Music Genre Classification

by Rushika B Prasad

Submission date: 19-Aug-2021 12:56PM (UTC+0530)
Submission ID: 1633137777
File name: Music_Genre_Classification_Report.pdf (616.44K)
Word count: 1728
Character count: 8833

Music Genre Classification

Rushika B Prasad
Computer Science Engineering
PES University
Bengaluru, India
rushikabp@gmail.com

Prof. K S Srinivas
Computer Science Engineering
PES University
Bengaluru, India
srinivasks@gmail.com

I. Introduction

Music is one of the universal arts and aspects of human activities. It is basically the art of arranging together the sounds in time to produce a melodious tune or composition with certain elements such as the timbre, melody, rhythm and harmony. Music has various types of genres such as classical, rock, blues, hip-hop, jazz etc.,

Today's world is completely dependent on the internet and most of the things in our daily lives are directly or indirectly related to machine learning. Machine Learning is one of the trending topics at present using which many problems are solved efficiently. Coming to the world of music, nowadays listening to music on various platforms is very common. Listening to music has become very simple these days on platforms such as Wynk Music, Amazon Music, Spotify and many more which run with internet access. If a person wants to listen to a particular genre of music/song or a person wants to find out the genre of a particular song, then a music genre classifier would do the needful. Inorder to classify the music into its respective genre will need a music genre classifier. Music Genre Classifiers can be built with different Machine Learning techniques such ANN, CNN, SVM or many other techniques.

In this paper, we have considered one of the efficient algorithms in Machine Learning, that is Convolutional Neural Network (CNN). Convolutional Neural Network is a classification algorithm which is mainly used to classify or analyse the visual or image dataset. To obtain the images from the music sample we have made use of feature extraction. Using Feature Extraction, we can obtain the necessary features of the music in image form and this can be fed as an input to the CNN model for the classification of the music to its respective genre. Music Genre Classification was achieved by using a CNN model which has given 81% accuracy.

II. Literature Review

Music Genre Classification is a very obliging and a user-friendly application. As we know, Machine learning has numerous algorithms to achieve an efficient classifier. Variety of techniques have been used to achieve a good classifier till date. For example, Support Vector Machine (SVM), Naive Bayes, K-Nearest Neighbour (KNN), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN) and many more.

A. *Support Vector Machine (SVM) and LSTM:*

.SVM is one of the supervised machine learning techniques which is used for both regression and classification type of problems. SVM basically draws separation boundaries to classify multiple classes. These separation boundaries are nothing but hyper-planes. Long Short Term Memory (LSTM) is another machine learning algorithm and an artificial recurrent neural network. First the SVM model and LSTM model are trained with the dataset individually. Later, the two models are combined to obtain the final prediction or the result. Individual training of the dataset on the SVM model resulted in 84% accuracy and training with the LSTM model the accuracy was around 69%. Now, the combined model (SVM and LSTM) resulted in 89% accuracy.

B. *Recurrent Neural Network (RNN):*

RNN is a class of artificial neural networks in deep learning algorithm in which the connection between the directed graphs along a temporal sequence. RNN has many different algorithms in it such as Long Short Term Memory (LSTM), GRUs and NTMs. First the feature extraction is performed by using scattering transform method to extract the features from the given music sample dataset. Next, the dataset is trained with a 5-layer IndRNN which solves the gradient vanishing problem. This output will be given as an input to the softmax classifier for the final prediction. Using this machine learning technique the accuracy achieved is 95%.

C. *Convolutional Neural Network (CNN):*

CNN is one of the most famous and extensively used deep learning algorithms in machine learning for image classification. When compared to the other machine learning techniques implementation CNN has a very low accuracy rate. Most of the papers who have implemented CNN methods have not obtained a good accuracy. Initially, the music sample is given to the feature extractor to extract the features and this output is fed as an input to the CNN classifier for the classification. The accuracy that has been achieved is 47%.

III. Implementation

Music Genre Classification using the machine techniques has several steps involved. The dataset is pre-processed initially and then feature extraction is done. In order to build a classifier we have used a well known machine technique that is Convolutional Neural Network (CNN). The extracted features are fed to the CNN model as the input to classify the music to its respective genre.

