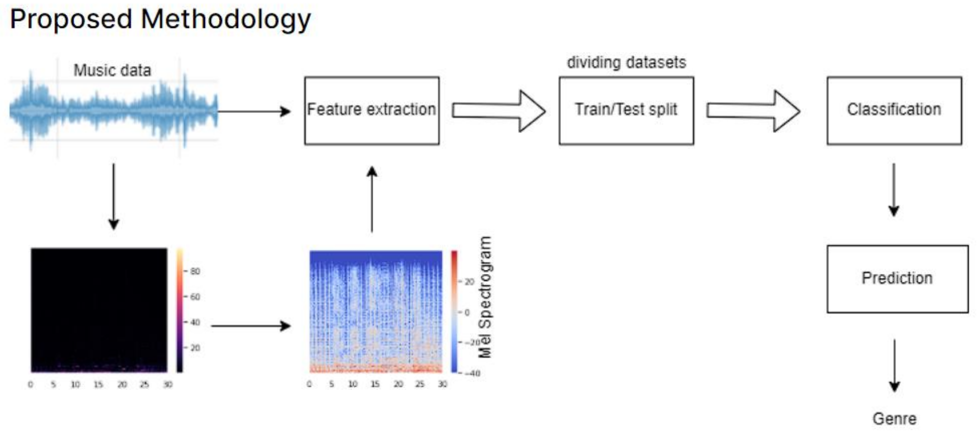
GENRE BASED AUDIO CLASSIFICATION OF MUSIC

Ankur Kumar
Dept. EE, IIT Kanpur

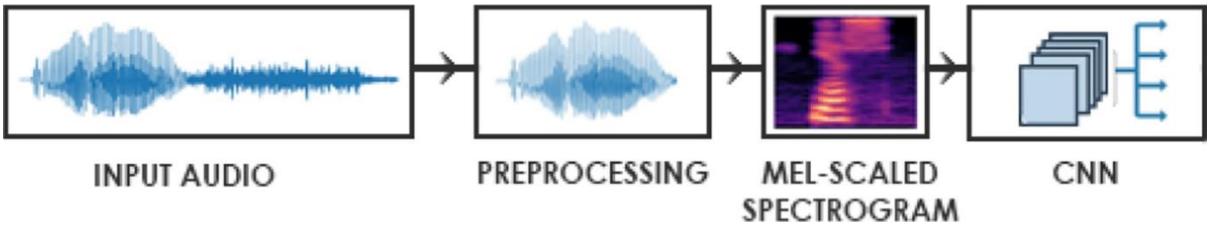
Dr. Vipul Arora
Dept. EE, IIT Kanpur

Introduction:

With the ever so growing archive of media on the internet it becomes a necessity to be able to categorize and manage databases of such massive size efficiently and there is a very obvious need for a reliable automation for this. Such a task is to classify music on basis of genre (Rock, Jazz,etc.) , Context(Literature, History, etc.) and many more.



In this paper we took on the task to classify on the basis of Genre specifically. Another aspect of the research was figuring out if the model will perform differently on music from different cultural backgrounds. We train the model on western music dataset **(GZTAN dataset)** which we can test on Indian music datasets for any drastic performance differences.

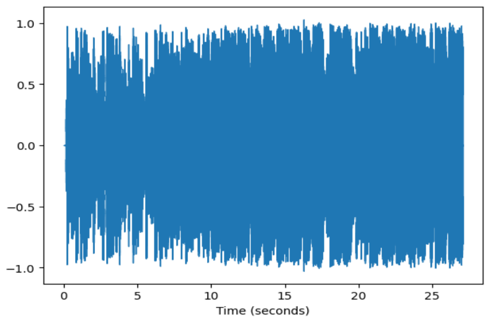


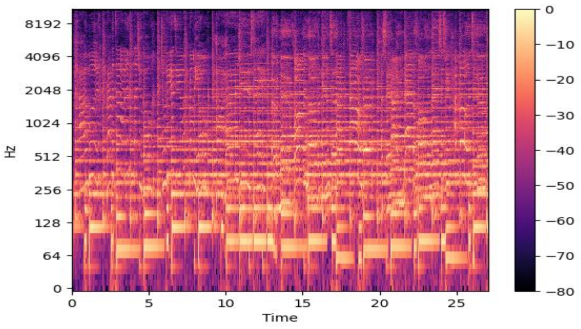
Model Architecture Overview:

For this problem statement we shall be using a Convolutional Neural Network Described below which will take as input Spectrogram images of the audio files and output its prediction into different genres.

