Music Genre Classification

Rui Ziqi
1023041115
Nanjing University of Posts and Telecommunications
Jiangsu, Nanjing China

***Abstract*—Music classification is a fundamental task in the field of music information retrieval (MIR). It involves automatically categorizing music tracks into predefined genres or other relevant classes. This paper presents a music classification program that utilizes machine learning techniques to achieve accurate and efficient classification results. This project uses the GTZAN dataset. The audio tracks are represented by various feature representations such as Mel-frequency cepstral coefficients (MFCCs), spectral features, and rhythmic patterns. The classification model employed in the program has shown promising results in various audio and image classification tasks. The model is trained using a supervised learning framework, where the ground truth genre labels are used to optimize the model parameters. To judge the overall performance, a confusion matrix is produced. A confusion matrix is a specific table layout that allows visualization of the performance of an algorithm. The Mel Frequency Cepstrum (MFC) encodes the power spectrum of a sound. It is calculated as the Fourier transform of the logarithm of the signal's spectrum. This music classification program showcases the potential of machine learning and deep learning techniques in automating the categorization of music tracks, offering valuable insights into the rich field of music analysis and organization.**

Keywords—music classification, MFCC, spectrogram, genres

I. INTRODUCTION

Music is an integral part of human culture, and with the advent of digital music platforms and streaming services, the availability of music has skyrocketed. Music genre classification is one of the most important tasks in music information retrieval studies. There have been many trials to improve the accuracy of this task. Classifying music is a fairly complicated task because there are some many classes to consider with subtle differences between some of them. Some genres of music are hybrids of other genres so it is very difficult to separate one type of genre from another. [1] The motivation behind this project stems from the desire to leverage machine learning and signal processing techniques to automate the music classification process. By developing an accurate and efficient music classification system, it can enhance music organization, and recommendation experiences.

The project builds upon previous research and advancements in the field of music information retrieval (MIR) and audio signal processing. Various approaches, such as feature extraction techniques, machine learning algorithms, and deep learning architectures, have been explored to extract meaningful representations from audio signals and enable effective music classification. However, there is still room for improvement in terms of accuracy, robustness, and scalability.

The primary objective of this project is to develop a music classification system that can accurately categorize music tracks into different genres. It aims to explore and compare different feature representations, such as Mel Frequency Cepstral Coefficients (MFCCs), spectral features, and rhythmic patterns, and evaluate their effectiveness for genre classification. This paper use the GTZAN dataset which is a widely recognized and extensively used benchmark dataset in the field of music information retrieval (MIR). It has been a fundamental resource for evaluating and comparing various algorithms and techniques for tasks such as music genre classification, music mood recognition, and audio analysis.

By addressing these research objectives, there are several potential benefits. Firstly, an accurate music classification system can improve the user experience on music platforms by enabling personalized recommendations and tailored playlists. Secondly, it can assist music industry professionals in music cataloging, copyright enforcement, and market analysis. Finally, the project's findings can contribute to the broader field of music analysis and signal processing, advancing the understanding of music perception and organization.

In the following sections, the paper will present the data preprocessing, related works about music genres classification, describe the solutions used and the experimental results, and analyze the evaluation of this approach.

II. DATA PREPROCESSING

A. GTZAN Dataset

The GTZAN dataset is a widely recognized and extensively used benchmark dataset in the field of music information retrieval (MIR), which was curated by George Tzanetakis and Perry Cook with the aim of creating a standardized and diverse dataset for music genre classification[2]. Musical genres are categorical labels created by humans to characterize pieces of music. A musical genre is characterized by the common characteristics shared by its members. These characteristics typically are related to the instrumentation, rhythmic structure, and harmonic content of the music.

The audio tracks within the GTZAN dataset were sourced from commercial music collections, ensuring a diverse selection of artists, recordings, and production styles. This diversity allows researchers to explore and analyze the characteristics and nuances of different music genres. The dataset consists of 1000 audio tracks each 30 seconds long. It contains 10 genres, each represented by 100 tracks. The tracks are all 22050Hz Mono 16-bit audio files in .wav format.