A. *Dataset*

For a machine learning project dataset is the most important aspect as we know. The dataset considered to classify the music is GTZAN Dataset which is taken from the kaggle website. The dataset consists of 10 different music genres such as Blues, Classical, Country, Disco, Hip-hop, Jazz, Rock, Metal, Reggae and Pop as seen in the figure 1(a) below. Each genre has 100 music files and each music file is of 30 seconds duration as observed in the figure 1(b) below.

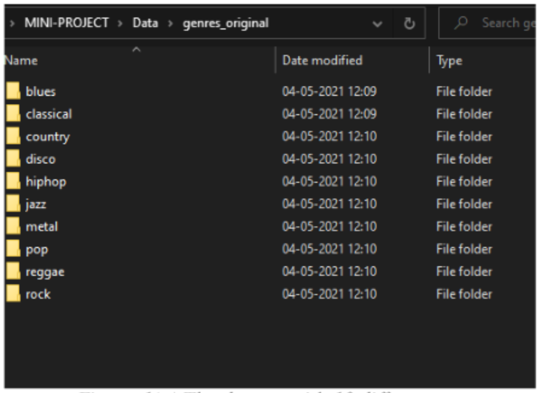

Figure 1(a) The dataset with 10 different genres

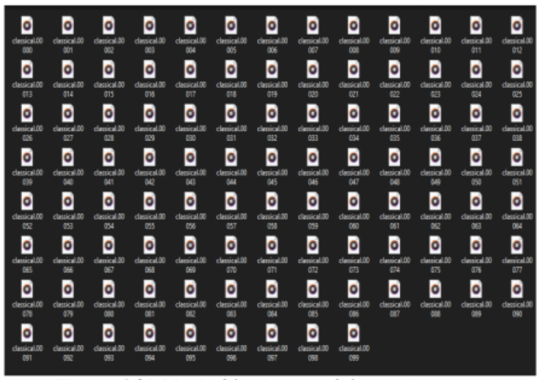

Figure 1(b) Music files in one of the music genres

B. Feature Extraction

Feature Extraction is one of the most important stages in music genre classification. The data which is provided in the dataset cannot be directly fed as an input to the CNN model for analyzing and classifying as it is not understood by the model. Therefore we need to convert the data into model understandable form. Inorder to do this, we need to extract the features from the music files to classify them into respective genres. Feature extraction is very important for classification, analysis and recommendation of music. Feature extraction is done using Mel Frequency Cepstral Coefficient (MFCC) as shown in the figure 2(a) which is the flowchart for feature extraction.

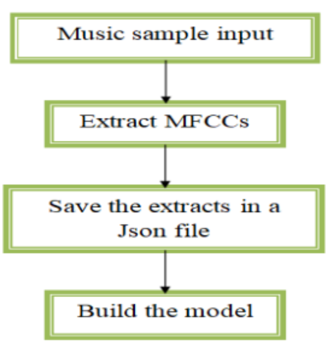

Figure 2(a) Process of feature extraction

Music data is audio data which is a 3-dimensional signal which consists of frequency, time and amplitude. The feature extraction of the music is performed with python packages such as librosa and IPython. A spectrogram is a visual representation of frequencies of a music signal changing with respect to time for a particular music signal. The data is converted to STFT i.e Short term Fourier Transform as shown in the figure 2(b). Mel-Spectrogram which is also known as Mel-frequency cepstrum (MFC) is a visual representation of the short-term power spectrum of the music signal as shown in the figure 2(c).

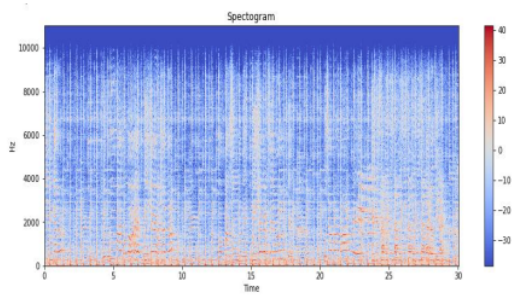

Figure 2(b): Spectrogram of classical music genre

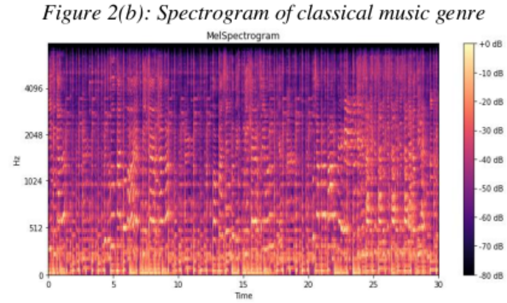

Figure 2(c): Mel-spectrogram of classical music genre

We have applied MFCC for feature extraction of features from the music signal. MFCC is basically a small set of features (around 10-20) which succinctly describes the overall shape of the spectral envelope. The shape of MFCCs is printed to know the number of MFCCs calculated on the number of frames. Therefore, we can conclude that MFCC is one of the efficient methods for feature extraction of a signal. MFCC is applied using the python package called librosa which extracts the features from the music data and stores the extracted features in a json file. The json file has mappings, music genre labels and the mfcc (extracted features).

