MUSIC GENRE CLASSIFICATION

Suhas A Bhyratae
Asst. Professor
Dept. of CSE
VTU
Mangaluru, Karnataka
suhasabhyratae.cs@sahyadri.edu.in

Anantha Krishna G.K
Dept. of CSE
VTU
Mangaluru, Karnataka
ananthak.cs18@sahyadri.edu.in

Deepraj Majalikar
Dept. of CSE
VTU
Mangaluru, Karnataka
majalikar.cv18@sahyadri.edu.in

Advithiya A Bangera
Dept. of CSE
VTU
Mangaluru, Karnataka
advithiya.cs18@sahyadri.edu.in

Maruthi
Dept. of CSE
VTU
Mangaluru, Karnataka
maruthi.me18@sahyadri.edu.in

ABSTRACT

People find it harder and harder to control the songs they listen to as internet music databases and simple access to music information pro life rate. Identifying music genres from audio data and classifying them is an important task. In the realm of music information retrieval, the classification of musical genres is frequently employed. The three key processes covered by the proposed framework are feature extraction, classification, and data pre-processing. Using a K-nearest neighbor (k-NN), the categorization of musical genres is attempted. In order to categorize songs into their respective music genres, the proposed system feeds feature values from spectrograms created from slices of songsinto a CNN. Following the categorization process, a recommendation system is also put inplace. The goal of the recommendation system is to provide music based on the tastes andpreferences of each user. Extensive tests performed on the GTZAN dataset demonstrate the efficacy of the suggested strategy in comparison to alternative approaches.

Keywords— Feature Extraction Convolutional Neural Network Classification Recommendation System, K-nearest neighbor (k-NN), music, genre, classification, features, Mel Frequency Cepstral Coefficients (MFCC).

I. INTRODUCTION

Today, a person's personal music collection often consists of a few hundred tracks, whereasa professional collection typically consists of thousands or even millions of music files. In comparison to specialized archives and private sound collections, music databases are continuously improving their reputation. The number of users accessing the music database has increased along with advancements in internet services and increased network band- width. Extremely huge music libraries require a lot of time and effort to manage.

The majority of music files are organized by song title or artist name. This could make it difficult to find music that fits a particular genre. Classical, blues, disco, country, hip-hop, metal, jazz, reggae, rock and pop are among the most popular musical genres. In addition to the music, lyrics have also been used to categorize music into genres and subgenres. This makes identifying the type of music challenging. Additionally, the concept of a musical genre has evolved over time. Pop songs created fifty years ago, for example, differ from pop songs created today. Fortunately, during the past few years, there has beena significant advancement in music data and its storage.

There are primarily two processes in the classification of music genres: feature extraction and model construction. The process of removing different characteristics from audio recordings is known as feature extraction. Features taken from the music include spectral contrast, zero crossing rate, spectral bandwidth, spectral centroid, mel-frequency cepstral coefficients and spectral roll-off. The feature extraction module influences the classifier's functionality and design by removing the most pertinent information from the raw music data. Compared to extracting features from music signals, speech recognition and speech discrimination feature extraction need more work. With the help of the Librosa Python package, all of these capabilities are possible.

II. METHODOLOGY

A. Data Collection

The data set that is been used here is GTZAN which has assess to the categorization precision of MusicRecNet. There are 1000 songs on it (16 bits resolution, 30 s duration, and sampling frequency 22,500 Hz). Classical, blues, disco, country, hip-hop, metal, jazz, reggae, rock and pop are among the genres represented in the GTZAN. There are 100 different examples of each genre.

B. Data Pre-Processing

The split convert function calls both to mel-spectrograms and split songs functions. The split convert function is used to splitting the songs into small chunks. These small chunks are fed into the Librosa python library. The Librosa library is used to visualize the spectrograms. The Librosa library internally uses a Fast Fourier Transform (FFT) algorithm to analyze the audio signals in time domain and convert spectrograms in the frequencydomain. The two mel-spectrograms function is used to convert small chunks of songs into spectrograms and assign labels to small chunks. Before Feature extraction, feature scaling and hot encoding is performed. Feature scaling is the process of normalizing the data ranging between 0 and 1. The one-hot encoding is a method used to convert categoricaldata into integer data.

