Music Genre Classification Techniques

Conference Paper · November 2021

CITATIONS
2

READS
3,099

2 authors, including:

Gautam Chettiar
University of Toronto
10 PUBLICATIONS **5** CITATIONS

SEE PROFILE

Music Genre Classification Techniques

Gautam Chettiar
SENSE
Vellore Institute of Technology
Vellore, Tamil Nadu, India
gautamsuresh.chettiar2019@vitstudent.ac.in

Kalaivani S
Department of Communication Engineering, SENSE
Vellore Institute of Technology
Vellore, Tamil Nadu, India
kalaivani.s@vit.ac.in

Abstract*— In today's world, where people are attached to their phones and air pods, listening to music becomes a mundane part of our lives. It occurs at times where we find particular songs catchy due to their pitch, choice of pattern, lyrics and much more! Hence in recommendation systems implemented in apps such as Spotify, classifying the music according to genres becomes very important to enhance user experience. *This paper aims to chart out various methods and parameters essential in the classification process, with use of Deep Learning techniques and an application of a Non-Linear Frequency Cepstrum. Music genres will be classified by taking FFT coefficients, followed by thy MFCC's and both coefficients will be used as inputs for Deep Learning models such CNN, RNN, KNN, Naïve Bayes Classifier and SVM, followed by a tabulation of the obtained results.

Keywords*— Music Genre Classification, MFCC, FFT, RNN, CNN, LSTM, SVM, Naïve Bayes, KNN.

I. Introduction

Since the application of apps such as Spotify, YouTube and many more, accessing various songs of our beloved artists has become a less tedious task. Now that we have access to online music libraries, how are they to be classified into sub-sections so that the listener can find more such similar songs to stay hooked on?

In actuality, every song is made by the artist who first composes the lyrics, then directs the song, chooses the instruments involved, the tempo, the drops and peaks and much more! In order to classify genres to music with complex analogies just by the provision of a simple audio file requires more than just a simple function. It is here, that we take to the concepts of Deep Learning.

In theory, it should be relatively easier to understand that providing more data to the classifier model should enhance its output prediction capabilities, and do a better job at getting the correct genre. But in order to understand which classifier to use, or which Deep Learning approach will yield better results, or what is the appropriate preprocessing required to obtain optimal results, we will incorporate various approaches and pre-processing techniques and attempt to classify the music genre.

The subsequent sections will provide a more in-detail insight of the modus operandi, and also elaborate on the theory of the approach wherever relevant.

The subsections will be in order as follows: 2) Preparing the Dataset 3) MFC vs FFT 4) Deep Learning Techniques 5) Conclusion 6) References.

II. Preparing the Dataset

A. Choosing the Dataset

For this paper, we have used the GTZAN Dataset which consists of 10 genres, each consisting of 100 audio tracks, each track having a duration of 30 seconds.

The dataset consists of pre-classified genres, namely blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae and rock. The benefit of having the genres mentioned respective to each class of audio files is it makes it easier to label the audio files with their genre name, which will help in training the Deep Learning model.

B. Pre-Processing

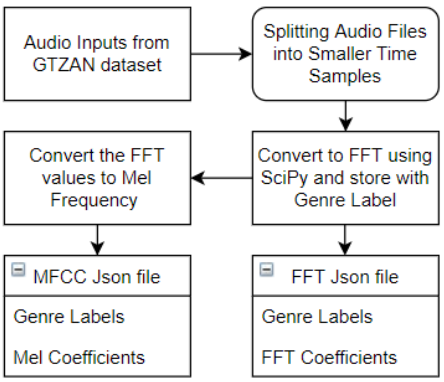
Before using the dataset as it is, we need to make a few tweaks to it, in order to obtain better results. Since our dataset is not very big, we can virtually increase the number of samples, by splitting the audio files into smaller samples, thereby increasing the number of samples. Now that we have more samples, we have more data for the model to work on.

For the sake of explanation, if we split the audio file vectors into 3 sub-vectors, each will have a duration of 10 seconds, and the total audio sample count triples, allowing us to operate on a greater number of samples.

C. Storing the data in appropriate format

Audio files can not be directly operated on, as important features such as power spectral density coefficients give better insights into frequency peaks, dips, the gap between two peaks, harmonics and much more.

