MUSIC GENRE DETECTION DESCRIPTION

by

AJAY SHIVHARE

A capstone project submitted to the

Rutgers Business School

Rutgers, The State University of New Jersey

In partial fulfilment of the requirements

For the degree of

Master of

Information Technology and Analytics

Written under the direction of

Dr. Mehmet Turkoz

Newark, New Jersey

MAY, 2020

ABSTRACT OF THE CAPSTONE PROJECT

Music Genre Detection (Music Information Retrieval) Description

By AJAY SHIVHARE

Capstone Project Director

Dr. Mehmet Turkoz

Audio files are an important source in Machine Learning for information extraction. Music Information Retrieval - MIR is the interdisciplinary science of retrieving information from music. MIR is a small but growing field of research with many real-world applications. Those involved in MIR may have a background in musicology, psychoacoustics, psychology, academic music study, signal processing, informatics, machine learning, optical music recognition, computational intelligence or some combination of these.

For this project - We aim to make the computer detect the genre of music (.mp3 file) with utmost accuracy.

Table of Contents

1. INTRODUCTION

Automatic genre classification of music is an important topic in Music Information Retrieval with many interesting applications. A solution to genre classification would allow for machine tagging of songs, which could serve as metadata for building song recommenders.

There has been an explosion of musical content available on the internet. Some sites, such as Spotify and Pandora, carefully curate and manually tag the songs on their sites. Other sources, such as YouTube, have a wider variety of music, but many songs lack the metadata needed to be searched and accessed by users. One of the most important features of a song is its genre.

Automatic genre classification would make hundreds of thousands of songs by local artists available to users and improve the quality of existing music recommenders on.

In this project, we investigate the following question: Given a song, can we automatically detect its genre? We look at spectrogram of the audio file to determine its genre.

2. BACKGROUND

2.1 Structure for Genre Detection Project

Our project is divided over 6 files named and described below :

1) *spectrogram_playmusic.ipynb* - Contains the spectrograms of distinct genres.
2) *load_fma_dataset.ipynb* - The FMA dataset is loaded & feature engineering was performed.
3) *Convert_to_npz.ipynb* - Code to convert .mp3 music files to npz.
4) *CRNN_model.ipynb* - Deep Learning Models CRNN
5) *CNN_RNN_Parellel.ipynb* - Deep Learning Model Parallel CNN RNN
6) *Heuristics.ipynb* - Output based on above two deep learning models.

2.2 Data Set

In the case of Music, there are a few different datasets with data — GTZan and Million Songs dataset (MSD) are 2 of the ones most commonly used. But both of these data sets have limitations. GTZan only has 100 songs per genre and MSD has well 1 million songs but with only their metadata, raw audio files are not available.

We have decided to use the *Free Music Archive Small dataset:*
(Link: https://github.com/mdeff/fma)
The Free Music Archive (FMA), an open and easily accessible dataset suitable for evaluating several tasks in MIR & concerned with browsing, searching, and organizing large music collections. The FMA small data set that we will be using has 8 genres and 1000 songs per genre evenly distributed (balanced dataset) with 30-second audio files & related meta-data. The eight genres are Electronic, Experimental, Folk, Hip-Hop, Instrumental, International, Pop, and Rock. FMA Small is split into a training set of 6400 songs, a validation set of 800 songs & test set of 800 songs.

2.3 Feature Set from Data Set

Spectrogram:

- Reason for choosing spectrogram - The spectrogram is a powerful tool to identify sound features either in real-time or in post-processing on recorded materials. Its utility however strongly depends on the type of audio material to be examined and on the accurate tuning of the analysis parameters. In some cases, it is powerful enough to recognize the features of interest more accurately and objectively than human hearing.