Label	FP IDed	By hand	In last.fm	Tags
Blues	63	96	96	1549
Classical	63	80	20	352
Country	54	93	90	1486
Disco	52	80	79	4191
Hip hop	64	94	93	5370
Jazz	65	80	79	4191
Metal	65	82	81	4798
Pop	59	96	96	6379
Reggae	54	82	78	3300
Rock	67	100	100	5475
Total	60.6	88.3	80.9	33814

Fig.1. Percentage of GTZAN excerpts identified

B. *Timbral Texture Features*

Timbral texture features are a set of audio features that capture the perceptual characteristics and qualities of sound related to its timbre and texture. Timbre refers to the unique quality or color of a sound that distinguishes it from others, while texture refers to the arrangement and composition of different timbres within a sound or music piece.

The features used to represent timbral texture are based on standard features proposed for music-speech discrimination [3] and speech recognition [4]. The calculated features are based on the short time Fourier transform (STFT) and are calculated for every short-time frame of sound. More details regarding the STFT algorithm and the Mel-frequency cepstral coefficients (MFCC). While primarily used for capturing spectral shape information, MFCCs can also provide insights into timbral texture. Higher-order coefficients beyond the first few static coefficients capture temporal variations and texture-related information.

C. *Spectrograms*

A spectrogram is a visual representation of the frequency content in a song. It shows the intensity of the frequencies on the y axis in the specified time intervals on the x axis; that is, the darker the color, the stronger the frequency is in the particular time window of the song. It provides a detailed view of how the spectral content of a signal changes over time, allowing for the analysis and visualization of complex audio signals.

To generate a spectrogram, we should follow these steps. Divide the audio signal into small overlapping frames. Each frame is multiplied by a windowing function, such as the Hamming or Hanning window, to reduce spectral leakage and ensure smooth transitions at frame boundaries. A Fourier Transform is applied to each windowed frame to obtain the frequency spectrum. The magnitude spectrum represents the energy distribution across different frequency bins. The squared magnitude of the spectrum is calculated, resulting in the power spectrum. And the power spectra of consecutive frames are typically displayed as a 2D image.

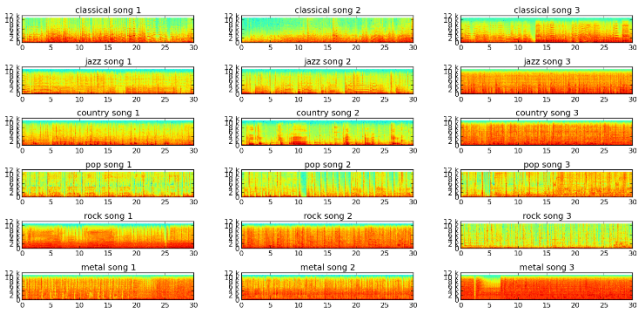

Fig.2. Sample spectrograms of a few songs from the GTZAN dataset

It can be clearly seen from the above image that songs belonging to the same genre have similar spectrograms. Keeping this in mind, we can easily design a classifier that can learn to differentiate between the different genres with sufficient accuracy.

D. *Using MFCC*

Mel Frequency Cepstral Coefficients (MFCCs) are a set of features just like chroma or spectral, which was developed at MIT during the late 1960s to study the echoes in seismic audio. Now it is widely used audio features that capture the spectral shape information of a sound signal. They are particularly effective in representing the perceptual characteristics of human hearing and have found extensive applications in speech and music processing tasks.