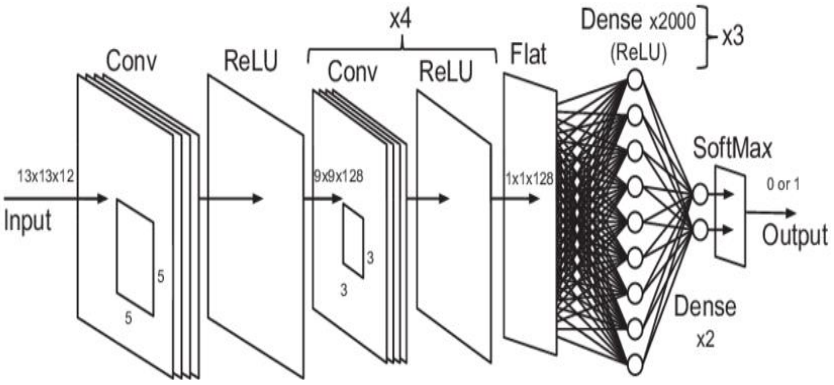
The GZTAN dataset contains annotated audio files(.wav) from 10 genres with 100 audio files on each having time interval of 30. We firstly split the dataset into 900 files(for training) and 100(for final testing) then do the preprocessingAll tracks are 22500Hz, Mono 16-bit audio files in .wav format (Strum, 2013). Therefoore we have taken sampling rate of 44kHz.

1. **Pre processing:** Firstly we take the audio files and generate the spectrogram from those audio files using Librosa library in python. This is achieved by doing the Fourier Transform on the Time domain waveform of the audio signal and converts it into the frequency domain. As we can see below the time domain rarely provides any insight on the patterns/melodies in the audio but in the spectrogram we see bare details of patterns emerging. We further created a plot of the spectrogram (Y axis is frequency, X axis is time, colour represents decibels for all respective frequencies). Now we create a tensor of size(IMG_SIZEx3), the 3 represents the RGB pixel values. We shall use this tensor for input.





2. The Fourier transform is done by dividing the 30 second audio into windows with sample frequency and using Librosa library of python. Then it is further transformed into **MEL spectrogram** to better match with human hearing (similar to the Decibels)



3. **CNN Model:**
    a. The model we use is called Convolutional Neural Network which comprises of first layer in which we rescale the pixel values from [0,255] range to [0,1] and feed the data into layers of convolutional network described below (diagram above):

        1. 16 filters 5x5 kernels with **ReLU** activation
        2. Next four layers are of kernel sizes 32, 64, 128, 512 respectively with 5x5 kernels
        3. Then we flatten the matrix into 1D after reducing the size by MaxPooling (taking local average and decreasing size)
        4. Finally we pass it through the Dense layers of fully connected neurons. We also use a dropout layer which mutes 20% of random neurons to introduce randomness and avoid overfitting
        5. Lastly we use Cross Entropy loss function since this is Multi Class categorization problem and hence requires dealing with final prediction probabilities

<u>Loss Function:</u>

$$L_{\text{CE}} = -\sum_{i=1}^{n} t_i \log(p_i), \quad \text{for n classes,}$$

where t_i is the truth label and p_i is the Softmax probability for the i^{th} class.

Optimizations:

The Conv2D layers employ kernel convolution to process the input matrix, where the kernel traverses the entire input, performing multiplications to generate a new matrix. This iterative process refines and emphasizes the patterns present in the spectrogram. Initially, we overlooked this step and solely utilized dense layers, resulting in significantly lower accuracy compared to our current approach.