- Spectrogram Representation of 8 Genre –

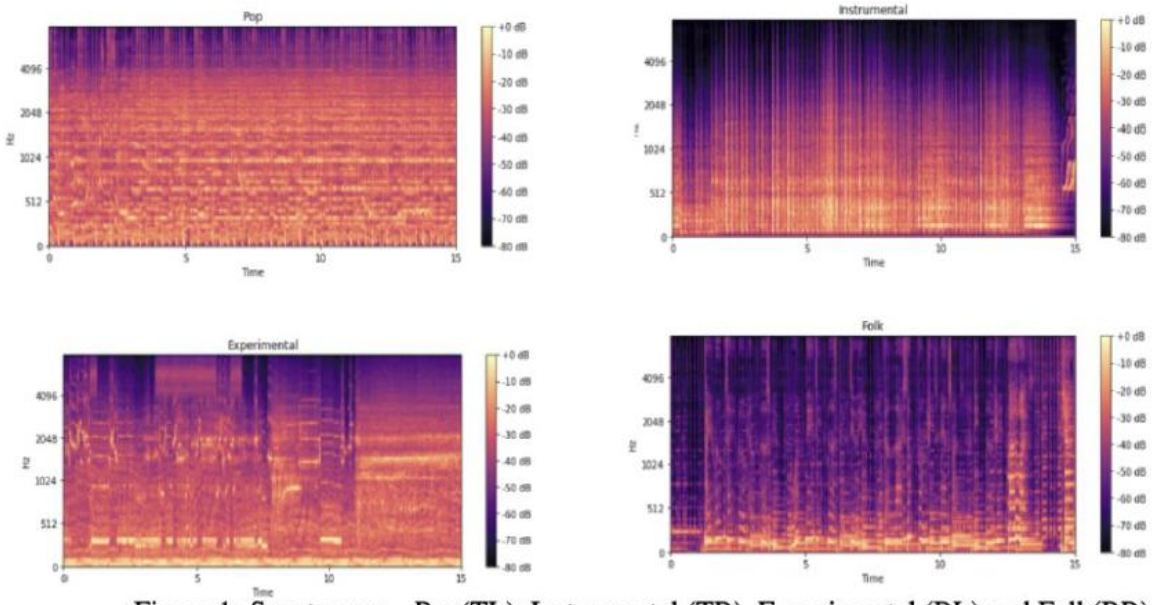


Figure 1: Spectogram - Pop(TL), Instrumental (TR), Experimental (BL) and Folk(BR)

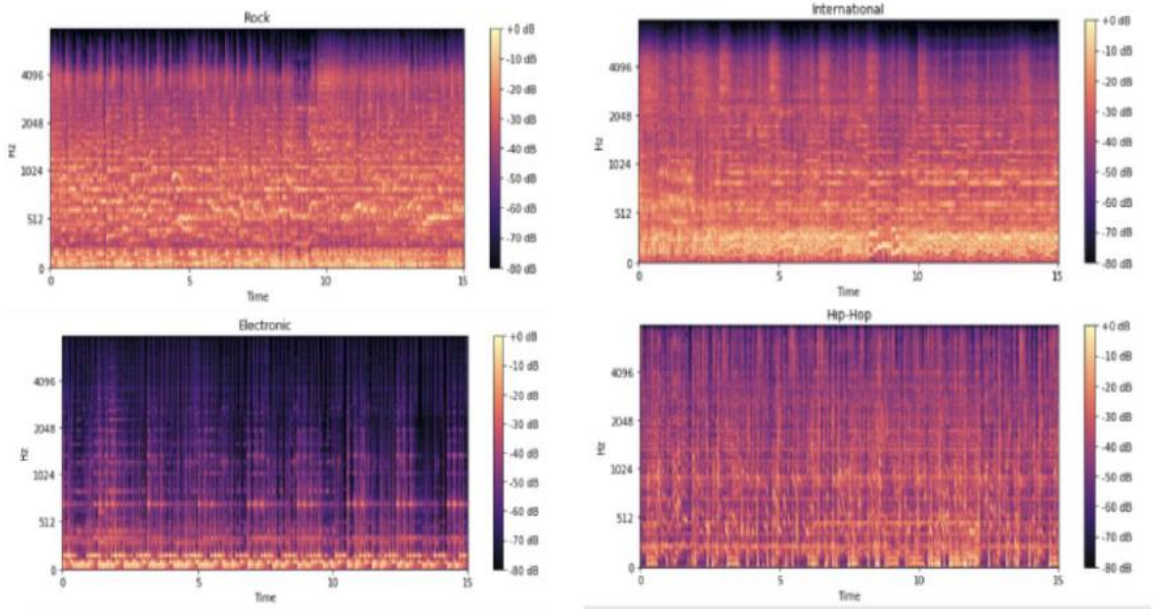


Figure 2: Spectogram - Rock(TL), International (TR), Electronic (BL) and Hip Hop(BR)

