**Final Report: Beat the Beats**

Report submitted to Fairfield University in partial fulfillment

of the requirements for the course of

IS 550: Business Analytics & Big Data Management

December 19, 2018

By

Moses Abishek Raj Pallapati

Course Instructor:

Dr. Jie Tao

Grade:

<To be filled by the instructor>

**Table of Contents**

0.Executive Summary

1. Introduction

1.1 Background

1.2 Methodology

1.2.1 Working of the Network

1.2.2 Working of the Algorithm

2. Analysis

2.1 Experiment Data

2.2 Data Processing

2.3 Modeling

2.3.1 Model Construction

2.3.2 Model Training

2.3.3 Model Testing

2.3.4 Model Evaluation

3. Results & Discussion

3.1 Interpretation

3.2 Comparison

4. Conclusion

5. Reference

6. Appendix A. Tables and Figures

**Executive Summary**

Being a musician myself, I always had difficulty selecting the songs which were musically challenging so that I could practice these songs. In our day to day life we listen to music which has been labeled under a genre and rating, however for a musician who wants to learn music would prefer having songs categorized based on the difficulty of these songs and also various factors which could be included in selection of such songs such as the time signature, tempo, beats overlapped etc.

The aim of this project would be creating such a model which could generate such

categories from a very large dataset of songs/audio tracks based on few factors using some deep learning methods. Hence the title was named as Beat the Beats, which means providing a sophisticated selection of songs tailored to a specific level of skill to a musician and through this generate some music for his/her practice sessions were the musician could beat the beats.

1. **Introduction** 
   1. **Background**

When I decided to work on the field of sound/audio processing I thought it that using

genre classification as a parallel to the image classification would be an ideal approach and would be easy. To my surprise I was faced with many challenges as there were not as many resources of audio classification when compared to the image classification using the neural networks. I was also not able to get the resources which would help me tackle specifically this problem. All I had was genre classification-based resources. One paper that did tackle this classification problem is Tao Feng’s paper [1] from the university of Illinois. So, after going through a lot of research papers which related to Image classification, signaling classification, Spectrograms. A very influential paper was Deep content-based music recommendation [2] This paper is about content-based music recommendation using deep learning techniques. The way they got the dataset, and the preprocessing they had done to the sound had really enlightened my implementation. Also, this paper was mentioned lately on “Spotify” blog [3].

**1.2 Methodology**

Deep learning has emerged as a new era in machine learning and is applied to several applications. The Deep learning algorithm namely CNN would be used in this project for image classification. The performance of this algorithm would be evaluated by the quality metric known as Mean Square Error (MSE) and classification accuracy.

**1.2.1 Working of the network**:

The working of the network is divided into two sections. Defined about the theory related to CNN in a brief manner. Section 2.2 deals with the properties of CNN and the function of layers.

Computational models of neural networks have been around for a long time, first model proposed was by McCulloch and Pitts as in []. Neural networks are made up of several layers with each later connected to the other layers forming the network. A feed-forward neural activation and the strength of the connections between each pair of neurons [4].

In FFNN, the neurons are connected in a direct way having clear start and stop place i.e., the input layer and the output layer. The later between these two layers, are called as the hidden layers. Learning occurs through adjustment of weights and the aim is to try and minimize error between the output obtained from the output later and the input that goes into the input layer. He weights are adjusted by process of back propagation (in which the partial derivative of the error with respect to the last layer of weights is calculated). The process of weight adjustment is repeated in a recursive manner until weight layer connected to input layer is updated.

Convolutional neural networks are designed to process two-dimensional(2-D) image. The network consists of three types of layers namely convolutional layer, sub sampling layer and the output layer.

**1.2.2 Working of the Algorithm**

The input to the network is a 2-D image. The network has input layer which takes the image as the input, output layer from where we get the trained output and the intermediate layers called as the hidden layers. As stated earlier, the network has a series of convolutional and sub-sampling layers. Together the layers produce an approximation of input image data. Neurons in layer say ‘n’ are connected to a local subset of neurons from the previous layer of (n-1), where the neurons of the (n-1) later have contiguous receptive fields.

Convolutional layer

The convolutional layer is the first layer of the CNN network. The structure of this later is shown in the figure (1). It consists of a convolutional mask, bias term and a function expression. All these together generate the output of the layer. The bias term is also known as the error term. The bias layer is then added along with the sigmoid function to the matrix.