Hence, we split the audio data, then using python's Librosa Library convert the time domain signal to frequency domain, and also use Librosa to obtain its Mel Frequency Cepstral Coefficients and begin training the Deep Learning model.



*Corresponding Author

III. MFC VS FFT

After choosing an appropriate dataset, preprocessing it , and taking the decision to classify on frequency domain analysis, we now need to consider the different approaches in frequency domain.

A. *Fast Fourier Transform*

The best and most used method for conversion of a signal from time domain to frequency domain is the Fourier transform. The frequency domain values of the signal can be obtained by using a rather simple mathematical formula

$$x[k] = \sum_{n=0}^{N-1} x[n] e^{\frac{-j2\pi kn}{N}}$$

FFT is a frequency transformation algorithm, it converts the time domain signal to frequency domain linearly, which is versatile but may not be optimum for our purpose as the human ear does not perceive sounds linearly. Simply put, humans hear sound in the logarithmic scale. To explain this better, let's say we hear a synthetically produced vibration at 100Hz and another vibration at 200Hz. Now the process is repeated again, but we are exposed to vibrations of 900Hz and then 1kHz. Naturally, we will feel that the difference between the first two frequencies is much more distinct than the latter 2. This analogy can be confirmed practically and proves that humans hear at a nonlinear logarithmic scale, and this is where MFC comes in.

B. *Mel Frequency Cepstrum*

The non linear analysis of frequency spectrum can be made possible using the Mel Frequency Cepstrum, which fundamentally is an algorithm which converts the linear frequency scale to a logarithmic scale which has its coefficients such that it tries to mimic the way human ear perceives sound.

$$Mel(f) = 2595 \log_{10}(1 + f/700)$$

Since now we have a more practical scale, we can convert the linearly obtained frequency into logarithmic scale values in order for a more practical input value to be provided to the MLP for training the model.

C. *To use MFC or FFT values for inputs*

In order to determine which frequency Cepstrum will help us get better results, we simply preprocess the data and append the FFT and MFCC values into two different json files, and then we use them individually in a Multi-Layered Perceptron model to determine which has an overall better performance

We use an MLP which reads inputs from the json file, which has been fundamentally set to such parameters that the model gets 'over-trained' or 'over-fitted' and then we train the model using a suitable train-test split. Overfitting allows us to push the model weight parameters to the limit to understand its adverse characteristics.

The table below shows the tabulated results on a 60-40 train-test-split on an over-trained MLP model.

TABLE I. PERFORMANCE: FFT VS MFCC

Sl no.	Accuracy of the MLP on GTZAN dataset	
	Frequency Spectrum	***Accuracy***
1.	FFT Coefficients	58.26%*
2.	Mel Cepstrum Coefficients	62.21%*

[a.] Values may vary mildly as weight update doesn't have fixed output

From this, it becomes clear that MFCC is a better approach for the task, and now we will proceed with solving the issue using inputs as MFCC's with different Deep Learning Models.

IV. DEEP LEARNING TECHNIQUES

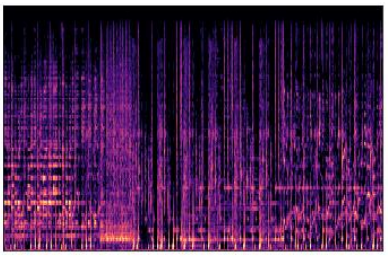
On simple analysis we realize that using MFCC will be better as the classifier input. So now we create the deep learning model according to the approach we desire. For this paper, where MFCC values along with their pre-classified genres as the labels are the inputs, we will train five types of models which are best suited for the case: Convolutional Neural Networks, Recurrent Neural Network, K Nearest Neighbors, Naïve Bayes Classifier and Support Vector Machine models and their performance will be noted.

Accuracy calculation for performance is as follows:

$$Accuracy\ \% = 100 * Correct / Total\ Guesses$$

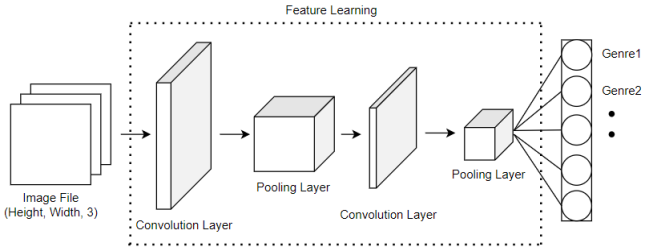
A. *Convolutional Neural Networks*

In this approach, we use the image files of the Mel spectrogram which we can generate for the values in dB provided with the GTZAN dataset as the input for the CNN model. The inputs look as shown below