C. Classification Model

Classification is a predominant stage in music genre recognition or classification. The features extracted from the feature extraction stage are fed as an input to the classification layer. The classification model is performed by one of the supreme models of deep learning, the Convolutional Neural Network (CNN) algorithm. The CNN model has several layers included such as input layer, convolutional layer, max-pooling layer and fully connected layer. We have basically considered a 5 layer neural network. The flowchart structure of the complete music genre process is as shown in the figure 3(a).

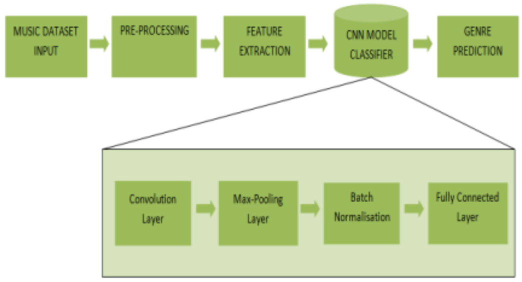

Figure 3(a) Structure of the Music Genre Classification

The model is trained with the training dataset to classify the music into its respective genre. As mentioned earlier the CNN model has 5 convolutional layers. Each convolutional layer has Conv2D, max-pooling2D, dropout of 0.3 and batch normalisation. The dataset passes through the 5 convolutional layers and reaches the fully connected layer and the output layer. In the fully connected layer we have a flatten, dense layer with relu as the activation function and dropout with 0.3. Then in the output layer we have a softmax activation function. The complete CNN model is built in Python Programming Language using Tensorflow and Keras.

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

import os, json, math, librosa

import IPython.display as ipd
import librosa.display

import tensorflow as tf
import tensorflow.keras as keras

from tensorflow.keras import Sequential
from tensorflow.keras.layers import Conv2D

import sklearn.model_selection as sk

from sklearn.model_selection import train_test_split
```

Figure 3(b): Python packages used to implement the CNN model

The CNN model also has an optimizer without which classification of music genre is impossible. An Optimizer is a mathematical function which is chiefly dependent on the model's learnable parameters. The optimizer used in the model is Adam with a learning rate of 0.001. The Adam optimizer is an in-built optimizer in keras. Tensorflow Keras is used to implement the optimizer.

```
# create network
input_shape = (X_train.shape[1], X_train.shape[2], 1)
model = build_model(input_shape)

# compile model
optimiser = keras.optimizers.Adam(learning_rate=0.001)
model.compile(optimizer=optimiser,
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.summary()
```

Figure 3(c): Optimizer used in the model

IV. Result and Conclusion

In this paper, we have used the Mel-Frequency Cepstral Coefficient method for feature extraction. The Convolutional Neural Network algorithm is used for classification and prediction of the music genre. The dataset was run for 500 epochs and with a batch size = 64 to achieve the aspired results as shown in the figures (4a, 4b, 4c) below. The CNN classifier has successfully resulted in 81.30% test accuracy. The primary aim of the project is classification and prediction of a given particular music genre was successfully attained by the CNN model. In figure 4(c), we can see that the actual music genre that is given to the CNN classifier is country and the genre predicted is country. Hence, we can conclude that the CNN classifier is predicting the music genre of the given music precisely.

The project can be further extended to a music recommendation system which would be very helpful to people to recommend the music genre of their choice.

