Music Classfication Based on Genre

Satyaki Ghosh

M.Tech Student at IIT Guwahati, Roll number: 204102311

Abstract

In this project, it has been tried to classify small segments of music based on its genre. It is a difficult task because audio or speech signals usually need sequence models to process, which are often difficult to tune.Two different approaches are given here. One method involves calculating the MEL spectrogram from the audio samples and giving it as an input to a CNN. It gives the label as the output. Because audio signals are perceptually more relevant in log-scale, the use MEL scale seems useful here. And using a CNN on the MEL-spectrograms is a different idea than using sequence models. The second idea is to use hand-crafted features from the audio samples and then use three different classifiers on them. The features include both time-domain and frequency-domain features.

1 Introduction

With millions of songs (made of both lyrics and music) or just pieces of music present in so many diverse cultures, it will be helpful if they can be classified into some clusters. Obviously, some of them could be based on their languages (if lyrics is present) or based on their composers/singers. But some general form of classification could be based on their genre. Previously, it has been done using rhythm and harmonic content [1].

1.1 Background

In this project, I am using the idea of some papers which find the MEL spectrogram of the audio files and then use them as an image input (.jpg) to CNNs for classification into their respective genres [2], [3]. The other method involves extracting time and frequency domain features and then using as input to classifiers like Random Forest, Support Vector Machines (SVM) and Gradient Boosting.

1.2 Terminologies

Humans perceive sound intensity logarithmically. An ideal audio feature has a time-frequency domain representation, where both the amplitude and frequency

are perceptually relevant to human-hearing. Mel-spectrogram helps to achieve this. Mel-scale is a logarithmic scale, where equal distances on the scale have equal distance in perception (to the humans).It also satisfies the relation that 1000 Hz = 1000 Mel.

$$m = 2595.log(1 + \frac{f}{700}) \quad (1)$$

where m is in Mel-scale and f is in Hertz

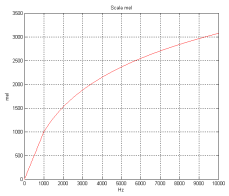


Figure 1: Mel scale vs Hz scale

1.3 Organization

The rest of the report is organized as follows:
Section 2 discusses the dataset and evaluation metrics used. Section 3 includes a Literature review of the previous methods used in this problem and research gap present, if any. Finally the proposed method is explained in brief. Section 4 describes the Methodology (algorithm and formulations), along with plots, tables and outputs. Finally in section 5, the conclusions of this report are mentioned.

2 Datasets used

The dataset by Gemmeke et.al is used here.[4].

2.1 Dataset description

It contains 10-second audio clips generated from 2.1 million Youtube videos. These are annotated into 527 classes including musical intruments, speech, traffic noise, etc. I have used the following 7 classes from these 527 classes. The raw audio clips are not given, rather the Youtube ID, start and end times are given. From this, the clips are downloaded as .mp4 files and saved as .wav files. The count of the audio samples used for each of the genres is listed below:

Genre	Count
Techno	402
Pop_music	392
Rhythm_blues	348
Rock_music	161
Hip_hop_music	112
Vocal	89
Reggae	29
TOTAL	1533

2.2 Evaluation Metrics

The following evaluation metrics are used:
1. Accuracy: The fraction of correctly classified samples.
2. F1-score: The harmonic mean of precision and recall.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

3 Literature

Music genre classification is a interesting topic. Many previous research has been done on this and some of them are discussed here.

3.1 Previous methods

Tzanetakis (2002) had applied GMM and k-NN classifiers on features extracted from the timbral structure, rhythmic content and pitch content which also included MFCC (Mel Frequency Ceptsral Coefficients) [1]. Nanni et.al (2016) used both visual and audio features on SVM and AdaBoost classifiers [5].

With the advent of Deep Learning, deep neural networks have also been used in this context. Due to high sampling frequency of audio-signals, it becomes difficult to use it as an input (since the sequence becomes too long) to an RNN (Recurrent Neural Network) which usually performs well on sequence data like text. However spectrograms contain both time and frequency domain information in a concise format. Wyse et al. used spectrograms as image input to CNNs in this context.