A picture containing building

Description automatically generated

Figure 1.1 Layers in convolutional neural network

Sub sampling layer

The sub sampling layer comes after the convolutional later. It has same number of planes as the convolutional later. It reduces the size of the feature map. It then divides the image into blocks of 2x2 and performs averaging. This preserves the information between features and removes the relation.

Throughout this project we would be using the basic methodologies followed in audio analysis using Image classification. Then we would develop a Convolutional neural network for model and then generate the results. The steps for the process are as follows

1. Reading the audio files
2. Calculating the Bpm/tempo of the audio files to classify them
3. Generating Spectrograms
4. Slicing the Spectrograms for Image classification
5. Labelling the Images
6. Developing the model- Convolutional neural network
7. Generating results- Accuracy metric, error term

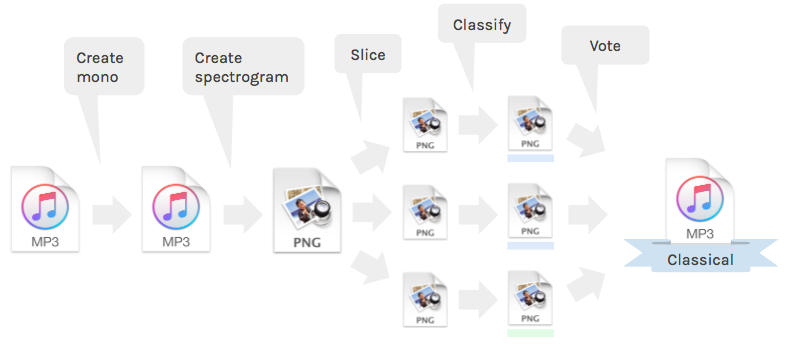


Figure 1. Diagrammatic representation of the process before using CNN

**2.Analysis**

**2.1 Experimental Data:**

The dataset used in this project was obtained from FMA (Free Music Archive), an interactive library of high quality, legal downloads. This dataset is an open and easily accessible dataset suitable for evaluating MIR, a field concerned with browsing, searching and organizing large music collections [5]. The original dataset consists of 106,574 tracks from 16,341 artists and 14,854 albums altogether. The size of this dataset is 917 Gi Bytes. However, for convenience

Sake we have only used the smaller dataset provided from the same archive. The smaller dataset used consists of 8000 tracks of 30seconds each, from across 8 balanced genres and total file size of 7.12 Gi Bytes. The file was downloaded in the form of a zip file and later extracted. The metadata file consisted of 7 csv files which consisted of data such as genres, features, albums, artist, tracks and echo nest features. The dataset consisted of 125 subfolders in which all the 8000 tracks were organized and numbered.

**2.2 Data Processing**

Initially using the metadata files the important metadata was consolidated into a single .csv file and extracted for all the mentioned 8000 audio tracks. Using the feature tempo and energy level from the features metadata, we could extract and classify the audio tracks. The classification method used was kept simple as the dataset was large. The classification function was using the tempo feature. If the tempo was over 160 then the label would be ‘Difficult’ and the tempo between 80 and 160 was labelled as ‘Medium’ and the tempo below 80 as ‘Easy’. For this project only, the tempo was considered for convenience. All the data was formulated in a csv file using pandas where the track id, was used as the identifier or the key to the classifier.

For processing the data, the following steps had to be performed.

1. Reading of the mp3 audio file
2. Converting the mp3 file to a .wav file
3. Segmenting or dividing the audio file into smaller audio files for data augmentation and clearer spectrogram image to recognize patterns
4. Generating the spectrograms form these smaller audio files
5. Saving these spectrograms images into their respective classification folders.

**2.3 Modelling the neural network:**

For this project the high-level neural network API- Keras was used and implemented in Python notebook. However, for the backend the software library TensorFlow (Google) was used.

This Modelling phase consisted of four important tasks

2.3.1 Model Construction

* + 1. Model Training
    2. Model Testing
    3. Model Evaluation

**2.3.1 Model Construction**:

For model construction, we used the neural networks. We would initially initiate the model sequence by using the object: model = sequential (). Once this is done, we then add the layers according to their type of layer. In this stage we would also select a loss function and an optimizer while the model is being constructed by TensorFlow to analyze the model. The loss function written, would show the accuracy of the prediction made by the model.