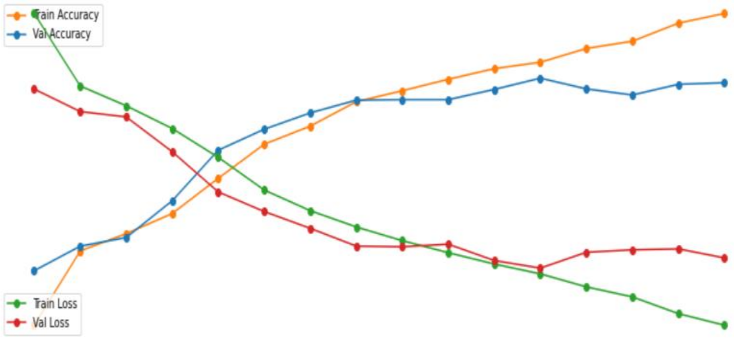
This observation can be explained by considering that directly converting the data into a 1D list discards the valuable information regarding the 2D relationship between neighbouring pixels in both vertical and horizontal directions. Consequently, crucial characteristics inherent in these relationships are not captured.

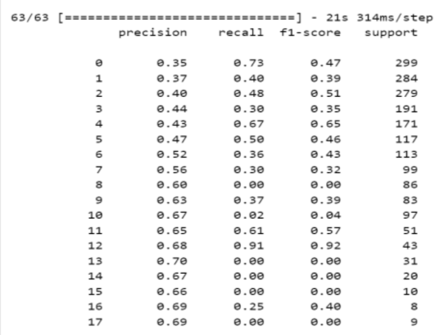
Also we will be storing the models at every epoch and maintain an object “best_model” which will be updated at every step to make sure we don’t lose progress in case of malfunction. The learning as we can expect plateaus after a certain point and we stop the learning (Early Stopping) if the change in accuracy is very small (0.2 here) to avoid overfitting.

Results and Conclusion:

As we can see from the graph the accuracy plateaus at about 69% after 20 iterations. Also the training and validation accuracy follow closely confirming that our dataset was grouped correctly and that there was very negligible bias (if any).

Also while training, we only used the spectrogram as input, ignoring other usable audio features like pitch, beats,etc. for simplicity. We can use something called Multi Column- DNN and pass the respective features in their columns and finally combine the results by HMM *(referred *[1]).* This is most helpful for high pitch resolution input





```
63/63 [==============================] - 21s 314ms/step
              precision    recall  f1-score   support

           0       0.35      0.73      0.47       299
           1       0.37      0.40      0.39       284
           2       0.40      0.48      0.51       279
           3       0.44      0.30      0.35       191
           4       0.43      0.67      0.65       171
           5       0.47      0.50      0.46       117
           6       0.52      0.36      0.43       113
           7       0.56      0.30      0.32        99
           8       0.60      0.00      0.00        86
           9       0.63      0.37      0.39        83
          10       0.67      0.02      0.04        97
          11       0.65      0.61      0.57        51
          12       0.68      0.91      0.92        43
          13       0.70      0.00      0.00        31
          14       0.67      0.00      0.00        20
          15       0.66      0.00      0.00        10
          16       0.69      0.25      0.40         8
          17       0.69      0.00      0.00         9
```

An extention of this research can be to use a mask that filters out the vocals from the instrumentals from the audio files and then use just the instrumentals (or just the vocals) and compare the accuracy with the counterparts to observe what plays more important role.

1. This classification after more refinement can be used in All India Radio which can be deployed by creating a pleasing UI for the users to access the data
2. Also it can be useful in more important sectors like assessing heartbeat patterns or ECG waveform in brain signals to look for similarities in diseases and medical conditions.

References:

[1]. Hariharan Subramanian, Audio Signal Classification

[2]. MDan-Ning hang, Lie Lu, Music type classification by Spectral Contrast

[3]. George Tzanetakis, Automatic Musical Genre Classification of Audio Signal

[4]. Sangeun Kum, Melody Extraction on Vocal Segments using MC-DNN