INTERNATIONAL JOURNAL ON HUMAN COMPUTING STUDIES

https://journals.researchparks.org/index.php/IJHCS e-ISSN: 2615-8159 | p-ISSN: 2615-1898

Volume: 03 Issue: 6 |Aug 2021

Music Genre Classification Using 1D Convolution Neural Network

Peace Busola Falola, Solomon Olalekan Akinola

Department of Computer Science, Faculty of Science, University of Ibadan, Ibadan, Nigeria

---***---

Abstract: Music genre classification system is a system that is important to the users for effectiveness in the digital music industry. One of the effective ways of genre classification is in music recommendation and access to users. With accurate classification system built, songs can be readily accessed by the users when the genre of the song is known and recommendation of songs to the users is made easy. Also, automatic classification of genre is important to solve problems such as tracking down related songs, discovering societies that will like specific songs and also for survey purposes.

In recent times, deep learning techniques have proven to be effective in several classification tasks including music genre classification. This paper therefore examines the application of 1D Convolutional Neural Network for music genre classification. A new dataset consisting of 1000 Nigerian traditional songs with seven genres was used for this work. As features extraction is crucial to audio analysis, seven low level features also known as content based features were extracted from the songs in the dataset which served as input into the classifier. Our results showed that the accuracy level of the system is 92.5% with a precision of 92.7%, recall of 92.5% and f1 score of 92.5%.

Keywords: Feature extraction; Low level features; Content based features; 1D Convolutional Neural Network; Deep learning; Classification.

1. Introduction

Music genre classification is an aspect broadly studied in Music Information Research (MIR) community. This is as a result of the high volume of digital music available over the internet. This has led researchers to develop music retrieval techniques that would be helpful for internet music search engines and listeners to find music from numerous options. Automatic classification of music into their genre is one of the techniques that is widely examined in MIR. Music genre is an efficient method for structuring, organizing and easy retrieval of the high volume of music files available over the internet.

This research paper therefore focuses on the classification of music into their genres. A new dataset was used for this work as against the popular dataset such as GTZAN, Million Song, Free Music Archive, MIDI. The new dataset consists of 1000 Nigerian traditional songs with four genres classes (Afro, Apala, Fuji and Juju). Each song has a duration of 30 seconds.

For an automatic genre classification system, three steps are usually involved: (1) Extraction of features such as timbre, spectro-temporal and statistical features are extracted from the audio signal [1]. Other features such as name of artist, cover album and many more can also be extracted as features.(2) Some techniques are then applied to select meaningful subset of the features [1,2] or aggregate features [1,3,4] to improve the classification accuracy. This is mostly termed pre-processing. (3) A classifier based on machine or deep learning methods is then trained over the selected features from (2) to classify the input music automatically into their various genres.

For the extraction of features, seven features were extracted from the compiled audio files which served as the input into the classifier. Feature extraction is an important stage in audio analysis as an efficient classification system is determined by extraction of good features and a good classification system [5].The model cannot understand raw data, hence the need to extract features from the raw data that adequately represent the data. Seven content based features which are embedded in the music were therefore extracted for this work.

1D Convolutional Neural Network was employed as the classifier as the content based features used in this work are in a one-dimensional format. Studies over the years have shown that 1D CNN with fairly shallow architecture which means smaller number of layers and neurons are able to learn challenging tasks involving 1D signal [6]. Other importance of 1D CNN includes easy training and implementation because it is a classifier with shallow architecture [6]. Also, CPU implementation over a standard computer is relatively fast for training 1D CNNs with few hidden layers [6]. 1D Convolutional Neural Network also has the advantage of low computational requirements which are well suited for real-time and low cost applications [6].

Generally, 1D CNN works well for analysis of a time-series of sensor data, analysis of signal data over a fixed-length period such as audio recording and also for natural language processing [7].

In previous works, spectrogram (a visual representation of the audio signal – 2D data) has been an excellent feature which has given excellent results with 2D Convolutional Neural Network for genre classification [8, 9, 10]. However, content based features which are one-dimensional data used as input into various machine learning and deep learning classifiers need more improvement to deliver an excellent accuracy result [11,12, 10].This work therefore focuses on using 1D Convolutional Neural Network to classify Nigerian traditional songs to genres with seven content based features as the input. This work seeks to verify whether 1D Convolutional Neural Network is effective to classify music into their genres using the accuracy evaluation metrics and some others metrics.

2. Related Works

Elbir and Aydin [8] implemented a music genre classifier and recommendation system based on signal processing and a CNN model named MusicRecNet as against existing systems that were only classification systems. The system built was also capable of checking plagiarism of songs. GTZAN dataset was used for this work. Melspectrogram was generated from each of the song in the dataset and was saved as an image. The images generated served as the input and were applied to the MusicRecNet for training. After the training, the model was used for genre classification. Also, the dense 2 layer of the MusicRecNet was used as a feature vector of the test music samples which were further fed into various classifiers such as MLP, Logistic regression, random forest, LDA, KNN and SVM for music genre classification, music similarity and music recommendation. Accuracy was the main performance metric that was used. Also, the average percentage of music similarity was also used as a metric for the quality of music recommendation. MusicRecNet as a standalone classifier gave a mean accuracy of 81.8% which performed better than the results of other studies. For the application of the dense 2 layer of MusicRecNet as feature vector to the test music samples, MusicRecNet with SVM gave an accuracy of 97.6% for the music genre classification, music similarity and music recommendation. The proposed MusicRecNet model showed improved performance in terms of music genre classification, music similarity and music recommendation as compared to previous studies.

Ghosal and Sarkar [9] proposed a novel approach for automatic music-genre classification system using a deep learning model with GTZAN as the dataset. The model leveraged on Convolutional Neural Nets (CNN) to extract spectrogram and the output was fed into LSTM sequence to sequence auto encoders. After a complete training, the activation of the fully connected encoded layer was used as representations of the audio sequence and was fed as input to Clustering Augmented Learning Method Classifier (the novel approach). Clustering Augmented Learning Method (CALM) classifier was based on the concept of simultaneous heterogenous clustering and classification to learn deep feature representations of the features obtained from LSTM autoencoder.

The performance of the model was evaluated using precision, recall and accuracy. Confusion matrix was plotted and the proposed model classified 80% of rock audio as rock correctly and labelled others mainly as country or blues. It incorrectly classified some country and a small fraction of blues and reggae as rock music.

Four traditional classification models were trained on the dataset as baseline classifiers which were k-nearest neighbours, logistic regression, random forest, multilayer perceptrons and linear support vector machine, using Mel Frequency Cepstral Coefficients (MFCC) by flattening them into a 1-D array. Also, the features obtained from Convoutional Net and LSTM Autoencoder was stacked on a Logistic Regression classifier to test the performance of the CALM classifier. CALM outperformed all the models with an accuracy of 95.4%.

Pelchat and Gelowitz [13] worked on 1880 songs with seven genres as the dataset. The duration of each song was three minutes. The dataset was processed by transforming the stereo channels into one mono channel, and SoundXchange command-line music application utility was used to convert the music data into a spectrogram. The songs were further divided into 2.56 seconds to approximately make 132,000 labelled spectrogram snippets which were the inputs into the classifier. Convolutional neural network was the neural network used for classification with Rectified Linear Unit (ReLu) activation function which gave a test accuracy of 67%.