The Mel Frequency Cepstrum (MFC) encodes the power spectrum of a sound. It is calculated as the Fourier transform of the logarithm of the signal's spectrum. The cepstral coefficient features are computed using a seven-step process. First, the signal is pre-emphasized, which changes the tilt or slope of the spectrum to increase the energy of higher frequencies. Next, a Hamming window is applied to the frame; a Hamming window reduces the effects of speech at the edges of the window, which is useful in obtaining a smooth spectral representation. Third, the power spectrum is computed, without taking the log operation. A filter-bank operation is applied to the power spectrum, thereby measuring the energy in different frequency bands. The frequency bands are spaced along the frequency axis according to the perceptually based nonlinear Mel scale, in which higher frequencies are represented with lower resolution. In the fifth step, the log of the energy in these frequency bands is computed. In the sixth step, the spectral-domain representation is translated into the cepstral domain; cepstral features can be obtained by taking the inverse Fourier transform of the log-power spectrum. The cepstral domain representation has the advantage that the features are less correlated, which is important in efficient implementation of GMM-based classifiers. Finally, the cepstral features are weighted so that the range of all feature values is approximately equal. This weighting is useful in implementation of the classifier, although theoretically not required.

The Talkbox SciKit (scikits.talkbox) contains an implementation of MFC that we can directly use. The data that feed into the classifier is stored as *ceps*, which contain 13 coeffecients to uniquely represent an audio file.

III. RELATED WORK

At the beginning of MIR research, researchers used traditional machine learning methods to solve the problem of music genre classification. The typical process included three steps: (1) extracting timbre from the original audio signal, spectral time and statistical features; (2) adopt some techniques to select meaningful subsets of features[5] or aggregate features[6] to improve classification accuracy; (3) use machine learning classifiers to train selected objects.

In 2002, Tzanetakis et al.[2]obtained underlying acoustic feature sets from 10 types of music files of different styles, and then trained three types of Gaussian classifiers, Gaussian Mixture Model GMM (Gaussian Mixture Model) and K-Nearest Neighbor KNN (K-Nearest Neighbor) classifier, released on the widely used GTZAN dataset, and achieved an average classification accuracy of 61% on this dataset. Since then, the GTZAN dataset has been widely used for music genre classification tasks. In 2003, Li et al.[7] used Daubechies wavelet coefficient histograms to capture local and global information of music for genre classification, and compared various traditional machine learning methods. The biggest drawback of traditional methods is that they need to manually select and extract music signal features. The feature extraction process is complex and has poor versatility, which leads to problems such as single music signal features, unstable model performance, and poor generalization.

With the gradual development of deep learning, neural networks have achieved good results in many fields including MIR. Convolutional neural networks CNN and recurrent neural networks RNN are widely used in music genre classification. Allamy et al. [8] use one-dimensional convolutional neural network to classify the time-frequency characteristics of audio signals. Choi et al. [9] combined CNN and RNN and introduced the two-dimensional convolutional recursive neural network CRNN (Convolutional Recurrent Neural Network). CRNN uses CNN to extract local features, and then uses RNN to summarize the extracted features over time. Sigtia et al. [10] studied the use of convolutional neural networks to improve audio data. According to the three methods of feature learning, the classification accuracy on the GTZAN data set reached 83%. Fulzele et al. [11] used the Long Short-Term Memory LSTM (Long Short-Term Memory) neural network for music type classification, and combined it with a support vector machine classifier to enhance its performance. The accuracy rate reached 89% on the GTZAN data set.

This also shows the potential of CNNs and RNNs in music content analysis. But at the same time, they also have problems such as large amount of data required, difficulty in global feature extraction and long-range dependence.

This project uses python especially the NumPy, SciPy and scikit-learn libraires. Some of the key libraries used in the project and making the final model are listed below.

1. NumPy

NumPy (Numerical Python) is a fundamental library for scientific computing in Python. It provides powerful data structures, array manipulation capabilities, and a collection of mathematical functions that allow efficient numerical operations on large datasets.

2. SciPy

SciPy is an open-source library built on top of NumPy that provides a wide range of scientific and numerical computing capabilities. It extends the functionality of NumPy by offering additional modules and functions for tasks such as optimization, interpolation, signal processing, linear algebra, statistics, and more. SciPy is designed to be a comprehensive library for scientific computing in Python, providing efficient and reliable algorithms for various scientific and engineering applications.

3. Scikit-learn

Scikit-learn, also known as sklearn, is a popular open-source machine learning library for Python. It provides a wide range of tools and algorithms for various machine learning tasks, including classification, regression, clustering, dimensionality reduction, model selection, and preprocessing of data.