3.2 Research gaps or loopholes in the previous methods

HMM models, popular for speech recognition, have also been used for this purpose. Similarly many popular models like GMM, k-NN, SVM, etc have been used. But use of deep learning networks was not common in this problem statement before until now when the abundance of data has hugely developed these models. Hand-crafted features (time and frequency domain) are not always able to capture the details to make the classification.

3.3 Proposed Method

In this project, the idea of using pre-trained deep convolutional neural network models on Mel-spectrogram, has been taken from some of the recent papers published in this domain [2], [6]. Also, for a parallel comparison, some hand-crafted features are given to classifiers and compared with the above model.

4 Methodology

Methodology of the report is discussed in the following sections.

4.1 Algorithm

First we apply a pre-emphasis filter to boost the high frequency components.

$$y(t) = x(t) - \alpha x(t-1) \tag{4}$$

where the constant is taken here as 0.97. Next, for the method employing CNN, we calculate the MEL spectrogram.

1. Extraction of Short term Fourier Transform taking small windows
2. Converting amplitude dB
3. Converting frequencies to Mel scale
   (a) Choose number of Mel bands
   (b) Construct Mel filter banks
   (c) Apply the filters to the spectrogram

Here, we take the following values:

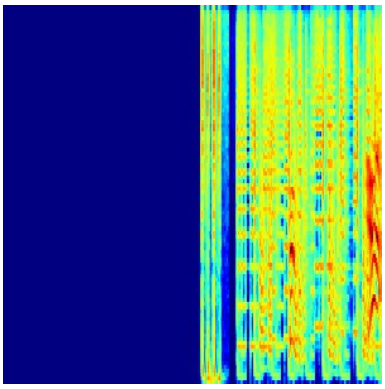
- Number of Mel filter banks = 96
- Sampling rate = 22050
- FFT window size = 2048

The pre-trained CNN model VGG_16 has been loaded. Its output is flattened and given to a dense layer with dropout layer which connects to the output layer with softmax activation. The model weights of the VGG-16 model are not trained. Only the dense hidden layer and output layers are trained. This is transfer learning.
The cross-entropy loss is calculated as

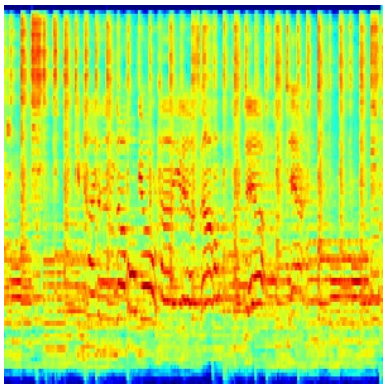
$$L = -\sum_{c=1}^{M} y_{o,c} * log p_{o,c} \tag{5}$$

where M is the number of classes; $y_{o,c}$ is 1 if observation o belongs to class c and 0 otherwise; $p_{o,c}$ is the model's predicted probability that observation o belongs to class c.

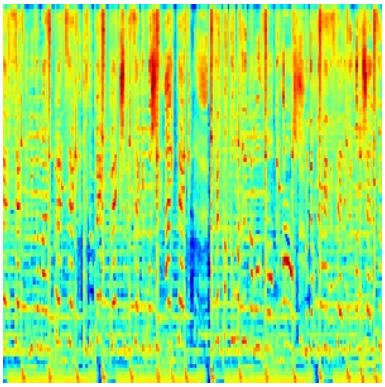
1. **Time-domain features**
   - Central moments: mean, standard deviation, kurtosis
   - Zero crossing rate



