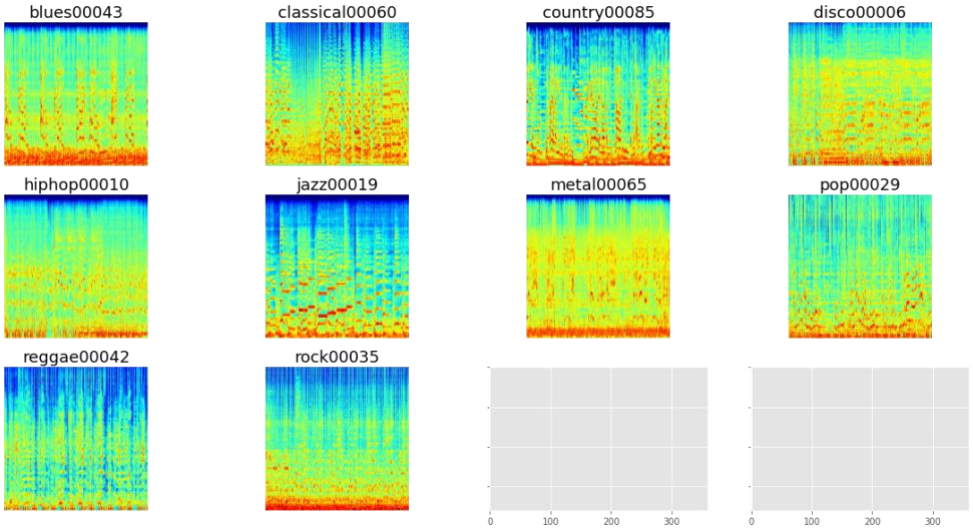

Figure 4: Example Spectrograms for each genre in the GTZAN dataset. These spectrograms clearly show how the frequencies in different genres of music produce different spectrograms.

the benefit of this is that the recurrent neural network can make predictions based on previous data. This gives it more potential to perform well, given that it can calculate the best decision given the decision history.

The final model I employed for the spectrogram image feature models was the VGG16 model from the Visual Geometry Group from Oxford University. This is a pre-trained CNN model that uses transfer learning to learn the different genre classifications. Transfer learning is when a pre-trained model is employed using either a partial number of original layers from the source model or sometimes all of the original layers. The model is then refined to adapt to the new type of data. VGG-16 is a good network to use for music genre classification as it has performed with promising results on other, similar datasets to the one I have used in this project. For my experiments, I used a feedforward transfer learning method with the VGG16 convolutional base.

6. Experimental Results

6.1. Metrics

The metrics that I used for the evaluation of my results are as follows.

1. Accuracy: The number of correctly predicted music genres out of all predictions
2. AUC ROC: Performance measurement for classification at various threshold settings
3. Per Class Precision: Positive class predictions that truly belong to the positive class
4. Per Class Recall: Positive class predictions out of all positive examples

6.2. Results

After training for a maximum of 100 epochs, I performed testing on all of my methods. To my surprise, the text based features outperformed the spectrogram based features in all cases. Results shown in the figures below are based on the best performing models from each training epoch.

Data Format	Model	Train Accuracy	Validation Accuracy	Test Accuracy	AUC ROC
Image	1-D CNN	0.286	0.347	0.34	0.569
	2-D CNN	0.185	0.233	0.24	0.648
	CRNN	0.237	0.30	0.22	0.886
	VGG-16	0.58	0.55	0.61	0.888
Text	Logistic Regression	0.85	N/A	0.75	0.953
	Random Forest	0.99	N/A	0.78	0.956
	XGBoost	0.99	N/A	0.71	0.952
	SVMs	0.99	N/A	0.77	0.969

Figure 5: Accuracy and AUC ROC of each model. The text-based models did not provide a way to use a validation set, so they were trained using a 85/15 train-test split. The text-based models were able to achieve much higher accuracy than the spectrogram-based models while achieving faster execution, training, and testing times.

7. Discussion of Results

I will preface the discussion of my results with the challenges I faced in finding implementations for the models with up-to-date libraries. All of the code that I was able to find used outdated Python packages and libraries—this would not have been an issue had the versions all been consistent. Specifically, for the CNN based models (excluding VGG-16), I had to reconfigure each of the parameters, audioutils, audiostructs, and audiomodels files, with little to no documentation in the code to work with. Luckily, the documentation for packages like scikit learn, matplotlab, and others made this challenge easier to overcome. After hours of reworking the code, I was able to get the models to run without any errors. I have had little experience with computer vision tasks in Python, so configuring the CNNs to run on my input was also a challenge that will soon be discussed.