Chillara, Kavitha, Shwetha,Neginhal, Haldia and Vidyullatha [10] in their work built multiple classification models (spectrogram-based models and feature based models) and trained them over the Free Music Archive (FMA) dataset. The performance of these models was compared and the results were based on the prediction of the accuracy. The features used were mel-spectrogram and time and frequency domain features. Convolutional Neural Network (CNN), Convolutional Recurrent Neural Network (CRNN) and Convolutional Neural Network plus Recurrent Neural Network (CNN-RNN) models were trained with mel-spectrogram which was the spectrogram-based models. While the time and frequency domain features were trained with Logistic Regression and Simple Artificial Neural Network models which were the feature based models.

One of the spectrogram based models gave the highest accuracy of 88.54% which was the Convolutional Neural Network model trained with the spectrogram feature.

Bahuleyan [11] in his work on “Music Genre Classification using Machine Learning Techniques” compared the performances of two models. The first approach was Convolutional Neural Network (CNN) model, which was trained end-to-end to predict the label of an audio signal using the spectrogram feature. Four traditional machine learning classifiers (Logistic Regression, Random Forest, Gradient Boosting, Support Vector Machines) using hand-crafted features both from the time and frequency domains was the second approach used. An ensemble combining the two approaches was also used. The experiment was conducted on a dataset called Audio set. The evaluation metrics used were accuracy, f-score and AUC. The CNN based approach outperformed the feature-engineering based models (time and frequency domain features) with an accuracy score of 89.1%. The ensemble proved to be beneficial and gave an accuracy score of 89.4%.However, there was recommendation to improve the classification by preprocessing the noisy data from the audio clips (dataset) which were gotten from YouTube videos before feeding it into the machine learning model.

Vishnupriya and Meenakshi [12] worked on classifying Million Song Dataset (MSD) into different genres. The dataset had 1000 songs with 10 genres. The feature vector extraction was done using the librosa package in python. The package is specifically used for audio analysis. The extracted feature vector was Mel-frequency Coefficient (MFCC). MFCC encode the timbral properties of the music signal by encoding the rough shape of the log-power spectrum on the Mel-frequency scale. Two types of feature vectors were obtained: Mel Spectrum with128 coefficients and another is MFCC with 13 coefficients. The feature vectors obtained was stored into a database. The data was then shuffled for a good form of generalization before it was fed into the convolution neural network. 800 song features were taken for the training of the model while the remaining 200 were for testing.

After training the model, the learning accuracy for Mel Spec feature vector and MFCC feature vector were 76% and 47% respectively. MFCC took less time for converging whereas Mel Spec was more time consuming for learning.

Valerio, Pereira, Costa, Bertolini and Silla Jr. [14] presented that in Music Information Retrieval field, songs datasets are usually very unbalanced. In lieu of this, they proposed a novel approach to face the class imbalance problem applied to music genre classification. The approach used vertical sliced spectrograms extracted from the songs' audio signal to apply oversampling and undersampling into the minority and majority classes, respectively. F-Score was the evaluation metric and this showed that the approach was able to beat the best result of Random Undersampling technique, using MultiLayer Perceptrons. Comparison of the result to the baseline results showed that the approach significantly increased the individual results for all the minority classes.As future work, the use of other visual features to generate the feature vectors from the spectrograms was suggested. The use of fusion approaches (early fusion and late fusion) to combine the visual-base features with audio-base features in order to improve the results was proposed. A development of a new strategy that combines different musical pieces in order to generate synthetic samples for the minority classes was also proposed.

With much focus on spectrogram feature with 2D Convolutional Neural Network as the classifier over the years for music genre classification, this paper focuses on content based features and 1D Convolutional Neural Network for music genre classification to achieve an excellent accuracy result.

3. Methodology

Figure 1 shows the developed methodology framework for this study, which contains two major phases: feature extraction and classification. These two phases can further be broken down into six stages:

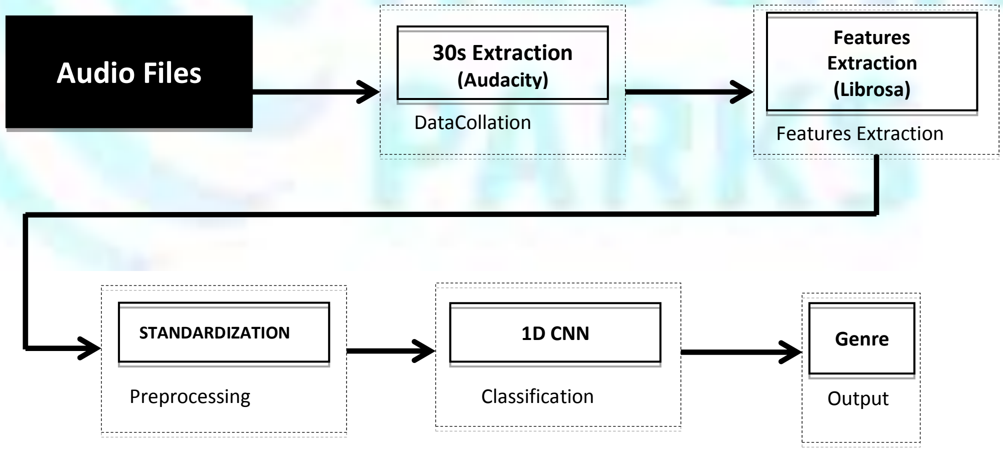


Figure 1 Methodology Framework

Stage One: This stage entailed the gathering of the audio files from which the features that were used as our dataset were extracted from.

Stage Two: Thirty seconds of the beginning, middle and end of each of the song were extracted from the audio files gathered in stage one.

Stage Three: The low level features were extracted from the audio files gathered in stage two and were stored in a CSV file.

Stage Four: Preprocessing of the dataset was achieved at this stage.

Stage Five: The training and testing of the model with the dataset was achieved at this stage.

Stage Six: The accuracy of the model was gotten at this stage to prove how well the model classified the songs accurately.

3.1 Source and Nature of Data Used

The raw audio files were Nigerian songs and they were gotten from music sales vendors. After preprocessing the audio files, the nature of the data was both numerical and categorical. The features extracted were numerical while the targets/ labels were categorical in nature. Our dataset contained one thousand songs with seven features each. The labels were four which are Afro, Apala, Fuji and Juju.

3.2 Features Extraction

The process of computing a compact or concise numerical representation that is used to typify a fragment of audio is referred to as feature extraction [15]. The aim of feature extraction is to represent a music piece or fragment into a compact and descriptive way [16]. Suitable machine learning algorithms or deep learning algorithms are then used to classify the audio signals into the desirable outputs (such as genre) using the significant features extracted. Feature extraction is the foremost procedure of pattern recognition systems [17].