2.4 Steps implemented for a Genre Classification

- Build Deep Learning Models:

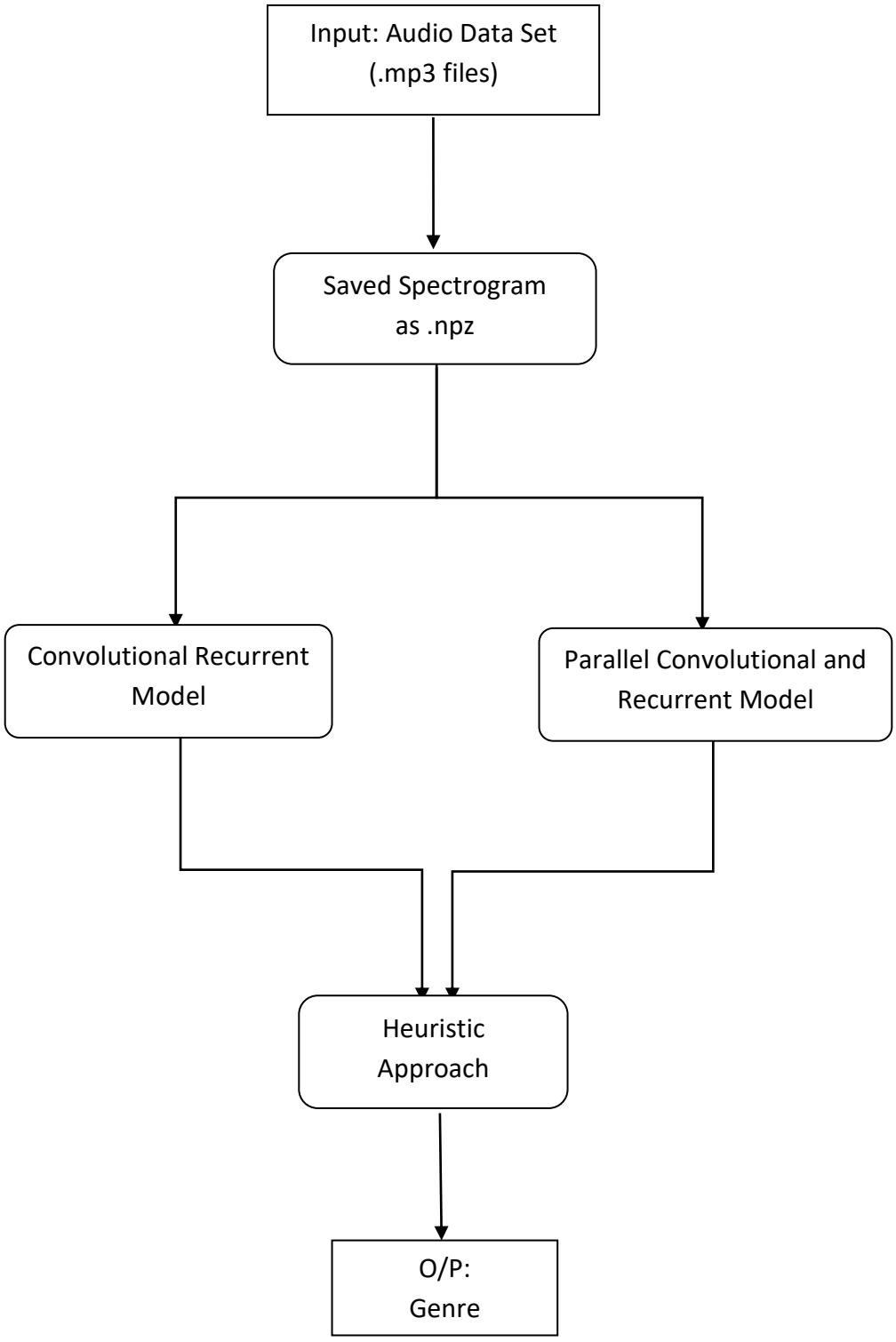
  We will be building the feature set for DL Models by **converting the audio files into a spectrogram**. A spectrogram shows the frequencies that make up the sound, from low to high, and how they change over time, from left to right.