```
[ ] history = model.fit(X_train, y_train, validation_data=(X_validation, y_validation), batch_size=64, epochs=500)

    # plot accuracy/error for training and validation
    plot_history(history)

    Epoch 1/500
    50/50 [==============================] - 11s 211ms/step - loss: 0.9433 - accuracy: 0.6601 - val_loss: 0.9056 - val_accuracy: 0.6901
    Epoch 2/500
    50/50 [==============================] - 11s 211ms/step - loss: 0.9558 - accuracy: 0.6566 - val_loss: 0.9367 - val_accuracy: 0.6635
    Epoch 3/500
    50/50 [==============================] - 11s 211ms/step - loss: 0.9458 - accuracy: 0.6652 - val_loss: 0.9489 - val_accuracy: 0.6872
    Epoch 4/500
    50/50 [==============================] - 11s 213ms/step - loss: 0.9691 - accuracy: 0.6544 - val_loss: 1.0601 - val_accuracy: 0.6449
    Epoch 5/500
    50/50 [==============================] - 11s 213ms/step - loss: 0.9483 - accuracy: 0.6544 - val_loss: 0.9431 - val_accuracy: 0.6857
    Epoch 6/500
    50/50 [==============================] - 11s 213ms/step - loss: 0.9110 - accuracy: 0.6766 - val_loss: 0.9279 - val_accuracy: 0.6827
    Epoch 7/500
    50/50 [==============================] - 11s 212ms/step - loss: 0.9376 - accuracy: 0.6668 - val_loss: 0.9770 - val_accuracy: 0.6620
    Epoch 8/500
    50/50 [==============================] - 11s 213ms/step - loss: 0.9319 - accuracy: 0.6636 - val_loss: 1.1212 - val_accuracy: 0.6079
    Epoch 9/500
    50/50 [==============================] - 11s 214ms/step - loss: 0.9072 - accuracy: 0.6747 - val_loss: 0.9620 - val_accuracy: 0.6664
    Epoch 10/500
    50/50 [==============================] - 11s 213ms/step - loss: 0.9238 - accuracy: 0.6839 - val_loss: 0.9700 - val_accuracy: 0.6686
    Epoch 11/500
    50/50 [==============================] - 11s 213ms/step - loss: 0.9186 - accuracy: 0.6677 - val_loss: 1.0215 - val_accuracy: 0.6553
    Epoch 12/500
    50/50 [==============================] - 11s 213ms/step - loss: 0.9161 - accuracy: 0.6712 - val_loss: 0.9452 - val_accuracy: 0.6783
    Epoch 13/500
    50/50 [==============================] - 11s 212ms/step - loss: 0.9109 - accuracy: 0.6652 - val_loss: 0.8309 - val_accuracy: 0.7235
```

Figure 4(a) Training of the model

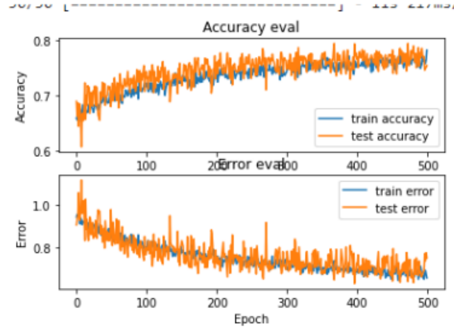

Figure 4(b): Accuracy and error graphs

```
[18] # evaluate model on test set
     test_loss, test_acc = model.evaluate(X_test, y_test, verbose=2)
     print('\nTest accuracy:', test_acc)

     47/47 - 1s - loss: 0.5623 - accuracy: 0.8131

     Test accuracy: 0.8130841255187988

     # pick a sample to predict from the test set
     X_to_predict = X_test[100]
     y_to_predict = y_test[100]

     # predict sample
     predict(model, X_to_predict, y_to_predict)

     Target: country, Predicted label: ['country']

[ ]  model.save("/content/drive/MyDrive/MINI-PROJECT/music1.h5")

[17] from tensorflow import keras
     model = keras.models.load_model('/content/drive/MyDrive/MINI-PROJECT/music1.h5')
```

Figure 4(c): Prediction of the music genre

Music Genre Classification

ORIGINALITY REPORT

4%	1%	2%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Indian Institute of Technology Patna Student Paper	1%
2	Submitted to Syracuse University Student Paper	1%
3	link.springer.com Internet Source	1%
4	danieltakeshi.github.io Internet Source	1%
5	Megha Banerjee, Sudhan Majhi. "Multi-class Heart Sounds Classification Using 2D-Convolutional Neural Network", 2020 5th International Conference on Computing, Communication and Security (ICCCS), 2020 Publication	<1%

Exclude quotes	On	Exclude matches	< 5 words
Exclude bibliography	On		