Content- based features and text-based features are the two major features that can be extracted from the music audio signals. The content-based features involve the extraction of features that are embedded in the audio file that describes the music audio signal and these features are used to classify music into their genres or identify important information about the music. The content-based features consist of the low-level features and high-level features. Low-level feature is the numerical values describing the contents of a signal according to different kinds of inspection: temporal, spectral, perceptual, e.t.c. [18]. These features do not make any sane meaning to humans. Low-level features are obtained directly from various 13 signal processing techniques like fourier transform, spectral/cepstral analysis, autoregressive modeling, e.t.c. [19].

There are 3 main typical features of characterizing music content. They are Timbre, Melody/Harmony, and Rhythm.

3.2.1 Timbre Features

Timbre features describe sound on a fine scale (they are typically computed for segments of signal of 10–60s) [17]. They can also be referred to as tonal qualities that define a particular sound/source [20]. These features have a little or no meaning to the users but are well utilized by the computer systems [21]. The calculated features are based on the short time Fourier transform (STFT) and are calculated for every short-time frame of sound [15]. These features are also referred to as low-level features, time and frequency domain features and short term features. The time domain features are determined by summing up the absolute or squared values of the amplitudes [22]. The frequency domain features are based on a preliminary Fast Fourier Transformation (FFT) which is used to obtain a representation of the audio signal in the frequency domain. Such approaches assume that the signal is periodical [22, 23]. 14 Standard features proposed for music-speech discrimination and speech recognition are used to represent timbral texture. The calculated features are based on the short time Fourier transform (STFT) and are calculated for every short-time frame of sound [15]. Some of the timbre features used in genre classification are as follow:

A. Temporal features: These features are computed from the audio signal frame (zero crossing rate and linear prediction coefficients) [17].

B. Energy features: This refers to the energy content of the signal (root mean square energy of the signal frame, chroma_stft, energy of the harmonic component of the power spectrum, and energy of the noisy part of the power spectrum) [17].

C. Spectral shape features: This describes the shape of the power spectrum of a signal frame. (Spectral centroid, spread, skewness, kurtosis, slope, spectral bandwidth, Spectral rolloff, Mel-frequency cepstral coefficients (MFCCs) [17].

D. Perceptual features: This is computed using a model of the human earring process (relative specific loudness, sharpness, and spread) [17].

The timbre features extracted for this project are:

- **Chroma_stft:** Music audio has 12 distinct semitones of the musical octave [23, 24]. The chroma features correlate to the total energy of the signal in each of the 12 distinct semitones or pitches. Chroma feature gives the pitch class. The chroma vectors are then combined across the frames to derive a representative mean and standard deviation [10].

- **Spectral_Centroid:** This measures at which frequency the energy of a spectrum is centered upon [25].

$$f_n = \frac{\sum_k S(\text{ꝁ}) f(\text{ꝁ})}{\sum_k f \text{ꝁ}}$$

- **Spectral_Bandwidth:** The p-th order spectral band-width corresponds to the p-th order moment about the spectral centroid [26]. This is similar to a weighted mean:

$$(\sum_{\mathrm{k}} \mathrm{S}(\text{ꝁ})(\mathrm{f}(\text{ꝁ}) - f_{\mathrm{c}})^p)^{\frac{1}{p}}$$

- **Spectral_Rolloff:** This correlates to the frequency below, which is a percentage of the total spectral energy lies. The average and standard deviation of the spectral roll off is calculated across all frames of the audio signal[26].

- **Root Mean Square Energy:** The energy of a signal correlates to the total magnitude of the signal [27]. For audio signals, it corresponds to how loud the signal is [15]. RMSE is calculated frame by frame. The average and standard deviation is then calculated across all the frames of the audio signal [11].

The energy present in a signal is calculated as follows [25].

$$\sum_{n=1}^{N} |x(n)|^2$$

The root mean square energy (RMSE) in a signal is calculated as:

$$\sqrt{\frac{1}{N} \sum_n |x(n)|^2}$$

- **Zero_Crossing Rate:** A zero crossing point (ZCR) is when the audio signal changes sign from positive to negative [27]. ZCR depicts the number of times the waveform crosses 0. For highly percussive sounds, ZCR has higher values. The number of zero-crossings present in each frame after the division of the 30 second audio signal is calculated [11]. The average and standard deviation of the ZCR across all frames are chosen as representative features [11].

- **Mel-Frequency Cepstral Coefficients:** MFCC is a useful feature for characterizing music timbre [28]. The MFCC of a signal are a small set of features (between10-20) which compactly represent the overall shape of a spectral envelope [29]. MFCC represents a set of short term power spectrum characteristics of the sound [30].

3.2.2 Melody/Harmony

In music, the study of pitch concurrency and chords whether actual or implied is referred to as harmony. On the contrary, the chronological sequence of pitched events recognized as a single entity is melody. Harmony is also referred to as the vertical element of music and melody referred to as the horizontal element [17]. Melodic and harmonic content are better defined by lower-level attributes than notes or chords. Harmonic and melodic content have been used more intensively in the context of semantic segmentation and summarization of music [31]. The fundamental idea is to use a function that describes pitch distribution of a short segment like most melody/harmony analyzers; the difference is that no decision on the fundamental frequency, chord, key or other high-level feature is undertaken [17]. On the contrary, a set of descriptors such as amplitude and positions of its main peaks, interval between peaks, sum of the detection function are computed from this function and possibly any kind of statistical descriptor of the distribution of the pitch content function [17]. Two versions of the pitch function are typically used: an unfolded version that contains information about the pitch range of the piece and a folded one, in which all pitches are mapped to a single octave giving a good description of the harmonic content of the piece [17].

3.2.3 Rhythm

The pattern of pulses/notes of varying strength is referred to as rhythm. Terms such as tempo, meter or phrasing are used in describing rhythm [32]. Five features are proposed to describe the rhythmic property of music: rhythm strength, rhythm regularity, rhythm clarity, average onset frequency, and average tempo [33].

- Rhythm strength is the average onset strength in the onset detection curve [34].
- Rhythm regularity and clarity are computed by performing autocorrelation on the onset detection curve. If a music segment has an obvious and regular rhythm, the peaks of the corresponding autocorrelation curve will be obvious and strong as well [32].
- Onset frequency, or event density, is calculated as the number of note onsets per second [32].
- Tempo is estimated by detecting periodicity from the onset detection curve [35].

Music genre classification requires extraction of efficient features from the audio files for correct classification. Librosa, a python library was used to extract the features. Feature extraction is an important stage in audio analysis. The aim of this phase is to extract a set of features from the compiled audio files which serves as our dataset. Content based features consisting of low level features were extracted from each of the audio file. Seven features in total were extracted from the 1000 audio files to make up the dataset for this project and they were saved in a csv file.

3.3 Data Preprocessing

The preprocessing phase is an important phase where raw formats of data are transformed or encoded into understandable formats the model can easily work with. Several reasons such as noisy data, many dimensions, missing data contribute to the need for data preprocessing.

Feature scaling and feature encoding were the two data preprocessing techniques that were used to preprocess the dataset.

3.3.1 Feature Scaling

Feature scaling, also referred to as data normalization, was performed during the data pre-processing where range, magnitudes and units of variables were normalized so that they are compared on common ground. Machine learning algorithms weigh greater values higher than smaller values irrespective of the units of the values if feature scaling is not performed on the data [36]. Feature scaling is therefore important in order to transform data into a standard scale.