  After that, we aimed to try ***Convolutional Recurrent Model & Parallel Convolutional and Recurrent Model*** to work on the problem of (basically) spectrogram image classification.

- Heuristics to compare accuracy:

  After we build our deep learning models with the spectrogram data.
  We aim to apply heuristics and give weights to the deep learning model to produce the highest accuracy.

2.5 Flowchart of Model



3. METHODOLOGY

3.1 Converting audio data into Mel-Spectrogram

- Each audio file was converted into a spectrogram which is a visual representation of the spectrum of frequencies over time.

- A regular spectrogram is squared magnitude of the short-term Fourier transform (STFT) of the audio signal.

- **This regular spectrogram is squashed using the Mel scale**. We used the built-in function in the librosa library to convert the audio file directly into a Mel spectrogram.

- The important parameters used in the transformation are — window length which indicates the window of time to perform Fourier Transform on and hop length which is the number of samples between successive frames.

- The typical window length for this transformation is 2048 which converts to about 14.64ms (10ms approx.), the shortest reasonable period a human ear can distinguish. We chose the hop length of 512. determine loudness in decibels (dB) as it relates to the human-perceived pitch. As a result of this transformation, each audio file gets converted to a Mel-spectrogram of shape — 640, 128.

3.2 Parallel CNN-RNN Model

- The key idea behind this network is that even though CRNN has RNNs to be the temporal summarizer, it can only summarize temporal information from the output of CNN's. The temporal relationships of original musical signals are not preserved during operations with CNN's.

- This model passes the input spectrogram through both CNN and RNN layers in parallel, concatenating their output and then sending this through a dense layer with SoftMax activation to perform classification as shown below. The convolutional block of the
- model consists of a 2D convolution layer followed by a 2D Max pooling layer.

- This is in contrast to the CRNN model that uses 1D convolution and max-pooling layers. There are 5 blocks of Convolution Max pooling layers. The final output is flattened and is a tensor of shape None, 256.

- The recurrent block starts with 2D max pooling layer of pool size 4,2 to reduce the size of the spectrogram before LSTM operation. This feature reduction was done primarily to speed up processing. The reduced image is sent to a bidirectional GRU with 64 units. The output from this layer is a tensor of shape None, 128.

- The outputs from the convolutional and recurrent blocks are then concatenated resulting in a tensor of shape, None, 384. Finally, we have a dense layer with SoftMax activation.

- The model was trained using RMSProp optimizer with a learning rate of 0.0005 and the loss function was categorical cross entropy. The model was trained for 50 epochs and Learning Rate was reduced if the validation accuracy plateaued for at least 10 epochs.

- ***Accuracy using two highest probabilities in the output (dense - softmax layer) is : 68.875%.***

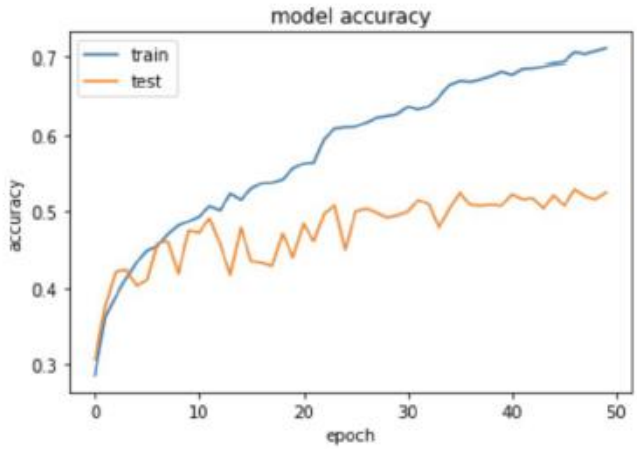
In [79]:
```
from itertools import chain
from collections import Counter
c = Counter(tuple(x) for x in iter(name_top_two))
#sorted(c)
sorted(c.items(), key=lambda pair: pair[1], reverse=True)
```