Figure 5 shows the accuracies and AUC ROC scores for each model. As I previously stated, the textual feature based models outperformed the spectrogram based features in all cases by a pretty significant margin. The difference between the highest accuracy of the text-based (random forest) and spectrogram-based (VGG-16) models is 17%. There are a handful of possible reasons for this wide difference. The first possibility is error in updating the code for the CNN models (again excluding VGG-16). As I previously stated, I ran into a lot of errors when first compiling the CNN code to run with my data. After reconfiguring, I was able to get them to run, but they were not getting nearly as good accuracy as other research articles that I have read. This would explain why VGG-16 outperformed the other models by such a wide margin—the average difference between the VGG-16 accuracies and the best performing, 1-dimensional CNN accuracies is 25.6%, a margin wider than the VGG-16 model compared to the text-based models.

Regardless of the possibility of said error, I believe that the 1-dimensional CNN would outperform the 2-dimensional and CRNN because the flattening of the data into 1-dimension could potentially be a better representation for the time series signal's frequencies as they evolve over time. Additionally, the 1-dimensional CNN had very low training and testing times, on the magnitude of seconds.

Another possible explanation for the wide gap between accuracies amongst the image and text based models is that I used a different dataset in my research than what was used in Bahuleyan (2018)—their performance on the text based models was worse than the image based models. Interestingly, I was able to achieve better testing accuracy in all of the same text based models than any of the models in the previously listed paper. My hypothesis is that this could be that the GTZAN dataset uses 30-second audio clips, whereas the dataset used in Bahuleyan (2018) uses a larger dataset with only 10-second audio clips—I believe that the longer audio clips are able to give more distinguishing per-genre features that are important in classifying music for these models.

Another hypothesis that I have regarding the differences in the accuracies between the spectrogram-based and text-based models is that the text-based models performed so much better due to using an ensemble of audio features to interpret musical genre. The spectrogram-based models rely on a single type of feature in the form of a spectrogram in order to determine music genre. This could give the spectrogram-based models a disadvantage because they do not work with as many distinct features.

		Precision									
		Genres									
Data Format	Model	Blues	Classical	Country	Disco	Hiphop	Jazz	Metal	Pop	Reggae	Rock
Image	1-D CNN	0.0	0.222	0.429	0.5	0.0	0.333	0.5	0.444	0.5	0.143
	2-D CNN	0.143	0.0	1.0	0.333	0.2	0.125	0.1	0.139	0.0	0.0
	CRNN	0.188	0.0	0.455	0.368	0.0	0.333	0.833	0.333	0.0	0.0
	VGG16	0.70	0.588	0.563	0.571	0.722	0.5	0.677	0.35	1.0	0.714
Text	Logistic Regression	0.737	1.0	0.471	0.75	0.733	1.0	0.905	0.714	0.545	0.643
	Random Forest	0.929	0.813	0.571	0.706	0.923	1.0	1.0	0.818	0.5	0.667
	XGBoost	0.611	0.867	0.529	0.632	0.75	1.0	0.947	0.80	0.60	0.438
	SVMs	0.917	0.812	0.647	0.737	0.923	1.0	0.95	0.70	0.412	0.667

Figure 6: Precision of each model on each class in the GTZAN dataset. Text-based methods achieved much higher and more consistent precision than spectrogram-based models

The precision results in Figure 6 shows each of the models' precision for each genre in the dataset; the confusion matrices for the best performing models in each mode are in Figure 7. There are a few interesting things to note about this table. The first is that the textual based models were able to achieve perfect precision for the jazz genre. Jazz was the only genre that I felt confident at least one classifier would get 100% precise, because, as I previously stated, the genre of jazz is typically quite improvisational and breaks the traditional rules of music. Metal, which is also another genre with very distinct characteristics, also got high precision in all of the text based models, and classical lagging behind in third place. For the spectrogram based models, country achieved the most consistent precision across all models. The precision in VGG-16 is much higher compared to the other spectrogram-based models. Figure 8 shows each of the models' recall for each genre in the dataset. Interestingly, the text-based models achieved perfect recall in the classical genre. Metal achieved similar recall as it did precision. The third highest recall in the text-based models is blues. It is interesting, yet not necessarily surprising that jazz's recall was not as high as its precision for text-based models—I think that this is due to the variation in jazz that I've previously covered. It makes sense that the models would correctly identify jazz songs while also labeling other songs as jazz. The genre with the most consistent recall across the spectrogram based images was metal, while disco and country were not too far behind.

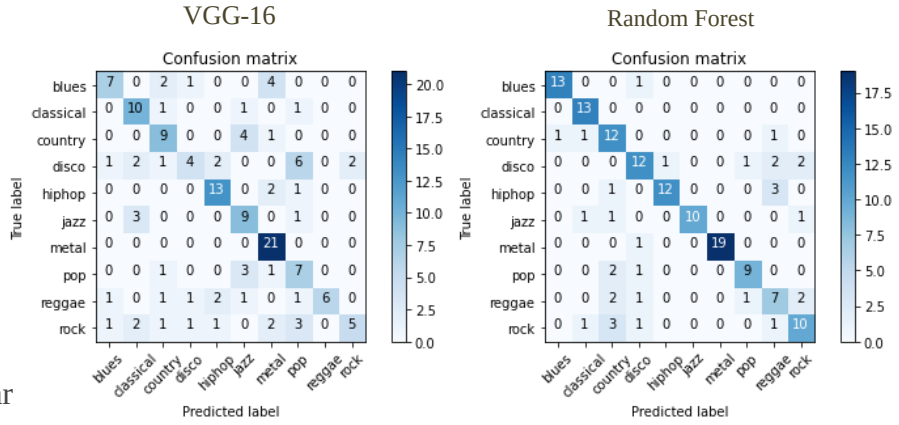

Figure 7: Confusion Matrix on the Left is for VGG-16. Confusion Matrix on the Right is for Random Forest