The min-max scaling method was used in this research to scale the features. Min-max scaling technique scales the feature value to a fixed range between 0 and 1. This results in smaller standard deviations and suppresses the effect of outliers [37]. This is very much useful for features with hard boundaries.

$$x' = \frac{x - \min}{\max(x) - \min(x)} \qquad (1)$$

3.3.2 Feature Encoding

Feature encoding transforms data into an easily acceptable input for machine learning algorithms while still retaining its original meaning. Some of the features to encode are the categorical features. The categorical features will not be recognized by the machine except it is converted to numerical values. The labels (genres), which are the final outputs that were used to train the model alongside the features, are not numerical values. Therefore, they were encoded for the deep learning model to understand. Four labels/ genres were in the dataset. These are Apala, Fuji, Juju and Afro. After encoding these labels, a numerical value was assigned to each of them starting from zero in an alphabetical order of the names of the genres. The encoding techniques used were one hot encoding and label encoding.

Label encoding was used to encode the target labels from categorical to numerical. After this, one hot encoding was performed on the encoded target labels for easier modelling. It was possible to use one hot encoding because we had few categories in our dataset.

3.4 Data Exploratory

Exploration of data entails knowing the correlation or relationship between the data. Correlation is a statistical measure to assess relationships among variables [38].

A significance test is a great test for ascertaining the relationship between the data. This is because correlation alone can be misleading if working with sample (r) data because an existing correlation in a sample doesn't automatically mean it is present in the population (ρ) from which the sample came from [38]. This proves why a significance test should be done after correlation to confirm the reliability.

Statistical significance test was carried out on our data to know what relationship existed in the data and a hypothesis test was carried out to achieve this [38]. Hypothesis test is an essential part of statistical inference. The statistical inference makes inferences on the population based on a sample of the population.

The hypotheses are as follows [38].

$$\rho = 0$$

$$\rho \neq 0$$

When $\rho = 0$, this simply means there is no correlation among the two variables compared and when $\rho \neq 0$, it shows there is an amount of correlation between the two variables compared in the data, this means a linear correlation exists between the variables but it doesn't imply that one affects the other [39].

When $\rho = 1$, it shows there is a strong correlation. This often happens when a variable is compared to itself. It shows that a variable will always correlate with itself. Also, values close to 1 are said to have a good correlation as well, that is, a linear relationship exists between the variables [38]. Values close to 0 are said to have little or no correlation and values below 0 are said to have a non-linear relationship.

3.5 Classification Tool Employed

The classifier used to classify the songs into their genres for this project was 1D Convolutional Neural Network (1D CNN).

3.5.1 1D Convolutional Neural Network (1D CNN)

Convolutional Neural Network is one of the most popular algorithms for deep learning, a type of machine learning in which a model learns to perform classification tasks directly from images, video, text and sound [40]. It is used for identifying simple patterns which is later used to form complex patterns within higher layers. A 1D CNN is effective in deriving features from shorter (fixed-length) segments of the overall data set. It is also effective where the location of the feature within the segment is not of high relevance [40].

1D CNN consists of two distinct layers [6]

1. CNN layers where both 1D convolutions, activation functions and sub-sampling (pooling) occur
2. Fully-connected (dense) layers that are identical to the layers of a typical Multi-layer Perceptron

The configuration of a 1D-CNN is formed by the following hyperparameters [6]:

1. Number of hidden CNN and MLP layers/neurons (3 and 2 hidden CNN and MLP layers, respectively).
2. Filter (kernel) size in each CNN layer
3. Sub-sampling factor in each CNN layer
4. The choice of pooling and activation functions.

CNN is basically broken down to the following operations:

3.5.1.1 Convolution Layer

This is the core building block of CNN. This layer's parameters consist of a set of learnable filters also called kernels [41]. A convolutional layer simply transforms the input data in order to extract features from it. The convolution slides the kernel over the input data, a procedure referred to as shift-compute procedure [42]. This happens in two ways: Non casual convolution and casual convolution.

In non-casual convolution, the output is dependent on the future input while in casual convolution, the output is not dependent on future inputs [42].

3.5.1.2 Pooling Layer

The dimensionality of a given mapping that is the number of parameters is reduced while the prominent features are being highlighted [42]. Pooling is simply employed to reduce the dimension of the convolution output, which reduces in return reduces the computational cost [43]. This layer also helps to avoid over fitting. The max pooling technique is the most common technique and it works by selecting the maximum value in each patch of each feature map [44].

3.5.1.3 Flatten Layer

This layer converts the output from the convolutional layers to form a flat structure or a 1-dimensional array [42].This is fed into the Multilayer Feed Forward Network, also called fully connected layer.

3.5.1.4 Multilayer Feed Forward Network (MLFF)

The MLFF or fully connected layer is an interconnection of perceptron in which data flows from the input data to the output. It is simply a structure where the neurons from a previous layer are connected to all the neurons in the next layer [42].

3.6 Activation Functions

Activation function transforms input into outputs of a different kind. The common activation functions properties are monotonic, continuous, differentiable, range and non-linear. The common activation functions used in this project were: Rectified Linear Unit (ReLU) and softmax activation functions.

The rectified linear unit is one of the simplest activation functions. It is used in the hidden layers. ReLU is non-linear and it doesn't have any back propagation errors. It also avoids and rectifies the vanishing gradient problem. The softmax activation function is used in the output layer for a multiclass classification and such networks are usually trained under a log loss or cross entropy regime.

Neural networks for classification that use sigmoid or softmax activation function in the output layer learn faster and more robustly using a cross-entropy loss function [45].

3.7 Dropouts

Dropouts prevents over fitting which memorizes the inputs in contrast to learning general traits of the inputs [42]. Dropouts simply drop the output of a neuron which implies zero input to the next layer. The dropout rate determine s whether a neuron is dropped or not [42].

3.8 Loss Function

Loss function has to do with the derivation of the predicted output from the target output. For a classification problem, the most common loss function used is the cross entropy, this is also called Maximum Likelihood Estimation [42]. The goal of every classification problem is to reduce the loss function and Gradient descent optimization can be used to reduce the loss function [42].

3.9 Optimizer

The Adaptive Moment Estimation (Adam) optimizer is the optimizer that was engaged in this project. Adam computes adaptive learning rates for each parameter [46]. It stores exponentially decaying average of past squared gradients like adaptive gradient algorithm (AdaGrad), root mean square propagation (RMSprop) and it also keeps an exponentially decaying average of past gradients similar to momentum [46]. Momentum can be likened to a ball running down a slope. Adam behaves like a heavy ball with friction which prefers flat minima in the error surface [46].

3.10 Epochs

Epoch is a hyperparameter in deep learning which is defined before training a model. An epoch equals to an iteration over the full dataset or it is a measure of the number of times all the training vectors are used once to update the weights [47]. One epoch is when an entire dataset is passed both forward and backward through the neural network once [47]. Since one epoch is too big to feed to the computer at once, they are divided into several smaller batches.

3.11 Performance Metrics

The evaluation metrics for the model built are: Accuracy, Precision, Recall and F1-score.