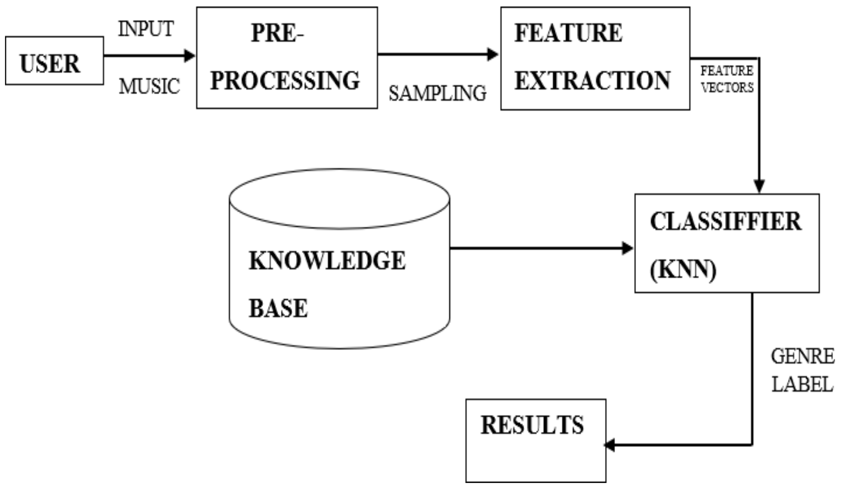

Figure 1: Proposed System

The above Figure 1 illustrates the proposed system architecture. The user inputs audio files, perform data preprocessing, feature extraction and model building. After classification, a recommendation system is implemented.

C. Feature Extraction

In feature extraction, extracting meaningful features from slices of spectrograms such as Mel-spectrograms, Spectral centroid, Spectral roll-off, Zero-crossing rate, Spectral band- width, and Chromo frequencies. Feature extraction is a valuable activity for analyzing and understanding the relations between songs and genres. Extracting the feature values from the slices of songs using Librosa python library. The Librosa python library performs feature extraction using different signal processing techniques. The mel-spectrograms function not to convert spectrograms into images and only using the spectrogram features.

These features are stored in a NumPy array (arr spec) with all the numerical values comprising the spectrogram. The arr genres are composed of categorical variables of genres up to 0 to 9. After feature extraction, data augmentation is performed. The function GTZAN generator is created using the sequence method from TensorFlow. It belongs to the data generator classes. The advantage of using data augmentation is to increase the amount of training data. It reduces overfitting and gradually increases the accuracy of the model.

D. K-Nearest Neighbors

The algorithm, K-NEAREST NEIGHBORS (KNN) forecasts the new data points value based on similarity. The values of the new data point are identified by finding the distancebetween data points and finding how closely they are related to one another in training data set. How to proceed:

1. Step one, the training and test data set must be created from the original data set.
2. The value of K must be chosen in random in order to find the data points that arenearest to test data.
3. Repeat the following steps for each values in test data:
    a. Find the distance for each value in training data set and the test data. The mostcommonly used method to compute the distance between 2 points is Euclidean.
    b. With the help of distance value sort them in ascending order.
    c. Choose the top K values in the sorted array.
    d. Based on the most frequent class, an class will be assigned to the test point forthose rows.
4. End

E. **Classification**

Compile the model by giving the loss function and optimizer. The optimizer is used as adam optimizer. These will be the parameters for the neural network training process. ReduceLRonPlateau is used to reduce the learning rate when a matric has stopped im- proving. Batch size is mentioned which takes into a number of input samples at each epoch of training. More batch size can increase the accuracy as well as training time. Train the model for 200 epochs. The training is done purely in CPU using TensorFlow. After training, the model is saved using the function 'model.save' function in Keras. For testing purposes, reloading the model. Then pass the test data into it and generate the model accuracy classification report and confusion matrix. Next stage, put the file wantto test in the 'unknown' folder.

Then reuse the read data function to read the unknown music. This will extract the features from the music and convert it into a format that is readable for the neural network. The unknown data is split into sub-samples. Each sub-samples is then passedinto the neural network to predict and give the output label. From the output labels, implement a majority voting system that checks which genre has the most count and then classifies the music belonging to that genre. There are 30 slices of unknown audio thatwill pass into the model and predict the output. All the 30 predicted labels are added toa list. Use a concept of dictionary element in python to find out which label is the most predicted from among the 10 labels.

F. **Recommendation System**

After classification of music genre, perform a recommendation system. Users may utilize the recommendation system to propose goods based on their interests. The recommendation of songs using features extracted from time slices of songs. Feature extraction was carried out by means of digital signal processing methods. In the recommendation engine, bring out the top 5 songs that are similar to the input songs. This is done using feature similarity and cosine similarity. The cosine similarity is used to measure the similarity be- tween two items. Using the method find all features when spanned in vector plane are lying close to one another. The test data folder is considered a database for the recommendation. The output will contain songs from different genres.

III. EXPERIMENTAL RESULT

Testing is a procedure of executing the program with unequivocal intension of discovering mistakes, assuming any, which makes the program, fall flat. This stage is an essentialpiece of the product improvement. It performs an incredibly fundamental role in quality assurance and in ensuring programming quality without fail. It is a method for identifying errors and missing operations, as well as a thorough confirmation to determine whether the goals and client requirements have been attained.

The testing objective is to reveal prerequisites, outline or coding blunders in the projects. Therefore, unique levels of testing are utilized in programming frameworks. The testing results are utilized amid upkeep. This area manages the points of interest in the various classes of the test which should be directed to approve capacities, imperatives and execution. This can be accomplished fundamentally by using the methods for testing, which assumes a crucial part in the improvement of a product.

