JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Music Genre Classification Using Deep Learning

D. Bharadwaja[1], T. Indu Priya[2], N. Vishnu Priya[3], M. Srirama Chandra[4], T. Lohith[5]
Assistant Professor, Department of Information Technology[1]
B. Tech Students, Department of Information Technology [2,3,4,5]
Prasad V. Potluri Siddhartha Institute of Technology, Vijayawada, Andhra Pradesh, India

Abstract: *A music genre is a term used to group distinct musical styles together based on a custom or set of guidelines. It needs to be separated from musical form and style. There are many different ways to divide music into different genres. Pop, Hip-Hop, Rock, Jazz, Blues, Country, and Metal are the most popular musical genres. In the field of music information retrieval (MIR), classifying music files by genre is a difficult task. To retrieve music from a vast collection, automatic music genre classification is crucial. It has real-world applications in a variety of domains, such as automatic tagging of unknown piece of music (useful for apps like Gaana, Spotify, Saavn, Wynk etc.). This project aims to predict the genre using KNN algorithm with an audio signal as its input. Making the choosing of songs quick and less laborious is the goal of automating the music classification. If one has to manually classify the songs or music, one has to listen to a whole lot of songs and then select the genre. We are doing the experiment using the GTZAN data set, a popular public data set for music recognition research. (MGR).*

Keywords: *Genres, Audio files, MFCC, KNN Algorithm*

1. INTRODUCTION

A piece of music's genre can be predicted by a piece of software that analyses audio files. These gadgets are utilised for things including selecting the right background music for occasions and automatically tagging songs for distributors like Spotify Compared to image processing and other classification methods, audio processing is one of the most challenging data science projects. The categorization of musical genres is one such application, which aims to classify audio files into the relevant sound categories to which they belong. The application is essential and requires automation to reduce manual error and time because manually categorising music necessitates listening to each track for the complete. As a result, we automate the process using machine learning and deep learning approaches, which we'll use in this article. Genre classification is now done manually by people using their individual musical knowledge. Since the differences between music genres are rather subtle, this task has not yet been mechanised by typical computational techniques. If the songs or music must be manually categorized, one must first listen to a huge number of tracks before choosing the genre. Machine intelligence is ideally suited to this task because genre classification is ambiguous. Machine learning can notice and forecast using these ill-defined patterns given enough audio data, of which vast volumes can be obtained with ease from freely available music online. Traditionally, music genres are defined based on a combination of musical elements such as rhythm, melody, harmony, timbre, instrumentation, and lyrics. Some common genres include rock, pop, jazz, classical, hip-hop, country, and electronic music. However, there are many sub-genres within each of these broad categories, and the classification of a particular song or piece of music can be subjective and challenging.

2. LITERATURE SURVEY

There are numerous studies that have explored the use of deep learning for music genre classification, and here are some recent ones:

- Lee, S., Lee, Y., & Kim, Y. (2019). Music genre classification using K-means clustering and convolutional neural networks. IEEE Access, 7, pp. 108182-108193. In this paper, the authors propose a music genre classification system that combines K-means clustering and convolutional neural networks (CNN). The features used in this study include Mel-spectrogram features, chroma features, and tonnetz features. The features are clustered using K-means clustering, and the resulting clusters are used to initialize the weights of the CNN layers.

- "Music Genre Classification using K-Nearest Neighbors (KNN) Algorithm" by Muhammad Fajri Kurniawan and Yanuar Nugroho. This paper proposes a music genre classification method using KNN with various audio features, including tempo, spectral centroid, and mel-frequency cepstral coefficients (MFCCs). The results show that KNN can achieve high accuracy in genre classification tasks.

- "Music Genre Classification Using KNN with Spectral Features" by S. M. Ashraful Islam, Md. Rashedul Islam, and Md. Ariful Islam. This paper presents a music genre classification approach using KNN with spectral features extracted from audio signals. The study shows that KNN performs well in terms of accuracy and computational complexity.