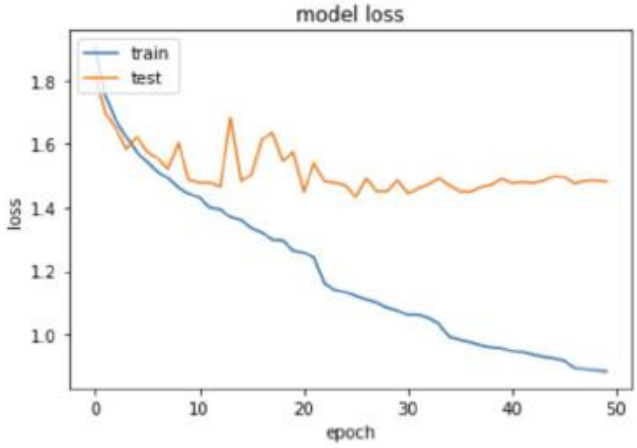
Out[79]:
```
[(('Hip-Hop', 'International'), 57),
 (('Rock', 'Pop'), 54),
 (('Experimental', 'Instrumental'), 49),
 (('Electronic', 'Experimental'), 43),
 (('Instrumental', 'Experimental'), 39),
 (('Folk', 'International'), 32),
 (('Rock', 'Experimental'), 32),
 (('Folk', 'Experimental'), 29),
 (('Experimental', 'Electronic'), 28),
 (('Electronic', 'Hip-Hop'), 28),
 (('Hip-Hop', 'Electronic'), 27),
 (('Folk', 'Pop'), 26),
 (('International', 'Pop'), 25),
 (('International', 'Experimental'), 24),
 (('Folk', 'Instrumental'), 24),
 (('International', 'Hip-Hop'), 19),
 (('International', 'Folk'), 19),
 (('Electronic', 'International'), 19),
 (('Experimental', 'Rock'), 16),
 (('Pop', 'Rock'), 16),
 (('International', 'Electronic'), 14),
 (('Rock', 'International'), 12),
 (('Instrumental', 'Folk'), 12),
 (('Hip-Hop', 'Pop'), 11),
 (('Experimental', 'Folk'), 11),
 (('International', 'Rock'), 10),
 (('Pop', 'International'), 10),
 (('Instrumental', 'Electronic'), 10),
 (('Electronic', 'Pop'), 10),
 (('Experimental', 'International'), 9),
 (('Rock', 'Folk'), 8),
 (('Pop', 'Experimental'), 8),
 (('Experimental', 'Hip-Hop'), 8),
 (('Pop', 'Folk'), 8),
 (('Instrumental', 'Rock'), 7),
 (('Hip-Hop', 'Experimental'), 7),
 (('Instrumental', 'Pop'), 6),
 (('Experimental', 'Pop'), 6),
 (('Pop', 'Instrumental'), 4),
 (('Rock', 'Instrumental'), 4),
 (('Folk', 'Rock'), 3),
 (('Electronic', 'Instrumental'), 3),
 (('Rock', 'Hip-Hop'), 2),
 (('Pop', 'Electronic'), 2),
 (('Rock', 'Electronic'), 2),
 (('Folk', 'Hip-Hop'), 2),
 (('Hip-Hop', 'Instrumental'), 1),
 (('Hip-Hop', 'Folk'), 1),
 (('Instrumental', 'International'), 1),
 (('Electronic', 'Rock'), 1),
 (('Hip-Hop', 'Rock'), 1)]
```

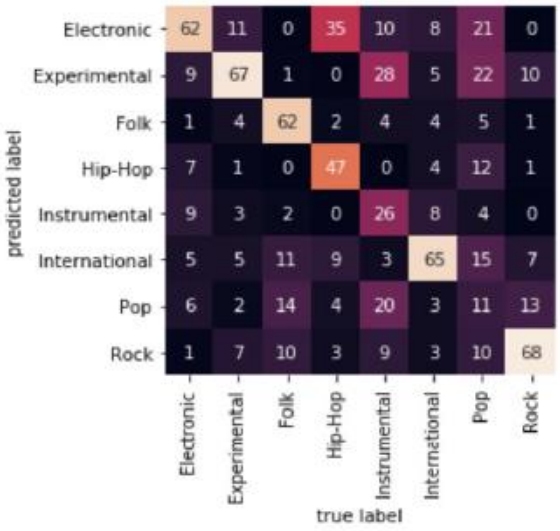
- **Class Wise Accuracy Score** -
  - Electronic - 62%
  - Experimental - 67%,
  - Folk - 62%
  - Hip Hop - 47%
  - Instrumental -26%
  - International - 65%
  - Pop - 11%
  - Rock - 68%

- ***Overall Accuracy of the Model - 51%.***

- Model Loss, Accuracy and Confusion matrix of CNN-RNN Model.