IV. MUSIC CLASSIFICATION

A. Experiment Setup

First of all, this project needs to have the following requirements on the Python environment: NumPy, PyDub, SciPy, scikit-learn, scikits.statsmodels and scikits.talkbox.

All the configuration can be done using the config.cfg file. This file follows a particular syntax for storing the configurations. This file contains three variables that the user can modify as per his need. Comments begin with a # symbol and run till the end of a line. Rest everything is supposed to be valid configuration data. The variables are:

- GENRE_DIR - This is directory where the music dataset is located (GTZAN dataset).
- TEST_DIR - This is the directory where the test music is located.
- GENRE_LIST - This is a list of the available genre types that you can use. Modify this list if you want to work with a subset of the available genres.

Set these three variables according to your system before proceeding to the next steps. The dataset used for training the model is the GTZAN dataset, which consists of 1000 audio tracks each 30 seconds long. It contains 10 genres, each represented by 100 tracks. The tracks are all 22050Hz Mono 16-bit audio files in .wav format. Since the files in the dataset are in the au format, which is lossy because of compression,

they need to be converted in the wav format (which is lossless) before proceed further.

The script ceps.py analyzes and converts each file in the GTZAN dataset in a representation that can be used by the classifier and can be easily cached onto the disk. This little step prevents the classifier to convert the dataset each time the system is run. The GTZAN dataset is used for training the classifier, which generates an in-memory regression model. This process is done by the LogisticRegression module of the scikit-learn library. The classifier.py script has been provided for this purpose. Once the model has been generated, we can use it to predict genres of other audio files. For effecient further use of the generated model, it is permanently serialized to the disk, and is deserialized when it needs to be used again. This simple process improves performance greatly. For serialization, the joblib module in the sklearn.externals package is used. As of now, the classifier.py script must be run before any testing with unknown music can be done. Once the script is run, it will save the generated model. Once the model has been saved, the classification script need not be run again until some newly labelled training data is available.

The tester.py script is used for the classification of new and unlabelled audio files. This script deserializes the previously cached model and uses it for classifying new audio files.

B. Result

When the classifier.py script is run, it generates and saves the trained model to the disk. This process also results in the creation of some graphs which tell the performance of the classification process.

1. ROC Curves

For each selected genre type, a ROC (Receiver Operating Characteristic) curve is generated. The ROC curve is created by plotting the fraction of true positives out of the total actual positives (True positive rate) vs. the fraction of false positives out of the total actual negatives (False positive rate), at various threshold settings. Some of the sample graphs are shown below along with their proper interpretation.