- Accuracy: This gives the ratio between the correctly predicted outcome and the total sum of all predictions [40].

$$Accuracy = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

- Precision: This metric confirms whether when the model predicted positive, it was right or wrong [40].

$$Precision = \frac{\text{Number of true positives}}{\text{Number of positive predictions}}$$

Recall: This metric confirms how many positives the model identified out of all possible positives [40].

$$Recall = \frac{\text{Number of true positives}}{\text{Number of actual positives}}$$

- F1- score: This gives the weighted average of precision and recall [40].

$$F1\ score = \frac{\text{2 x recall x precision}}{\text{recall} + \text{precision}}$$

- Training and Validation Loss: A loss function is employed to optimize a deep learning algorithm. The loss is usually calculated on training and validation sets. The interpretation of the loss function is often determined by how excellent the model is doing on both sets [48]. Loss value implies how poorly or well a model behaves after each iteration of optimization. A good model gives low validation loss and high validation accuracy [48].

4 Implementation

4.1 Data Collation

30 seconds was extracted from the beginning, middle and end of each audio file (Nigerian traditional songs) gathered to be worked upon for this project. Audacity software was used to achieve this. 1000 files were gotten in all. These consisted of four genres. The genres/labels are Afro, Apala, Fuji and Juju. Each genre has 250 songs each. Figure 2 shows the interface of the audacity software that was used.

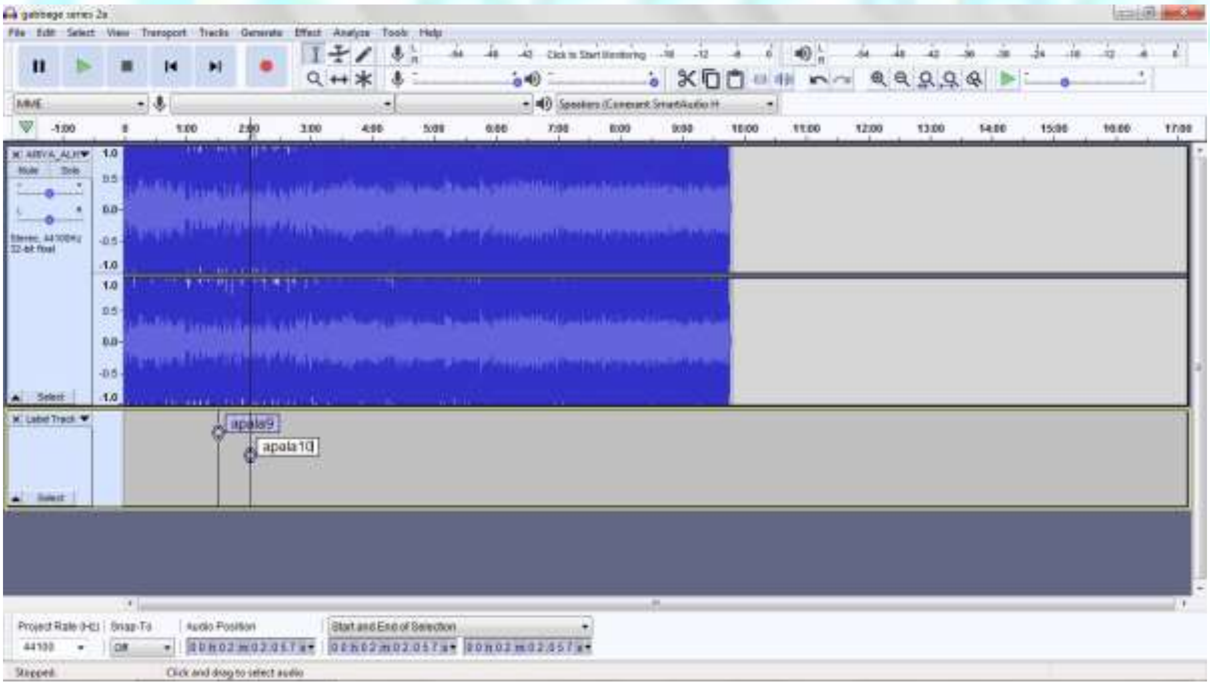
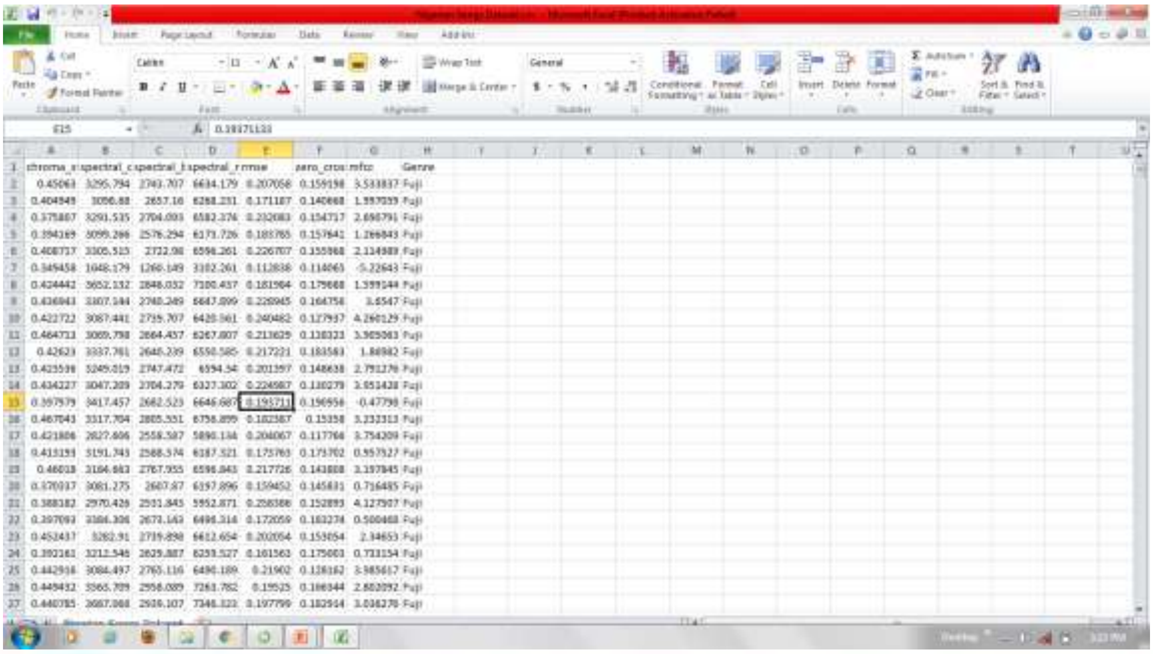

File Edit Select View Transport Tracks Generate Effect Analyze Tools Help
MME
Speakers (Conexant SmartAudio H
apala9
apala 10
Project Rate (Hz)
Snap-To
Audio Position
Start and End of Selection
Stopped.
Click and drag to select audio

Figure 2 Screenshot of Audacity Software

4.2 Feature Extraction

Seven content based features were extracted from the 1000 audio clips which served as the input into the classifier. A python library called librosa was used for the extraction of the features. The files were saved on github. Google colab was the online integrated development environment (IDE) where we ran our codes to extract the features. Figure 3 shows the screenshot of part of the extracted features in a csv file format.