- **Reasons for CNN & RNN based network**

  - For image classification, the local images are correlated thereby producing nearby pixels to have similar intensities & colours. In spectrogram analysis, there is often harmonic correlation which is spread along the frequency axis while local correlation may be weaker.

  - We chose the above-mentioned model because the **convolutional model is good with image recognition task** & on the other hand **RNN excels in understanding the time series data**. Cause in music time t is dependent on time t -1.

3.3 Convolutional Recurrent Model

- This model uses 1D CNN that performs convolution operation just across the time dimension. Each 1D convolution layer extracts features from a small slice of the spectrogram. RELU activation.

- This model uses 1D CNN that performs convolution operation just across the time dimension. Each 1D convolution layer extracts features from a small slice of the spectrogram. RELU activation is applied after the Convolution operation. Batch normalization is done and finally, 1D Max Pooling is performed which reduces the spatial dimension of the image and prevents overfitting.

- This chain of operations — 1D Convolution — RELU — Batch Normalization — 1D Max Pooling is performed 3 times.

- The output from 1D Convolution Layer is fed into an LSTM which should find short term and long-term structure of the song. The LSTM uses 96 hidden units. The output from LSTM is passed into a Dense Layer of 64 units.

- The final output layer of the model is a dense layer with SoftMax activation and 8 hidden units to assign a probability to the 8 classes. Both dropout and L2 regularization were used between all the layers to reduce overfitting of the model.

- ***Accuracy using two highest probabilities in the output (dense - SoftMax layer) is: 70.625%.***

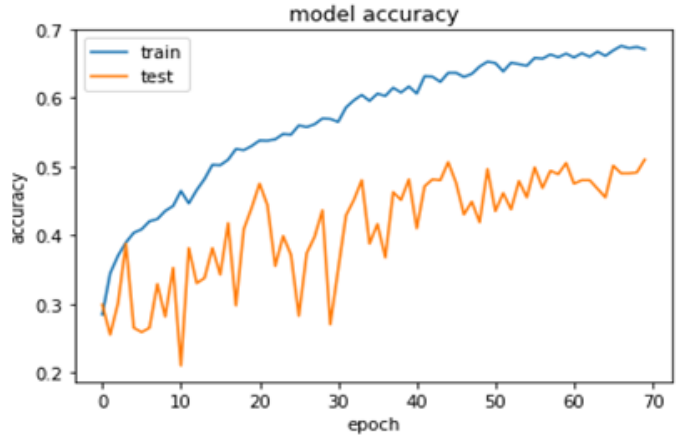
In [20]:
```
from itertools import chain
from collections import Counter
c = Counter(tuple(x) for x in iter(name_top_two))
#sorted(c)
sorted(c.items(), key=lambda pair: pair[1], reverse=True)
```