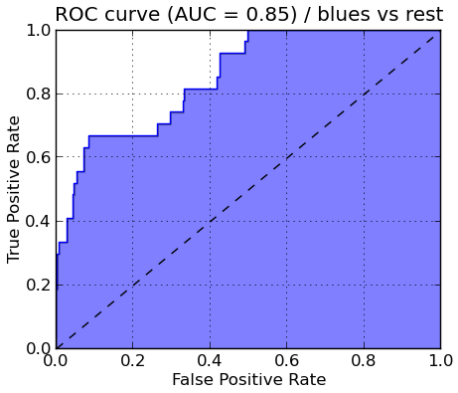

Fig.3. ROC curve of BLUES genre

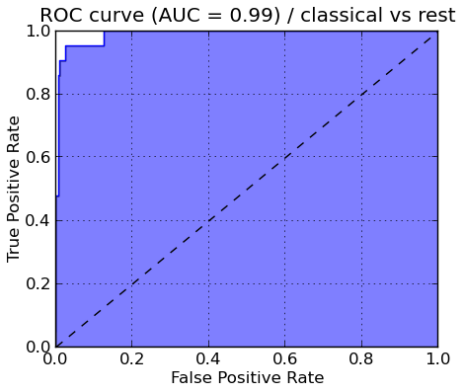

Fig.4. ROC curve of CLASSICAL genre

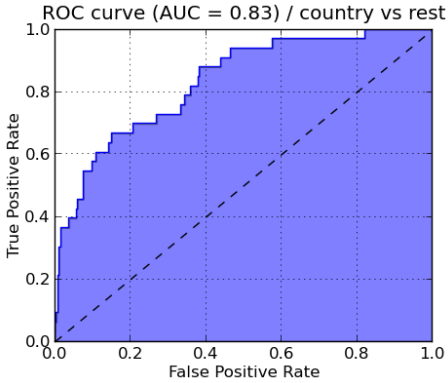

Fig.5. ROC curve of COUNTRY genre

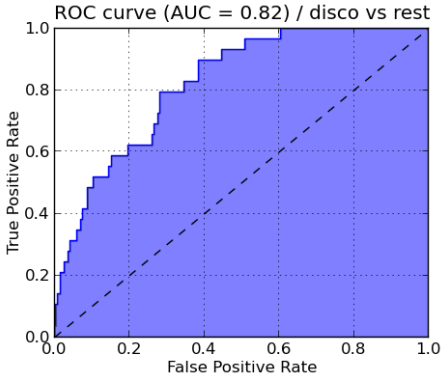

Fig.6. ROC curve of DISCO genre

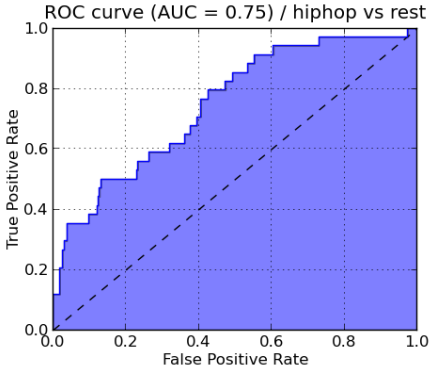

Fig.7. ROC curve of HIPHOP genre

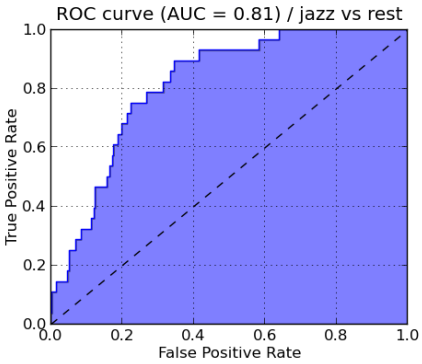

Fig.8. ROC curve of JAZZ genre

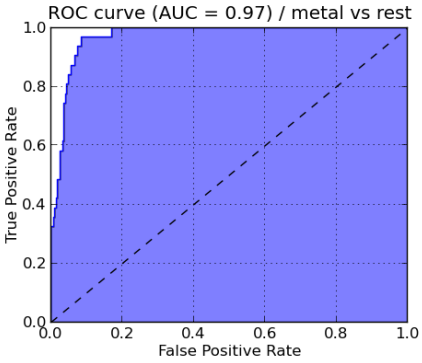

Fig.9. ROC curve of METAL genre

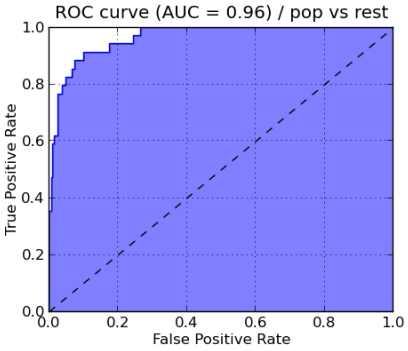

Fig.10. ROC curve of POP genre

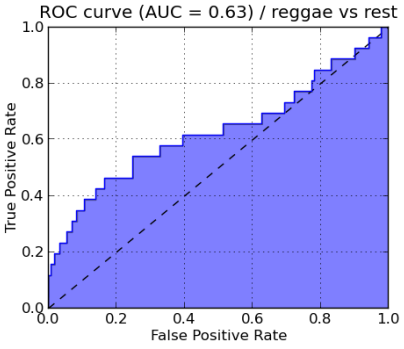

Fig.11. ROC curve of REGGAE genre

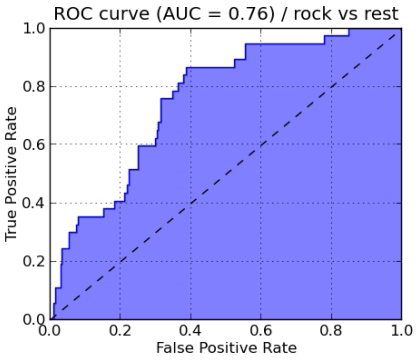

Fig.12. ROC curve of ROCK genre

2. Confusion Matrix

To judge the overall performance, a confusion matrix is produced. A confusion matrix is a specific table layout that allows visualization of the performance of an algorithm. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. The confusion matrix with all the genres selected is shown below.

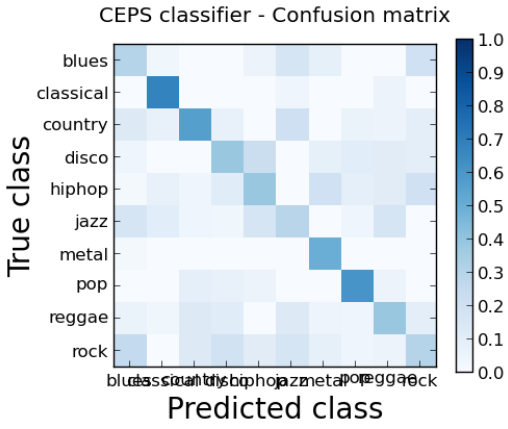

Fig.13. Confusion Matrix

CONCLUSION

In this paper, I try to use the knowledge of data mining and spectrogram representations to build a music genre classification system. The goal of this project is to accurately classify music into different genres through extracting informative features from audio signals using MFCCs and trained a model. However, there is still room for improvement in terms of accuracy, robustness, and scalability.

Throughout the semester of studying Big Data, I gained valuable insights into data representation and effective approaches for extracting meaningful information. It is with great appreciation that I acknowledge the guidance and expertise of my esteemed professor, Prof. Zou. The homework examples, particularly those related to music classification, significantly enhanced my understanding of the essence of data mining.