3. PROPOSED SYSTEM

The K-Nearest Neighbors (KNN) algorithm is used in the suggested method to classify the musical genres. KNN is a machine learning technique that is used for classification and regression. The lazy learner algorithm is another name for it. The output is the class where the majority of the neighbours are located after using a distance-based algorithm to discover the K number of similar neighbours to the new data. The K-Nearest Neighbors technique is one of the greatest algorithms for providing good performance, and organisations continue to utilise it as support in recommendation systems along with optimised models. For sound samples, the Mel Frequency Cepstral Coefficient (MFCC) is employed as a feature vector. The suggested technique uses feature vector extraction to categorise music into different genres. The KNN algorithm has a natural advantage to its accuracy over K-Means algorithm by virtue of possessing training data in advance. KNN can create non-linear decision boundaries which can which can be useful in complex classification tasks. KNN can be robust to noisy data because it takes into account the neighboring samples rather than relying on a single sample to make a prediction.

4. RESULTS

4.1 Installing the python package

```
pip install python_speech_features

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting python_speech_features
  Downloading python_speech_features-0.6.tar.gz (5.6 kB)
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: python_speech_features
  Building wheel for python_speech_features (setup.py) ... done
  Created wheel for python_speech_features: filename=python_speech_features-0.6-py3-none-any.whl size=5886 sha256=68ad2788524b7798ea90feaf43211b1ae2ec54b7d8f961256f98de237de9591c
  Stored in directory: /root/.cache/pip/wheels/09/a1/04/08e2688d2562d8f9ff89e77c6ddfbf7268e07dae1a6f22455e
Successfully built python_speech_features
Installing collected packages: python_speech_features
Successfully installed python_speech_features-0.6
```

Decription: The package "python_speech_features" provides a set of common speech feature extraction algorithms, including MFCC (Mel-Frequency Cepstral Coefficients), filterbank energies, and others. These features can be used for speech recognition, speaker identification, and other speech-related applications.

4.2 Importing the features

```python
from python_speech_features import mfcc
import scipy.io.wavfile as wav
import numpy as np

from tempfile import TemporaryFile
import os
import pickle
import random
import operator
import math
```

Decription: The first line imports the "mfcc" function from the "python_speech_features" package. The MFCC algorithm is a common technique used for speech feature extraction. The second line imports the "wavfile" function from the "scipy.io" package. This function is used to read WAV files, which are commonly used for storing audio data. The third line imports the "numpy" package, which provides a set of tools for working with arrays and numerical data in Python. The fourth line imports the "matplotlib.pyplot" module, which is a plotting library for Python. This library is commonly used to visualize data, including speech data. The fifth line imports the "TemporaryFile" class from the "tempfile" module. This class is used to create temporary files, which can be used to store data during processing. The sixth line imports the "os" module, which provides a set of functions for interacting with the operating system. This module is commonly used to work with files and directories. The seventh line imports the "pickle" module, which provides a way to serialize and deserialize Python objects. This module is commonly used to save and load trained machine learning models.

4.3 Overview of the dataset

```
          filename  length  chroma_stft_mean  chroma_stft_var  rms_mean  \
0  blues.00000.0.wav   66149          0.335406         0.091048  0.130405
1  blues.00000.1.wav   66149          0.343065         0.086147  0.112699
2  blues.00000.2.wav   66149          0.346815         0.092243  0.132003
3  blues.00000.3.wav   66149          0.363639         0.086856  0.132565
4  blues.00000.4.wav   66149          0.335579         0.088129  0.143289

    rms_var  spectral_centroid_mean  spectral_centroid_var  \
0  0.003521             1773.065032          167541.630869
1  0.001450             1816.693777           90525.690866
2  0.004620             1788.539719          111407.437613
3  0.002448             1655.289045          111952.284517
4  0.001701             1630.656199           79667.267654

   spectral_bandwidth_mean  spectral_bandwidth_var  ...  mfcc16_var  \
0              1972.744388           117335.771563  ...   39.687145
1              2010.051501            65671.875673  ...   64.748276
2              2084.565132            75124.921716  ...   67.336563
3              1960.039988            82913.639269  ...   47.739452
4              1948.503884            60204.020268  ...   30.336359

   mfcc17_mean  mfcc17_var  mfcc18_mean  mfcc18_var  mfcc19_mean  mfcc19_var  \
0    -3.241280   36.488243     0.722209   38.099152    -5.050335   33.618073
1    -6.055294   40.677654     0.159015   51.264091    -2.837699   97.030830
2    -1.768610   28.348579     2.378768   45.717648    -1.938424   53.050835
3    -3.841155   28.337118     1.218588   34.770935    -3.580352   50.836224
4     0.664582   45.880913     1.689446   51.363583    -3.392489   26.738789

   mfcc20_mean  mfcc20_var  label
0    -0.243027   43.771767  blues
1     5.784063   59.943081  blues
2     2.517375   33.105122  blues
3     3.630866   32.023678  blues
4     0.536961   29.146694  blues
```