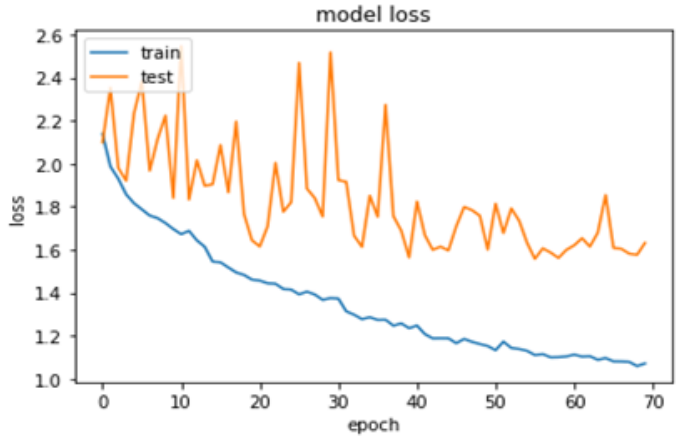
Out[20]:
```
[(('International', 'Pop'), 78),
 (('Electronic', 'Hip-Hop'), 74),
 (('Rock', 'Pop'), 70),
 (('Experimental', 'Instrumental'), 55),
 (('Folk', 'Pop'), 51),
 (('Hip-Hop', 'Electronic'), 36),
 (('Electronic', 'Experimental'), 32),
 (('Experimental', 'Rock'), 31),
 (('Pop', 'Rock'), 25),
 (('Hip-Hop', 'International'), 24),
 (('Experimental', 'Electronic'), 23),
 (('International', 'Folk'), 22),
 (('Rock', 'Instrumental'), 21),
 (('Instrumental', 'Experimental'), 20),
 (('Folk', 'International'), 19),
 (('Experimental', 'Pop'), 19),
 (('Rock', 'Experimental'), 17),
 (('Pop', 'International'), 15),
 (('Instrumental', 'Electronic'), 15),
 (('Electronic', 'Instrumental'), 14),
 (('Electronic', 'International'), 14),
 (('Instrumental', 'Folk'), 13),
 (('Pop', 'Folk'), 12),
 (('Electronic', 'Pop'), 12),
 (('Hip-Hop', 'Pop'), 8),
 (('International', 'Electronic'), 8),
 (('International', 'Experimental'), 7),
 (('Pop', 'Electronic'), 7),
 (('Pop', 'Experimental'), 7),
 (('Pop', 'Instrumental'), 6),
 (('Experimental', 'Folk'), 6),
 (('Experimental', 'International'), 6),
 (('Folk', 'Experimental'), 6),
 (('Instrumental', 'Pop'), 4),
 (('Hip-Hop', 'Experimental'), 4),
 (('Folk', 'Instrumental'), 4),
 (('International', 'Hip-Hop'), 3),
 (('Folk', 'Rock'), 3),
 (('Rock', 'Folk'), 2),
 (('Experimental', 'Hip-Hop'), 2),
 (('International', 'Rock'), 1),
 (('Pop', 'Hip-Hop'), 1),
 (('International', 'Instrumental'), 1),
 (('Electronic', 'Rock'), 1),
 (('Rock', 'International'), 1)]
```

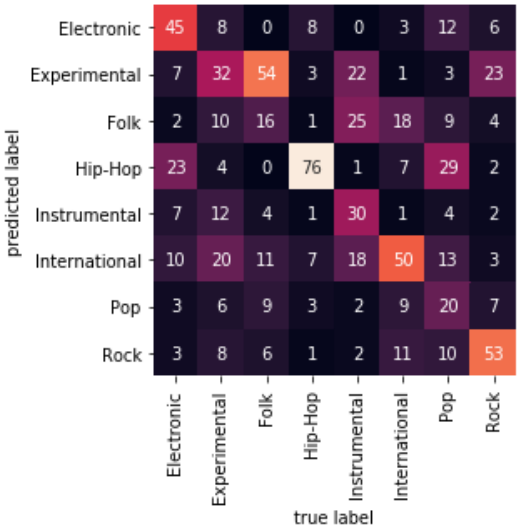
- **Class Wise Accuracy Score** -
  - Electronic - 59%
  - Experimental - 41%,
  - Folk - 10%
  - Hip Hop - 78%
  - Instrumental - 22%
  - International - 62%
  - Pop - 37%
  - Rock – 53%

- ***Overall Accuracy of the Model – 45.25%.***

- Model Loss, Accuracy and Confusion matrix of CNN Model.







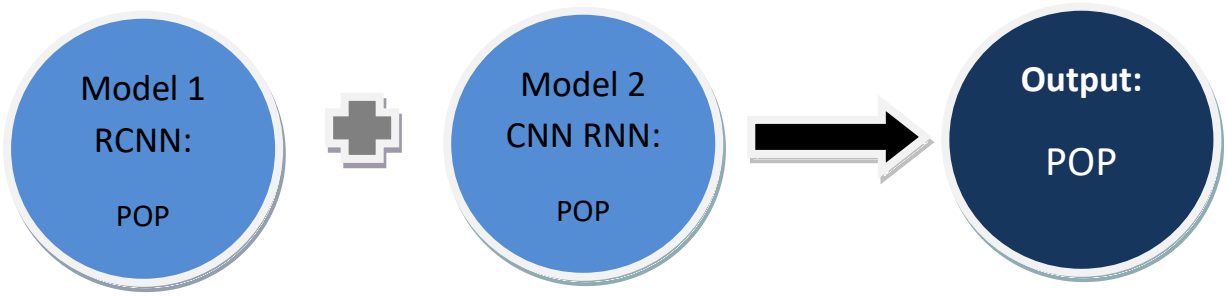
According to the confusion matrix:

Electronic music gets misclassified as Hip-Hop & Pop. Experimental Music gets misclassified as Instrumental & Pop.