The activation function ‘Relu’ was used to set the zero threshold and remove the unnecessary details. Max pooling 2D layer was used as this was spatial data. A Dropout layer was also used to prevent overfitting in the model and the dropout value to set to 0.5. Model was then complied to a binary cross entropy loss function which would return sum of all individual losses. The optimizer algorithm ‘RMSprop’ was used for as deemed best for recurring neural networks. The modelling was observed based on the accuracy metrics to evaluate the performance of the model.

**2.3.2 Model Training:**

In this phase, the model data is fitted to the appropriate label as the expected output and the training is done. The format followed was model fit (training data, expected output), where the expected output was the label generated using the classification and the input training data being the spectrograms.

**2.3.3 Model Testing:**

In this phase the model is imputed with the testing dataset which the model has never seen. This would generate the true accuracy of the model once the model is tested.

**2.3.4 Model Evaluation:**

In this phase, after the model has been tested, the model now can be used to evaluate new data.

**3.Results**

As the code written could not be executed (errors while mapping label to the images) the results could not be evaluated. However, the result expected was the accuracy of the prediction to the classification of the audio track and see if the accuracy is greater than the 80% depending on the saved epoch values. Given 50 epoch values would have given various accuracy values, which would have been plotted against the test data and checked if any overfitting was significant in the curve of accuracy. The validation data accuracy and the accuracy of the test data would have given us a clearer picture with regards with the overfitting of the model and the optimum number to use for the epochs to train the model.

**4. Conclusion:**

In conclusion the data was taken from the FMA dataset, consisting of the musical tracks of various genres. This data was used to classify the level of difficulty in playing such music track. The classification was done in three categories labelled as ‘Difficult’, ‘Medium’, ‘Easy’. This labelling was done based on the tempo of the audio track manually. These audio tracks then were further divided into smaller chunks of audio of 3 seconds each (initial being 30 seconds tracks). These smaller tracks were then used to generate the spectrograms (Images) for the Image classification using the neural networks. The Convolutional Neural networks was used as the working algorithm. The model constructed was a recurring neural network which based its performance on the loss function which gave the output as accuracy. The accuracy was the measure the prediction of the model. Unfortunately, the written code was not executed and had multiple compiling errors especially with regards to the mapping of the labels to the relevant images. The process was complex as the images were generated from an audio track which was divided into 10 segments of 3 seconds each and all the images had to be matched to the single-track Id and labelling of the images.

The expected output however would be the accuracy of the prediction on the test data, which could help us evaluate the model and enable to enhance the model based on the optimization method and the loss function.

**5. References:**

[1] Lillesand, T.M. and Kiefer, R.W. and Chipman, J.W., in “Remote Sensing and Image Interpretation” 5th ed. Wiley, 2004

[2] Li Deng and Dong Yu “Deep Learning: methods and applications” by Microsoft research [Online] available at: <http://research.microsoft.com/pubs/209355/NOW-Book-RevisedFeb2014-online.pdf>

[3] McCulloch, Warren; Walter Pitts, "A Logical Calculus of Ideas Immanent in Nervous Activity”, Bulletin of Mathematical Biophysics 5 (4): 115–133(1943)

[4] An introduction to convolutional neural networks [Online]available at: http://white.stanford.edu/teach/index.php/An\_Introduction\_to \_Convolutional\_Neural\_Networks

[5] “FMA: A Dataset For Music Analysis.” [Astro-Ph/0005112] A Determination of the Hubble Constant from Cepheid Distances and a Model of the Local Peculiar Velocity Field, American Physical Society, 5 Sept. 2017, arxiv.org/abs/1612.01840.

**6. Appendix A: Tables and Figures**

Figure 1: snapshot of the metadata file (attached separately is the notebook)

**A screenshot of a computer

Description automatically generated**

Figure 2: snapshot of the spectrogram generated (attached separately is the notebook)

A picture containing object

Description automatically generated

Figure 3: snapshot of the features used for classification (attached separately is the notebook)

A screenshot of a cell phone

Description automatically generated

Figure 4: snapshot of the labelled data (attached separately is the notebook)

A screenshot of a cell phone

Description automatically generated

Figure 5: snapshot of slicing of the audio file (attached separately is the notebook)

A screenshot of a cell phone

Description automatically generated

References : Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, Xavier Bresson. FMA: A Dataset For Music Analysis. [[Web Link]](https://arxiv.org/abs/1612.01840), 2017.