Decription: The variable "data" reads a CSV file located at "/content/drive/My Drive/music_genre_project/features_3_sec.csv". The CSV file is assumed to have a header row and be delimited by commas. The "read_csv" function is provided by Pandas and is used to read data from a CSV file into a DataFrame, which is a two-dimensional table of data with rows and columns. Next, it prints the first few rows of the data using the "head" method. This method is provided by Pandas and is used to retrieve the first n rows of a DataFrame. By default, n is 5. The purpose of this line is to quickly inspect the data and make sure that it was read correctly.

4.4 Accuracy of the predictions

```
accuracy1 = getAccuracy(testSet , predictions)
print(accuracy1)
```

```
0.6893203883495146
```

Decription: The first line calls the "getAccuracy" function with two arguments: the test set and a list of predicted class labels. This function compares the predicted class labels to the true class labels in the test set and returns the accuracy of the classifier as a percentage. The result is stored in a variable called "accuracy1". The second line prints the accuracy of the classifier on the test set, which was stored in the "accuracy1" variable. The accuracy is printed as a floating-point number between 0 and 100, representing the percentage of test instances that were classified correctly.

4.5 Genre of unknown audio file

```
path = "/content/drive/My Drive/music_genre_project/genres_original/"
path_to_new_audio_file="/content/drive/My Drive/music_genre_project/sample_genre/sample/sample_audio.wav"
```

```
print(results[pred])
```

```
classical
```

Decription: The first variable, "path", stores a file path to a directory that contains audio files. The directory is located at "/content/drive/My Drive/music_genre_project/genres_original/". The purpose of this variable is likely to be used for reading and processing audio files in that directory. The second variable, "path_to_new_audio_file", stores a file path to a new audio file. The file is located at " /content /drive /My Drive/ music_genre_project /sample_genre /sample /sample_audio.wav". The purpose of this variable is likely to be used for reading or writing the new audio file, depending on the context of the code.

5. SCOPE OF FUTURE USE

Music streaming services such as Spotify, Apple Music, and Pandora could use music genre classification algorithms to improve their recommendation systems. Music discovery platforms such as Shazam, SoundHound, and Musixmatch could use genre classification to help users identify songs they like. Music supervisors for film, TV, and advertising could use genre classification to quickly identify music that fits the mood or style of a particular project. This would save time and effort in the music selection process. Music teachers could use genre classification to help students learn about different types of music. By introducing students to a range of genres, teachers could help them develop a deeper understanding of music and a more diverse taste in music.

6. CONCLUSION

Music genre classification plays a significant role in the music industry and beyond. With the help of machine learning and artificial intelligence, music genre classification algorithms are becoming increasingly accurate and efficient, allowing for a wide range of applications in areas such as music streaming, music discovery, music education, music research, and music supervision. By using music genre classification, we can better understand and appreciate the diversity of music and its cultural significance. We may anticipate much more fascinating breakthroughs in the realm of identifying musical genres as technology develops.

REFERENCES

[1]. Rajeeva Shreedhara Bhat,Rohit B. R., Mamatha K. R, "Music Genre Classification" SSRG International Journal of Communication and Media Science 7.1 (2020): 8-13

[2]. Robert Adragna, Yuan Hong (Bill) Sun "Music Genre Classification"

[3]. Athulya K M, Sindhu S,"Deep Learning Based Music Genre Classification Using Spectrogram", published on International Conference ICICNIS 2021 on 14 Jul 2021

[4]. Sam Clark, Danny Park, Adrien Guerard, "Music Genre Classification using Machine Learning Techniques", 2012.

[5]. M.D.Nevetha, A.Nithyasree, A.Parveenbanu,Mrs.Jetlin CP,"Music Genre Classification Using Machine Learning",published on 2013

[6]. T. Feng, "Deep learning for music genre classification", 2014.