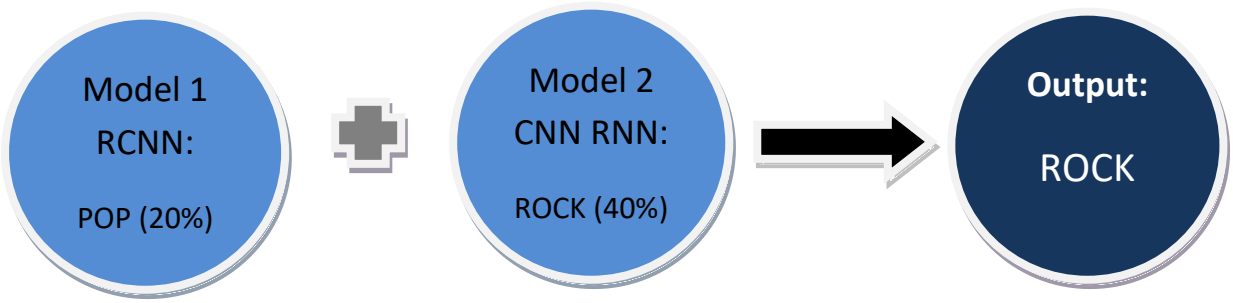
3.4 Heuristic Approach

- If two models are predicting that the song to be - ‘x’ genre. Our heuristic model predicts ‘x’. This is shown in Flow 1.

- But if both the models are predicting different genres then the model with the highest accuracy is taken into consideration. This is shown in Flow 2.



Flow 1

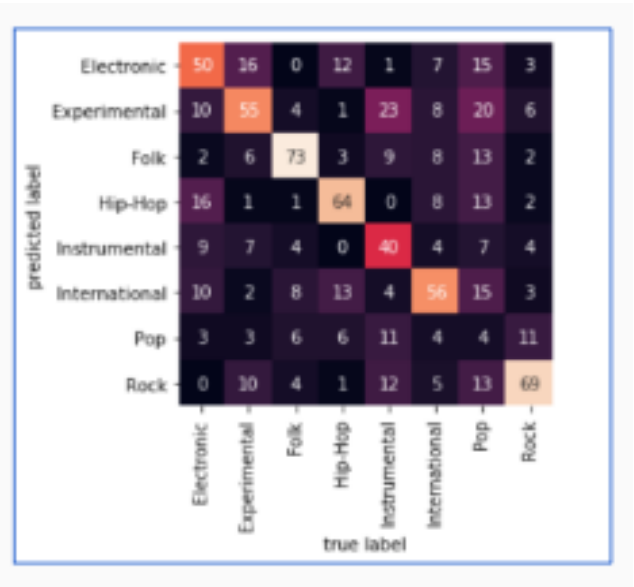


Flow 2

3.5 Calculating Overall Accuracy of Model

- **Class Wise Accuracy Score** -
  - Electronic - 50%
  - Experimental - 55%,
  - Folk - 73%
  - Hip Hop - 64%
  - Instrumental - 40%
  - International - 56%
  - Pop - 4%
  - Rock – 69%

- ***Overall Accuracy of the Model – 51.375%.***



Final Accuracy – 51.375 %

4. CONCLUSION

- We created a model which is used to detect the genre of a music file. The overall accuracy we could achieve from our model is 51.375%.

- Accuracy using two highest probabilities in the output (dense - SoftMax layer) is 70.625%.

REFERENCES

[1] Recurrent Neural Network Tutorial.

http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/

[2] RNN Effectiveness.

http://karpathy.github.io/2015/05/21/rnn-effectiveness/

[3] LSTM Networks

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[4] CNN for Visual Recognition.

https://cs231n.github.io/convolutional-networks/