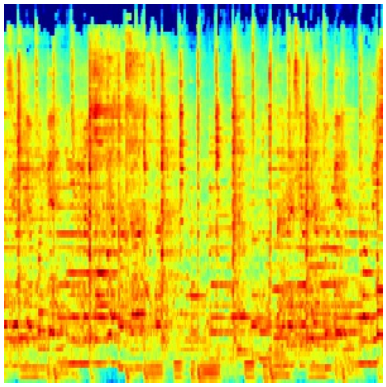
(a) Reggae



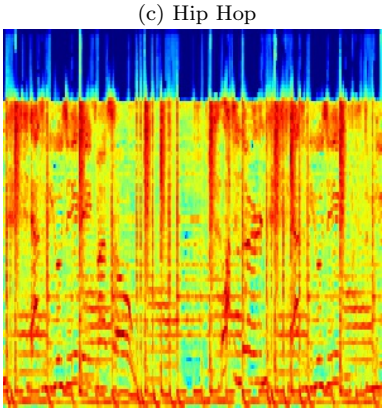
(b) Rock



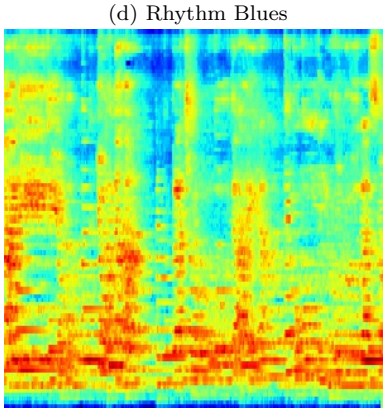
(c) Hip Hop



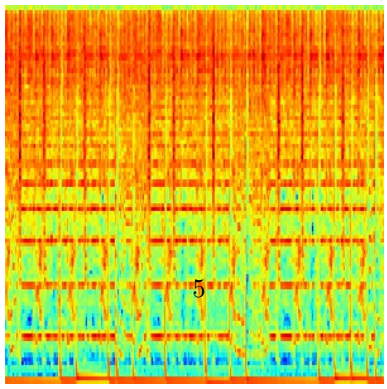
(d) Rhythm Blues



(e) Pop



(f) Vocal



(g) Techno

Figure 2: Spectrogram of various genres of music

- Root Mean Squared Error

2. **Frequency-domain features**

   - Mel-Frequency Cepstral Coefficients (MFCC): It is similar to finding the Mel spectrogram. After applying the filter banks, the Discrete Cosine Transform of the logarithm of the filterbank energies are calculated.
   - Spectral Contrast: The windows are divided into several frequency bands and in each such band the difference in maximum and minimum frequency is taken.

4.2 Formulations

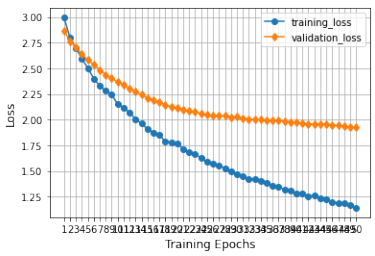
The spectrogram images are 216 x 216. There are 1379 training images, 77 testing images, and 77 validation images. After the VGG-16 model, a 512-unit hidden layer is implemented. In this problem, over-fitting was encountered. So the following were implemented:

- L2-Regularization - the loss term is added with $\frac{1}{2}\lambda \sum w^2$ where λ is set to a small value
- Dropout - weights of random chosen neurons are set to 0 during training iteration

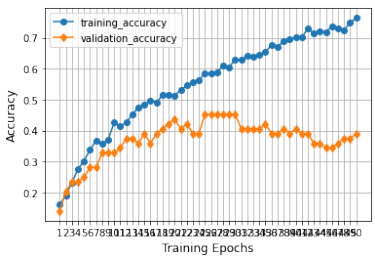
Batch-size used is 32, optimizer is Adam, and number of epochs is 50. The loss decreases sharply at first, and then it becomes very slow.

4.3 Plots tables and outputs

- Loss: The training loss keeps decreasing but the validation loss gets saturated.
- Accuracy: It gives a similar plot to Loss.