REFERENCES

[1] B. Liang and M. Gu, "Music Genre Classification Using Transfer Learning," 2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), Shenzhen, Guangdong, China, 2020, pp. 392-393, doi: 10.1109/MIPR49039.2020.00085.

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

[3] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," in Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP), 1997, pp. 1331–1334.

[4] S. Davis and P. Mermelstein, "Experiments in syllable-based recognition of continuous speech," IEEE Trans. Acoust, Speech, Signal Processing, vol. 28, pp. 357–366, Aug. 1980.

[5] Auguin N, Huang S, Fung P. Identification of live or studio versions of a song via supervised learning[C]//Proc of Signal and Information Processing Association Annual Summit and Conference, 2013:1-4.

[6] Bergstra J, Casagrande N, Erhan D, et al. Aggregate features and AdaBoost for music classification[J]. Machine Learning, 2006, 65:473-484.

[7] Li T, Ogihara M, Li Q. A comparative study on content-based music genre classification[C]// Proc of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2003:282-289.

[8] Allamy S, Koerich A L. 1D CNN architectures for music genre classification[C]// Proc of 2021 IEEE Symposium Series on Computational Intelligence, 2021:1-7.

[9] Choi K, Fazekas G, Sandler M, etal. Convolutional recurrent neural networks for music classification[C]// Proc of 2017 IEEE International Conference on Acoustics, Speech and Signal Processing, 2017: 2392-2396.

[10] Sigtia S, Dixon S. Improved music feature learning with deep neural networks[C]// Proc of 2014 IEEE International Conference on Acoustics, Speech and Signal Processing, 2014: 6959-6963.

[11] Fulzele P, Singh R, Kaushik N, et al. A hybrid model for music genre classification using LSTM and SVM[C]//Proc of 2018 11th International Conference on Contemporary Computing, 2018:1-3.