Mel Spectrogram plot of the audio wav files

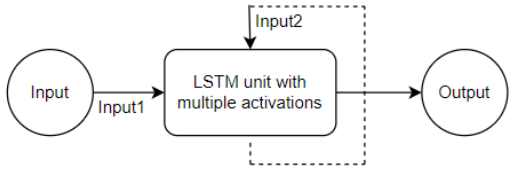
These images are used as inputs to the CNN model, which has 5 layers including soft-max function at the end. The model is then trained with a train test split of 25% with set to 30 epochs and trained.



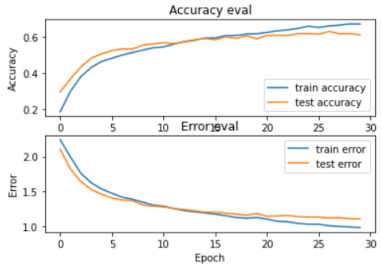
B. *Recurrent Neural Networks (LSTM)*

For RNN, we use the stored MFCC values as input. The model has 5 layers, total 58,000 trainable parameters and set to 30 epochs and trained.

Fundamentally, RNN LSTM can not be used to classify, however it can be used to predict future values, and then check the correlation between the predicted future values vs the actual future values and the accuracy can be set on correlational basis.



After weight training, now the model can predict the future values, and whichever corresponding genre has the highest correlational value is set as the classified genre.




```
79/79 - 2s - loss: 1.1821 - accuracy: 0.6107
```

C. *K- Nearest Neighbors*

Here we need to focus on 2 things mainly, one is the Principle Components (PCA) and the scatter-plot. KNN network tries to find the distance between a reference point on the scatter plot and the other points. The principle component can be anything.

The process is as follows. KNN is a supervised ML technique that takes in input as data points of any kind (here MFCC values) and calculate the distance between k (constant) number of pre-labelled data points. The classification is based on the distance score obtained from the nearest data points.

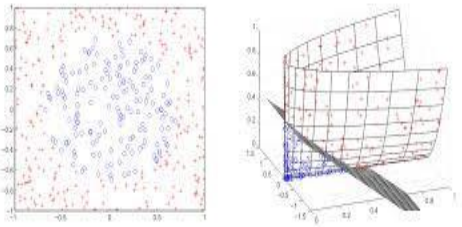
D. *Support Vector Machine*

Based on the principle of risk minimization and used widely in pattern recognition [1], we have the SVM constructing linear models to train the model and estimate decision functions only if the data is linearly separable. Basically it tries to fit a function that can separate the genre audio values and hence classify genre.

After feature extraction, working on fixed temporal lengths of time (by even splitting of dataset files) we allow the model to train having the SVM Kernel as Sigmoid sowing great results.

In this paper we have used SVM Kernel function as sigmoid, however [1] it has been observed that using Gaussian function has a much better performance than sigmoidal kernel, the same applies to the usage of polynomials as SVM Kernel.

$$k(x, y) = \tanh(\alpha x^T y + c)$$



*Kernel Classifies the genre by separating the points

E. *Naïve Bayer Classifier*

We store a few primary values which we would like to carry classifcation on, namely Short Time Energy, Spectral Centroid and Zero Crossing [2].

Naïve Bayes algorithm classifies on probability. After feature extraction, we use these parameters and store them in a json file. We now see Bayes Probability Theorem

$$P(A|B) = P(B|A) * P(A) / P(B)$$

We use the Naïve Bayes function from MATLAB to classify the genre after setting the parameters. Then we manually check for a few samples the accuracy of its prediction capabilities. We manually check for 5 songs each genre.

We now run all the Deep Learning Models on the Features we extract from the dataset, and begin training the model. The following table contains the accuracy of each of the models in classification of the genres from previously given train-test-split, and hence we obtain each methods performance.

The Naïve Bayes classifier only returns the probability of each genre's likeliness to be that songs correct genre. Hence we have to manually then read the probabilities, the highest probability indicating the genre, and then manually calculate the performance.