		Recall									
		Genres									
Data Format	Model	Blues	Classical	Country	Disco	Hiphop	Jazz	Metal	Pop	Reggae	Rock
Image	1-D CNN	0.0	0.211	0.581	0.5	0.0	0.364	0.545	0.421	0.444	0.167
	2-D CNN	0.1	0.0	0.2	0.2	0.2	0.1	0.3	0.5	0.0	0.0
	CRNN	0.6	0.0	1.0	0.7	0.0	0.5	0.5	0.1	0.0	0.0
	VGG16	0.5	0.769	0.643	0.222	0.813	0.692	1.0	0.583	0.462	0.313
Text	Logistic Regression	1.0	1.0	0.533	0.667	0.688	0.769	0.95	0.833	0.462	0.563
	Random Forest	0.929	1.0	0.8	0.667	0.75	0.769	0.95	0.75	0.538	0.625
	XGBoost	0.786	1.0	0.6	0.667	0.75	0.769	0.9	0.667	0.462	0.438
	SVMs	0.786	1.0	0.733	0.778	0.75	0.846	0.95	0.583	0.538	0.625

Figure 8: Recall measure of each model on each class in the GTZAN dataset. Text-based models achieved higher recall scores than image-based models

8. Future Work

With the poor results I received in the spectrogram-based models, I would be interested in doing further research in the domain of computer vision based models for music genre classification. This research would give me a broader understanding of machine learning techniques across more domains, while also investigating the potential for improvement in existing image-based models.

Another domain of interest is in cross-modal classification techniques. Prior work in this area exists, however cross-modal classification techniques proved challenging for me to implement, but given more time it would be interesting to learn how to build a multi-modal model and evaluate its performance. Given that my text-based models performed so well, I believe that combining the two modalities has the potential to achieve better performance over single modality models. Oramas, Barbieri, Nietto, and Serra (2018) investigated deep learning models for music genre classification using multimodal data and their results showed that the accuracy of the cross-modal method (combining an audio based CNN with an image based CNN) underperformed compared to a human annotator. However, their results did show improvement in the average accuracy when both audio and visual data are used for music genre classification. Their audio based model performed with 35% accuracy and their visual data only achieved a 25% accuracy; however, when they combined the two methods, the audio and visual network achieved a 43% accuracy. This research leads me to believe that there is

Further research would also involve studying the effect of the audio clip length's effect on model performance. Given more time, it would have been interesting to study this effect by clipping audio signals to different lengths and comparing the performance against different models for different clip-lengths. This research could also be beneficial in music genre classification because determining the clip-length that maximizes the features of any given genre could help reduce performance costs and execution time, especially in spectrogram-based models.

9. Conclusion

Music genre classification using machine learning techniques has become a subtle, yet core component for many people's music listening experience. Streaming services rely on these techniques to provide users with new recommendations based on songs from their listening history, while also categorizing new music into accurate genres. As I have come to learn through my research, machine learning models use features such as mel-frequency cepstral coefficients, root mean square energy, spectral contrast, and other sound frequency characteristics in order to achieve this goal. My research has also shown the potential for text-based methods in the task of music genre classification. While my spectrogram-based models performed poorly, research in other papers has shown that image-based models show potential to outperform text-based models. Further work would include combining the two modes used in this project to work together for the music genre classification task.

References

Alías, F., Socoró, J. C., & Sevillano, X. (2016). A review of physical and perceptual feature extraction techniques for speech, music and environmental sounds. *Applied Sciences*, *6*(5), 143.

Bahuleyan, H. (2018). Music genre classification using machine learning techniques. *arXiv preprint arXiv:1804.01149*.

Haggblade, M., Hong, Y., & Kao, K. (2011). Music genre classification. *Department of Computer Science, Stanford University*, *131*, 132.

Oramas, S., Barbieri, F., Nieto Caballero, O., & Serra, X. (2018). Multimodal deep learning for

music genre classification. *Transactions of the International Society for Music Information Retrieval. 2018; 1 (1): 4-21.*

Schörkhuber, C., & Klapuri, A. (2010, July). Constant-Q transform toolbox for music processing. In *7th Sound and Music Computing Conference, Barcelona, Spain* (pp. 3-64).

Tzanetakis, G., & Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, *10*(5), 293-302.

Zhang, W., Lei, W., Xu, X., & Xing, X. (2016, September). Improved Music Genre Classification with Convolutional Neural Networks. In *Interspeech* (pp. 3304-3308).