A. **Unit Testing**

Individual software modules or components are tested as part of the software testing processknown as unit testing. The goal is to confirm that each piece of software operates as intended. Test cases are built in order to test and make sure that all the components within the system interact properly. The goal is to detect the error in each module. The various test cases for the modules of the project are listed below. Testing modules are described below:

- Feature extraction.
- Classifier.

Sl No. Test Case	1	**Duration(in Weeks)**
Name of the Test	Unit Testing for Feature Extraction	1
Sample Input	.wav Audio file	1
Expected output	Features in the dimension of 1x104 matrix	2
Actual output	Same as expected output & Porting	2
Remark	Successful	2

Table 1: Unit testing for feature extraction

Table 1 illustrate the testing of the feature extraction of input .wav audio file. The features are extracted from the .wav audio file. The test has been conducted for successful case.

Sl No. Test Case	2	**Duration(in Weeks)**
Name of the Test	Unit testing for classifier	1
Sample Input	Extracted feature vector	1
Expected output	Label the genre	2
Actual output	Same as expected output & Porting	2
Remark	Successful	2

Table 2: Unit testing for classification

B. **Experimentation**

We recorded 1,000 audio tracks, each lasting 30 seconds. Each of the 10 featured genres has 100 tracks. The ten genres chosen from those represented in the GTZAN are classical, blues, disco, country, hip-hop, metal, jazz, reggae, rock, and pop. A total of 1000 songs made up our data set, of which 70% were used for training, 30% for testing and outcomesmeasurement, and additional songs were randomly selected for random cross validation.

IV. RESULT AND DISCUSSION

The proposed system architecture is implemented using Jupyter Notebook. Most of the experiments are being carried out on Jupyter Notebook and it is helping to write and run python code through the browser. Compile the model by giving all loss functions and an optimizer to use. We work through this project on GTZAN music genre classification data-set. It explains how to extract important features from audio files. In this classification system, we used K-Nearest Neighbor (K-NN) and Support Vector Machine (SVM) which is developed in Convolutional Kernal with the help of Convolutional Neural Network (CNN) that provide more accuracy compared to the previous system. The testing data-set gives an accuracy of more than 62% (see in Table 3).

Experiment Results for classification in KNN:

Conventional k-nearest neighbors	**Modified k-nearest neighbors**
CORRECT PREDICTIONS: **214**	CORRECT PREDICTIONS: **270**
TOTAL PREDICTIONS: **345**	TOTAL PREDICTIONS: **328**
ACCURACY: **0.6202898550**	ACCURACY: **0.8231707317**

Table 3: Experiment Results for classification in KNN

V. CONCLUSION

Music genre categorization uses k-NN to categories musical styles from audio recordings. In this study, the majority voting technique is utilized to classify an unknown audio clip into a musical genre while also utilizing the spectrogram feature values from time-slices of songs. The experimental findings demonstrate that the suggested system outperforms previous approaches on the GTZAN dataset. Some songs, we can argue, include elementsof several different genres. So, based on the probability, we have also attempted to obtainmultiple label outputs. Additionally, we looked for the genre class combinations that would produce the highest accuracy.

The future work is to improve the performance of the system and also We can deploythis as a Web App on the cloud. We can also provide API for developers. Another thingwe can try to improve the accuracy is trying a different data set or dynamically increasethe data set as the user input new music.

REFERENCES

[1] K. West and S. Cox, “Features and classifiers for the automatic classification of musicalaudio signals.,” in *ISMIR*, 2004.

[2] H. Bahuleyan, “Music genre classification using machine learning techniques,” *arXiv preprint arXiv:1804.01149*, 2018.

[3] G. Tzanetakis and P. Cook, “Musical genre classification of audio signals,” *IEEE Transactions on speech and audio processing*, vol. 10, no. 5, pp. 293–302, 2002.

[4] A. Tsaptsinos, “Music genre classification by lyrics using a hierarchical attention network,” ICME, 2017.

[5] Y. Xu and W. Zhou, “A deep music genres classification model based on cnn with squeeze & excitation block,” in *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 332–338, IEEE, 2020.

[6] R. Yang, L. Feng, H. Wang, J. Yao, and S. Luo, “Parallel recurrent convolutional neural networks-based music genre classification method for mobile devices,” *IEEE Access*, vol. 8, pp. 19629–19637, 2020.

[7] D. Kostrzewa, P. Kaminski, and R. Brzeski, “Music genre classification: Looking forthe perfect network,” in *International Conference on Computational Science*, pp. 55–67, Springer, 2021.

[8] R. Mayer, R. Neumayer, and A. Rauber, “Combination of audio and lyrics features for genre classification in digital audio collections,” in *Proceedings of the 16th ACM international conference*

on Multimedia, pp. 159–168, 2008.

[9] T. Feng, “Deep learning for music genre classification,” *private document*, 2014.