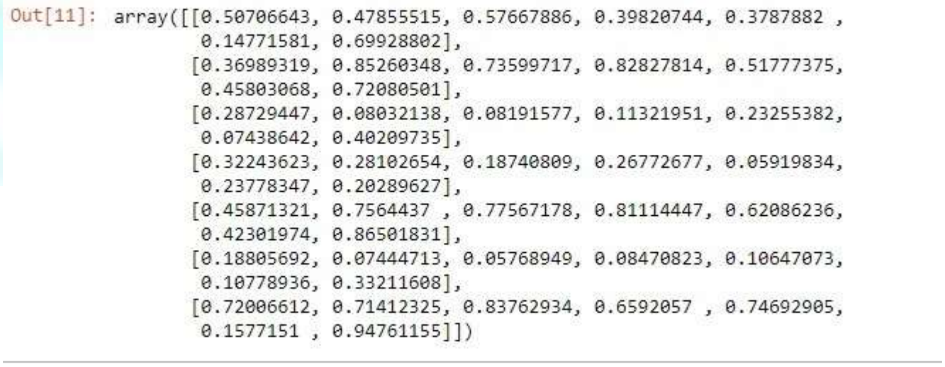
E15 0.19371133

	chroma_s	spectral_c	spectral_b	spectral_r	rmse	zero_cros	mfcc	Genre
2	0.45063	3295.794	2743.707	6634.179	0.207058	0.159198	3.533837	Fuji
3	0.404949	3056.88	2657.16	6288.231	0.171187	0.140668	1.557059	Fuji
4	0.375807	3291.535	2704.093	6582.376	0.232083	0.154717	2.698791	Fuji
5	0.394169	3099.266	2576.294	6171.726	0.183765	0.157641	1.266843	Fuji
6	0.408717	3305.515	2722.98	6598.261	0.226707	0.155968	2.114989	Fuji
7	0.349458	1048.179	1260.149	3102.261	0.112838	0.114065	-5.22643	Fuji
8	0.424442	3652.132	2848.032	7100.437	0.181964	0.179668	1.399144	Fuji
9	0.436943	3307.144	2740.249	6647.099	0.228945	0.164758	3.6547	Fuji
10	0.422722	3087.441	2735.707	6420.561	0.240482	0.127937	4.260129	Fuji
11	0.464713	3069.798	2664.457	6267.807	0.213629	0.138323	3.905083	Fuji
12	0.42623	3337.761	2646.239	6550.585	0.217221	0.183583	1.88982	Fuji
13	0.423598	3249.019	2747.472	6594.54	0.201397	0.148638	2.791276	Fuji
14	0.434227	3047.209	2704.279	6327.302	0.224987	0.130279	3.951428	Fuji
15	0.397979	3417.457	2682.523	6646.687	0.193711	0.190956	-0.47798	Fuji
16	0.467043	3317.704	2805.351	6758.899	0.182387	0.15358	3.232313	Fuji
17	0.421806	2827.606	2558.587	5890.134	0.204067	0.117764	3.754209	Fuji
18	0.413193	3191.743	2588.574	6187.521	0.175763	0.173702	0.957527	Fuji
19	0.46018	3184.663	2767.955	6598.843	0.217726	0.143808	3.197845	Fuji
20	0.370037	3081.275	2607.87	6197.896	0.159452	0.145831	0.716485	Fuji
21	0.388182	2970.426	2531.845	5952.871	0.258386	0.152893	4.127907	Fuji
22	0.397093	3386.306	2673.163	6498.314	0.172059	0.183274	0.500468	Fuji
23	0.452437	3282.91	2739.898	6612.654	0.202054	0.153054	2.34653	Fuji
24	0.392161	3212.546	2829.887	6259.527	0.161563	0.175003	0.733154	Fuji
25	0.442916	3084.497	2765.116	6490.189	0.21902	0.128162	3.985617	Fuji
26	0.449432	3565.709	2958.089	7261.782	0.19525	0.166344	2.802092	Fuji
27	0.440785	3687.088	2926.107	7348.323	0.197799	0.182944	3.038278	Fuji

Figure 3 Screenshot of part of the extracted features

4.3. Data Preprocessing

The dataset was preprocessed with the MinMax scaler as discussed in section 3.3.1. The MinMax scaler was used in transforming the data into a standard scale in order to force the features to appear similar. This technique scales the feature value to a fixed range between 0 and 1. This results in smaller standard deviations and suppresses the effect of outliers. Figure 4 shows the result of part of the scaled data (first 7 rows).



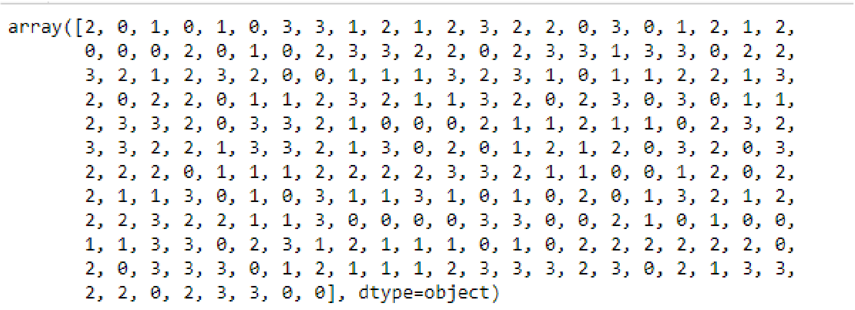
```
Out[11]: array([[0.50706643, 0.47855515, 0.57667886, 0.39820744, 0.3787882 ,
                 0.14771581, 0.69928802],
                [0.36989319, 0.85260348, 0.73599717, 0.82827814, 0.51777375,
                 0.45803068, 0.72080501],
                [0.28729447, 0.08032138, 0.08191577, 0.11321951, 0.23255382,
                 0.07438642, 0.40209735],
                [0.32243623, 0.28102654, 0.18740809, 0.26772677, 0.05919834,
                 0.23778347, 0.20289627],
                [0.45871321, 0.7564437 , 0.77567178, 0.81114447, 0.62086236,
                 0.42301974, 0.86501831],
                [0.18805692, 0.07444713, 0.05768949, 0.08470823, 0.10647073,
                 0.10778936, 0.33211608],
                [0.72006612, 0.71412325, 0.83762934, 0.6592057 , 0.74692905,
                 0.1577151 , 0.94761155]])
```

Figure 4 Outcome of the standardized data

The labels in the dataset were further preprocessed with label encoder. Machines understands numeric data and hence the need to convert categorical data (non- numeric data) to numerical data. The only categorical data in the dataset were the target labels and they were therefore converted to numerical data. The conversion from non-numeric to numeric was done alphabetically. Therefore, the outcome for label encoding of the target labels is as follows:

0	means Afro,
1	means Apala
2	means Fuji
3	means Juju

Figure 5 shows the result of the conversion for the first 100 data in the dataset.



```
array([2, 0, 1, 0, 1, 0, 3, 3, 1, 2, 1, 2, 3, 2, 2, 0, 3, 0, 1, 2, 1, 2,
       0, 0, 0, 2, 0, 1, 0, 2, 3, 3, 2, 2, 0, 2, 3, 3, 1, 3, 3, 0, 2, 2,
       3, 2, 1, 2, 3, 2, 0, 0, 1, 1, 1, 3, 2, 3, 1, 0, 1, 1, 2, 2, 1, 3,
       2, 0, 2, 2, 0, 1, 1, 2, 3, 2, 1, 1, 3, 2, 0, 2, 3, 0, 3, 0, 1, 1,
       2, 3, 3, 2, 0, 3, 3, 2, 1, 0, 0, 0, 2, 1, 1, 2, 1, 1, 0, 2, 3, 2,
       3, 3, 2, 2, 1, 3, 3, 2, 1, 3, 0, 2, 0, 1, 2, 1, 2, 0, 3, 2, 0, 3,
       2, 2, 2, 0, 1, 1, 1, 2, 2, 2, 2, 3, 3, 2, 1, 1, 0, 0, 1, 2, 0, 2,
       2, 1, 1, 3, 0, 1, 0, 3, 1, 1, 3, 1, 0, 1, 0, 2, 0, 1, 3, 2, 1, 2,
       2, 2, 3, 2, 2, 1, 1, 3, 0, 0, 0, 0, 3, 3, 0, 0, 2, 1, 0, 1, 0, 0,
       1, 1, 3, 3, 0, 2, 3, 1, 2, 1, 1, 1, 0, 1, 0, 2, 2, 2, 2, 2, 2, 0,
       2, 0, 3, 3, 3, 0, 1, 2, 1, 1, 1, 2, 3, 3, 3, 2, 3, 0, 2, 1, 3, 3,
       2, 2, 0, 2, 3, 3, 0, 0], dtype=object)
```

Figure 5 Label Encoding

4.4 Classification

1D Convolutional Neural Network was employed for the classification after the dataset had been preprocessed. Figure 6 shows the architecture of 1D Convolutional Neural Network.

The summary of the model is further shown in figure 7 and the parameters for the model is as shown in table 1

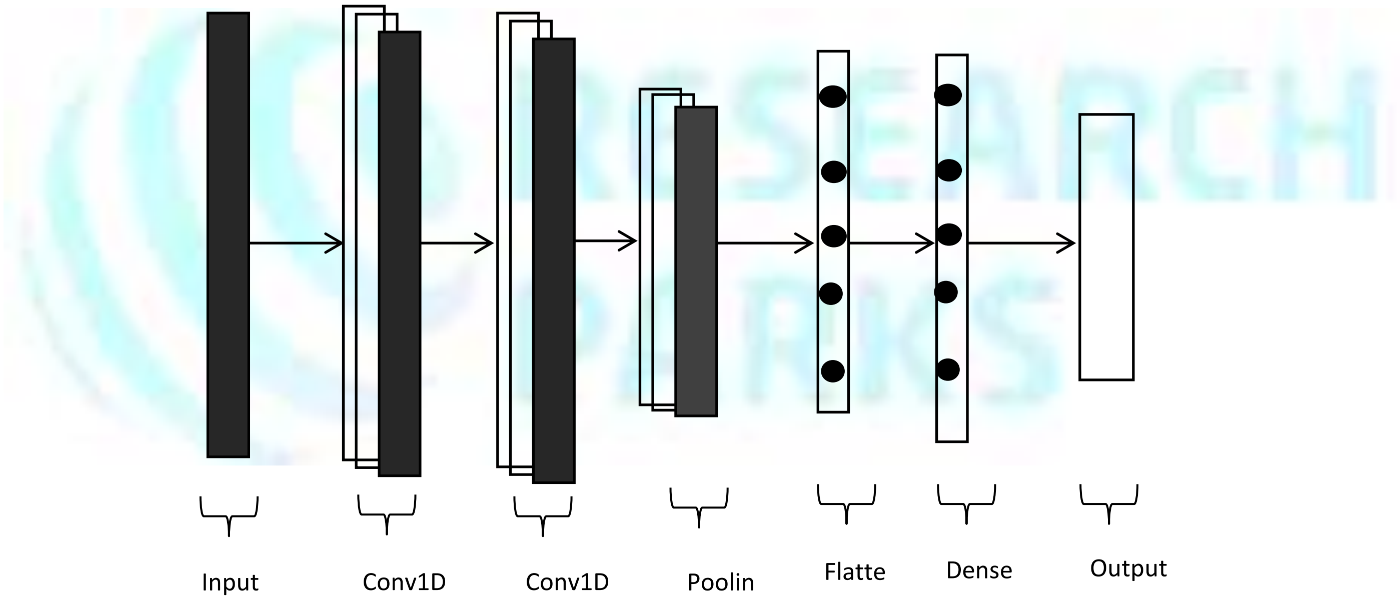
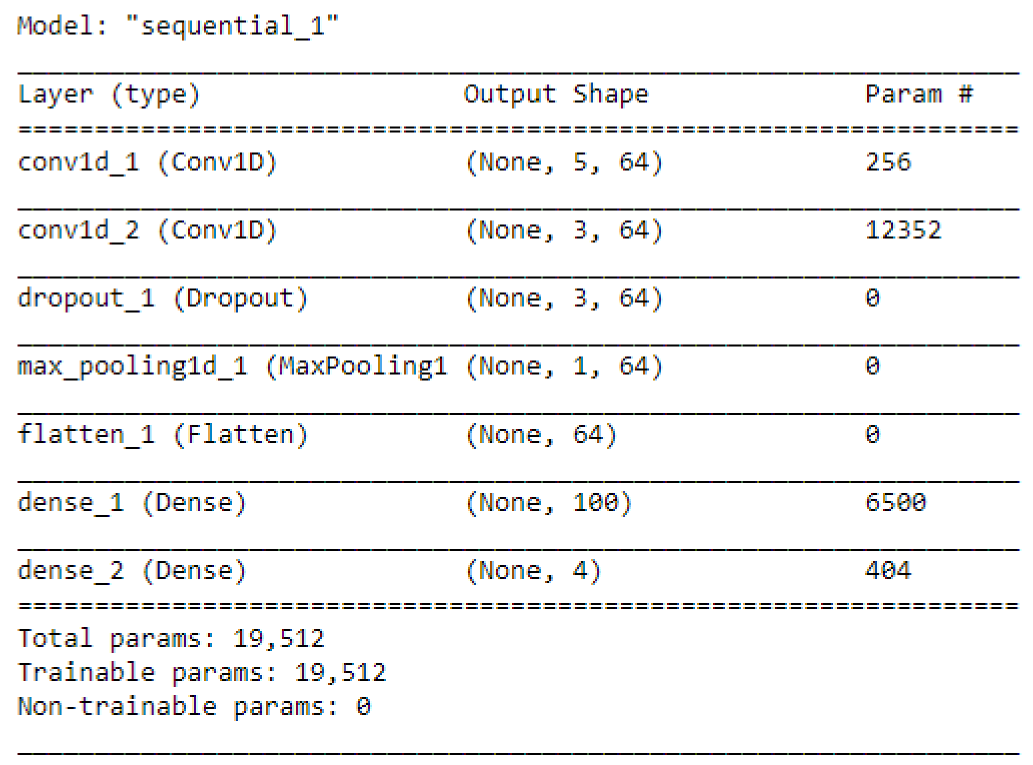