PERFORMANCE OF DIFFERENT METHODS

Sl No.	**Accuracy of different Deep Learning Methods**	
	Deep Learning Technique	***Accuracy***
1.	Convolutional Nerual Networks (CNN)	70.21%*
2.	Long Short Term Memory (RNN-LSTM)	61.07%*
3.	K-Nearest Neighbors (KNN)	66.43%*
4.	Naïve Bayes Classifier	56.66%
5.	Support Vector Machine (SVM)	70.66%*

*Values may differ a bit each time model is trained

CONCLUSION

As we can see most of the techniques fare comparatively well, some even go above 70%. The main thing to focus on is that the GTZAN Dataset does not contain enough number of audio samples to classify well enough, so having more data is necessary.

The tabulation of performance shows that relatively the Support Vector Machine and the Convolution Neural Network Models performed much better than the others. However, we must consider that our dataset was relatively much smaller than the actual number of songs that are present, hence increasing the dataset size, number of genres, usage of different features of the data may account to different Deep Learning Models outperforming these.

We also learn that by taking fundamental features from audio wav files, it is possible to learn about the features of music, and even possible to classify its genre and possible predict the future music just be having sufficient of its past values.

For future scope, we will try to use Deep Learning Techniques to identify the lyrics in the song and use the lyrics to classify the genre. Moreover, tabulated info about the song such as singer name, song description etc. can be used in classifying the genre with more precision.

REFERENCES

[1] R. Thiruvengatanadhan, “Music Genre Classification using SVM”, International Research Journal of Engineering and Technology (IRJET).

[2] Ardiansyah*, Boy Yuliadi**, Riad Sahara***, “Music Genre Classification using Naïve Bayes Algorithm”, International Journal of

[3] Computer Trends and Technology (IJCTT) – Volume 62 Number 1 – August 2018.

[4] Nilesh M. Patil1 , Dr. Milind U. Nemade, “Music Genre Classification Using MFCC, K-NN and SVM Classifier”, International Journal of Computer Engineering In Research Trends.

[5] Ahmet Elbir1 , Hilmi Bilal Çam2 , Mehmet Emre İyican2 , Berkay Öztürk2 , Nizamettin Aydın1 , “Music Genre Classification and Recommendation by Using Machine Learning Techniques”,

[6] Saad ALBAWI , Tareq Abed MOHAMMED, “Understanding of a Convolutional Neural Network”, 2017 International Conference on Engineering and Technology (ICET).

[7] N. Pelchat and C. M. Gelowitz, "Neural Network Music Genre Classification," in Canadian Journal of Electrical and Computer Engineering, vol. 43, no. 3, pp. 170-173, Summer 2020, doi: 10.1109/CJECE.2020.2970144.

[8] Music Genre Classification using Machine Learning Algorithms: A comparison Snigdha Chillara1, Kavitha A S2, Shwetha A Neginhal3, Shreya Haldia4, Vidyullatha K S5 International Research Journal of Engineering and Technology (IRJET)

[9] Y. Huang and L. Li, "Naive Bayes classification algorithm based on small sample set," 2011 IEEE International Conference on Cloud Computing and Intelligence Systems, 2011, pp. 34-39, doi: 10.1109/CCIS.2011.6045027.

[10] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf, "Support vector machines," in IEEE Intelligent Systems and their Applications, vol. 13, no. 4, pp. 18-28, July-Aug. 1998, doi: 10.1109/5254.708428.

[11] N. Jmour, S. Zayen and A. Abdelkrim, "Convolutional neural networks for image classification," 2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET), 2018, pp. 397-402, doi: 10.1109/ASET.2018.8379889.

[12] Yin, Qiwei & Zhang, Ruixun & Shao, XiuLi. (2019). CNN and RNN mixed model for image classification. MATEC Web of Conferences. 277. 02001. 10.1051/matecconf/201927702001.

[13] Guo, Gongde & Wang, Hui & Bell, David & Bi, Yaxin. (2004). KNN Model-Based Approach in Classification.

[14] M. A. Hossan, S. Memon and M. A. Gregory, "A novel approach for MFCC feature extraction," 2010 4th International Conference on Signal Processing and Communication Systems, 2010, pp. 1-5, doi: 10.1109/ICSPCS.2010.5709752.

[15] FEATURE EXTRACTION USING MFCC Shikha Gupta1 , Jafreezal Jaafar2 , Wan Fatimah wan Ahmad3 and Arpit Bansal4 Signal & Image Processing : An International Journal (SIPIJ) Vol.4, No.4, August 2013