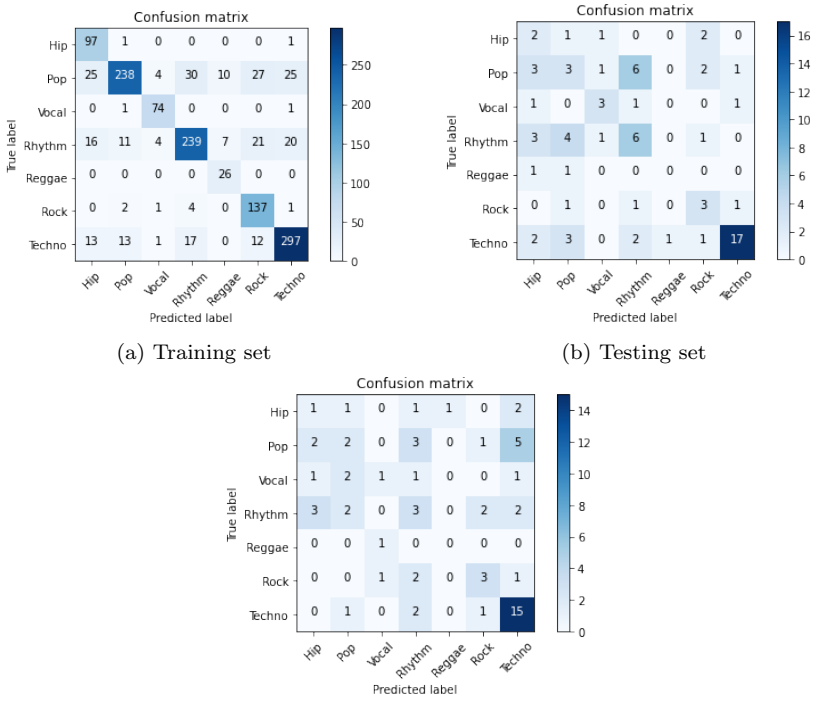
(a) Loss vs Epochs



(b) Accuracy vs Epochs

Figure 3: Loss and accuracy variation with epochs 1 to 50

- Confusion Matrix of the Training, Testing and Validation sets are shown in Figure 4. The Techno class is classified with highest accuracy.



(a) Training set

(b) Testing set

(c) Validation set

Figure 4: Confusion Matrices

Model	Accuracy	F-score
CNN with spectrogram	0.44	0.45
Random Forest	0.42	0.25
Gradient Boosting	0.44	0.34
SVM	0.34	0.2

5 Conclusions

The CNN model did not perform satisfactorily better than the other feature-based classifiers maybe due to lack of data. The total data used was about 1.5 GB and since these were run on simple CPU, a high volume of data could not be used. Still it gives an accuracy of 0.44 which is much better than random guessing which is $\frac{1}{7}$ or about 0.14. Some of the other factors or hyperparameters which can be tried with more different combinations are:

- Number of layers in the CNN
- Number of neurons in each layer
- Activation function for hidden layers
- Number of epochs to run
- Learning rate
- Batch size

The feature-based classifiers can also improve by trying out different other time and frequency domain features.

References

[1] G. Tzanetakis and P. C. 2002, "Musical genre classification of audio signals," *IEEE Transactions on speech and audio processing 10(5):293– 302.*, 2002.

[2] M. R. Nirmal and B. S. S. Mohan, "Music genre classification using spectrograms," *2020 International Conference on Power, Instrumentation, Control and Computing (PICC) 10.1109/PICC51425.2020.9362364*, 2020.

[3] H. Bahuleyan, "Music genre classification using machine learning techniques," *arXiv:1804.01149*, 2018.

[4] J. Gemmeke, D. F. Daniel PW Ellis, W. L. Aren Jansen, M. P. R Channing Moore, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on. IEEE, pages 776–780*, 2017.

[5] L. Nanni, A. L. Yandre MG Costa, and S. R. B. Moo Young Kim, "Combining visual and acoustic features for music genre classification," *Expert Systems with Applications 45:108–117.*, 2016.

[6] L. Wyse, "Audio spectrogram representations for processing with convolutional neural networks.," *arXiv preprint arXiv:1706.09559*, 2017.