Research on Music Genre Classification and API Development to Detect Genre of a Song

Aman Prakash
University at Buffalo (Graduate Student)
Buffalo, United States
amanprak@buffalo.edu

Prashant Upadhyay
University at Buffalo (Graduate Student)
Buffalo, United States
pupadhya@buffalo.edu

Serath Chandra Nutakki
University at Buffalo (Graduate Student)
Buffalo, United States
serathch@buffalo.edu

Vamshivardhan Reddy Balannagari
University at Buffalo (Graduate Student)
Buffalo, Unites States
vamshiva@buffalo.edu

***Motivation* — Music is a comprehensive language and symbol that we design, understand, and adore with other people. They are numerous forms which contain universal features of music called as genres regardless of their rhythm, form, scoring or timbre. We postulate that we can determine the genre musical category of a song using these features. Our project main objective is to build a music genre classifier using machine learning approach that will attempt to predict the genre and the confidence level of the music from different candidate genres.**

I. INTRODUCTION

Nowadays with the expeditious advancement of the internet alongside with the widening of bandwidth accessibility made the ingredients of digital multimedia available in copious amount. Even during the preliminary stages of the Internet, the research of Music genre classification has been extensive. MP3 format is the paramount types of multimedia content distributed over the Internet. Which directed many researchers to establish music information retrieval (MIR) techniques. To discover music from diverse options that would be beneficial for Internet based music search engines, musicologist, and listeners.

In this study, we categorize music with different gerne and correlate the performance using different models. It is a machine learning approach where a model is trained end-to-end and then advance to predict the genre label of an audio signal, solely using its spectrogram. From the above derived models, we develop a flask-based webapp to detect the genre of a songs and store history of the searches and other data in a SQLlite database, from where the data can be retrieved in the Webapp.

II. DATA DESCRIPTION

A. *Dataset*

The dataset for the project was obtained from a well-known paper in genre classification "Musical genre classification of audio signals " by G. Tzanetakis and P. Cook in IEEE Transactions on Audio and Speech Processing 2002. 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. The process carried from EDA to model building was done using the training set.

B. *Pre-processing the Data*

Prior to training the classification model. The source data from audio samples should be converted into more significant representation by using special audio libraries in python for audio processing like Librosa and PyAudio to read the data.

C. Taking care of missing values if any

There are two ways to handle the missing values, either we can drop it or we can replace the same with the mean value of the cell. For our simplicity we removed the missing values from the table.

III. PROPOSED APPROACH

A. *Feature Selection*

To build a classification model it is not feasible to use all the features, so we need to extract meaningful features from audio files. The features that we used are Mel-Frequency Cepstral Coefficients, Spectral Centroid, Zero Crossing Rate, Chroma Frequencies, Spectral Roll-off.

B. *Model Analysis*

By using the extracted features are then converted into a csv file and then we apply different classification models to train the data.

Machine Learning Techniques:

- Importing Scikit-learn
- Organizing Data into Sets
- Building the Model using various algorithms
- Evaluating the models by different metrics

Classifiers used: Decision Tree, Logistic Regression, Random Forest, SVM

Model Evaluation: Compare the result from multiple architectures with different models.

C. *FLASK WebAPP*

WebAPP to interact with dataset and automate the process of genre identification for client using any approach from the models trained above.

D. *CRUD operations with Database*

DB CRUD operations to support various functionalities of WebAPP like search history, etc.

IV. RESULTS

A. *Research:*

The model is being trained by two different approaches one is of classical approach and another is on deep learning approach. We have trained the model in 5 different classical approaches which are Logistic Regression, Elastic Net, Decision Tree, Random Forest and Support Vector Machine and we have seen that all approach giving an average accuracy of 65% on test set and 75% on train set where're as the deep learning approach is giving an average accuracy of 82%. We have also developed flask API to determine the music genre using CLI command and used postman to develop the flask API. We tried to keep the search history of the user and for that we used the database.

B. *Application (Flask):*

Developed Flask API Server to implement the models generated from the above research into real world use by developing the two API's below. We have also made use of SQLite Database to demonstrate the implementation of Flask API development, Machine Learning Model and Database CRUD operations.

API Definitions –

1) API 1 –

Purpose-

The purpose of the API is to identify genre of a song where a user just gives the file path as input in the body of the request and the API will return the genre of the song as the output shown below. Also search history is saved in the database in the background so that user can fetch his/her search history to enhance their experience.

Curl Request-

```
curl --location --
request POST 'http://127.0.0.1:5000/api/search
' \
--header 'Content-Type: application/json' \
--data-raw '{
    "path": "C:\\\\Users\\\\ASUS\\\\Desktop\\\
\eas503_project\\\\data\\\\samples\\\\disco.wa
v"
}'
```

Output –

```
{
    "result": "disco"
}
```

2) API 2 -

Purpose-

This API is used to fetch the search history of all the music genre identification process performed by the user. This helps the user to fetch his search history without opening the database. A simple SELECT query is performed in the background and we can see the output given below.

Curl Request-

```
curl --location --
request GET 'http://127.0.0.1:5000/api
/search' \
--data-raw ''
```

Output –


```
[
    {
        "file_name": "C:\\\\Users
\\\\ASUS\\\\Desktop\\\\
eas503_project\\\\data\\\\
samples\\\\classical.wav",
        "result": "classical",
        "search_id": 1
    },
    {
        "file_name": "C:\\\\Users
\\\\ASUS\\\\Desktop\\\\
eas503_project\\\\data\\\\
samples\\\\disco.wav",
        "result": "disco",
        "search_id": 2
    },
    {
        "file_name": "C:\\\\Users
\\\\ASUS\\\\Desktop\\\\
eas503_project\\\\data\\\\
samples\\\\disco.wav",
        "result": "disco",
        "search_id": 3
    }
]
```


V. FUTURE ENHANCEMENTS

We are trying to develop a website which will take the input as a music file and try to find related music with the same genre type on real time.

VI. REFERENCES

[1] " Musical genre classification of audio signals " by G. Tzanetakis and P. Cook in IEEE Transactions on Audio and Speech Processing 2002. http://marsyas.info/downloads/datasets.html