Figure 6 Architecture of 1D Convolutional Neural Network (www.towardsdatascience.com)



```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv1d_1 (Conv1D)            (None, 5, 64)             256       
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 3, 64)             12352     
_________________________________________________________________
dropout_1 (Dropout)          (None, 3, 64)             0         
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 1, 64)             0         
_________________________________________________________________
flatten_1 (Flatten)          (None, 64)                0         
_________________________________________________________________
dense_1 (Dense)              (None, 100)               6500      
_________________________________________________________________
dense_2 (Dense)              (None, 4)                 404       
=================================================================
Total params: 19,512
Trainable params: 19,512
Non-trainable params: 0
_________________________________________________________________
```

Figure 7 1D CNN Model Summary

Table 1 1D CNN Model Hyperparameters

Parameter type	**1D CNN**
Training data	80% of the dataset
Testing data	20% of the dataset
Validation data	Same as the testing data
Batch size	16
Number of epochs	150
Kernel size	3
Dropout	0.5
Pool size	2
Activation functions	relu and softmax
Loss function	categorical crossentropy
Optimizer	Adam

The training and validation accuracy and loss values were obtained, after training and testing the model. The training/ validation accuracy and loss learning curves are as shown in figures 8 and 9 respectively. The summary of the training/validation accuracy and loss values are as shown in table 2

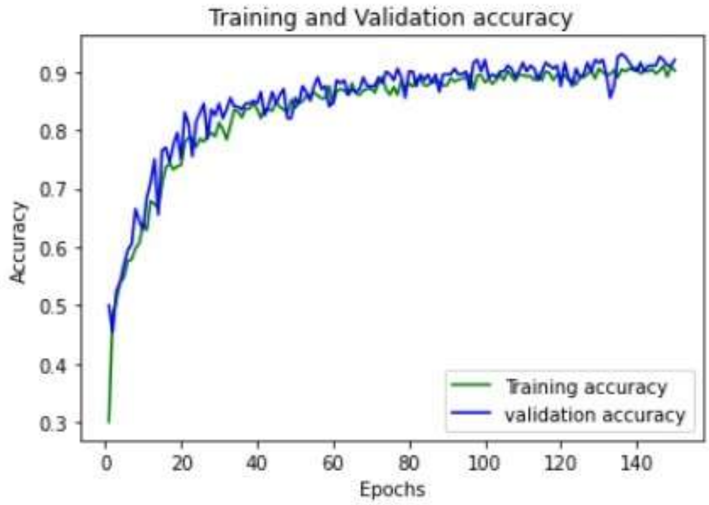


Figure 8 Training and Validation Accuracy over 150 epochs

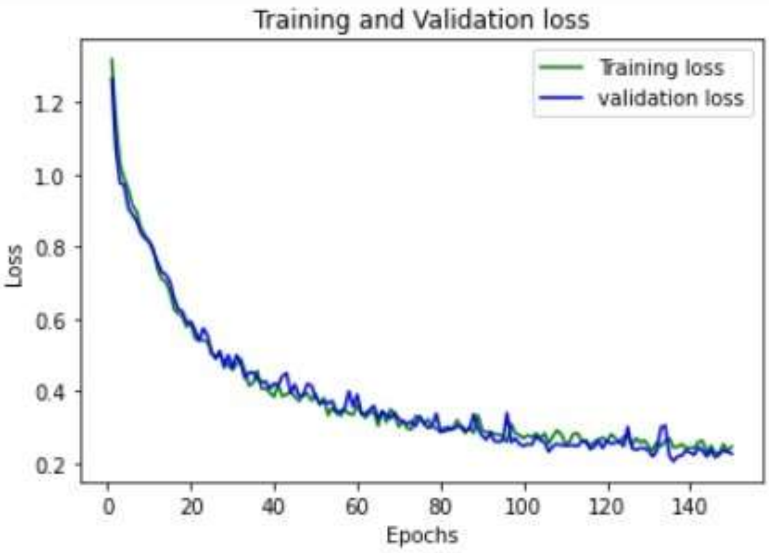


Figure 9 Training and Validation loss over 150 epochs

Table 2 Training/Validation Accuracy and LossValues

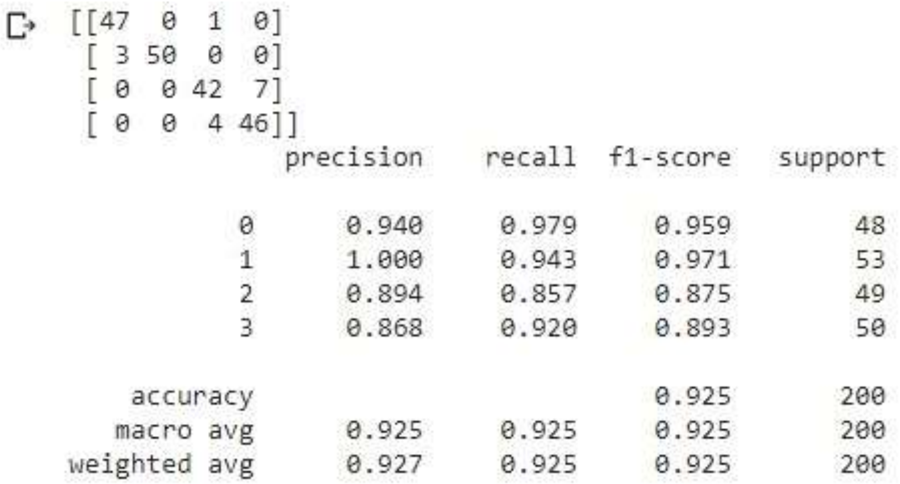
Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
93.88%	16.67%	92.5%	23.54%

4.5 Evaluation

Figure 10 shows the accuracy, precision, recall, f1 score and confusion matrix of 1D Convolutional Network after training and testing of the model.

The weighted average performances of 1D Convolutional Neural Network were 92.7% for precision, 92.5% for recall, 92.5% for f1-score and 92.5% for accuracy. As seen in table 2, the model gave low validation loss and high validation accuracy which depicts a good model.

Table 3 shows some existing works with content based features classified with other machine learning models.



```
[[47  0  1  0]
 [ 3 50  0  0]
 [ 0  0 42  7]
 [ 0  0  4 46]]
              precision    recall  f1-score   support

           0      0.940     0.979     0.959        48
           1      1.000     0.943     0.971        53
           2      0.894     0.857     0.875        49
           3      0.868     0.920     0.893        50

    accuracy                          0.925       200
   macro avg      0.925     0.925     0.925       200
weighted avg      0.927     0.925     0.925       200
```

Figure 10 Classification report

Table 3 Performance of other models

Author(s)	Classifier	Accuracy
Falola & Akinola	**1D CNN**	**92.5%**
Snigdha et al	Logistic Regression	60.892%
	Simple Artificial Neural Network	64.0625%
Hareesh	Logistic Regression	53%
	Random Forest	54%
	Support Vector Machine	57%
	Extreme Gradient Boosting	59%
Vishnupriya & Meenakshi	Convolution Neural Network	47%

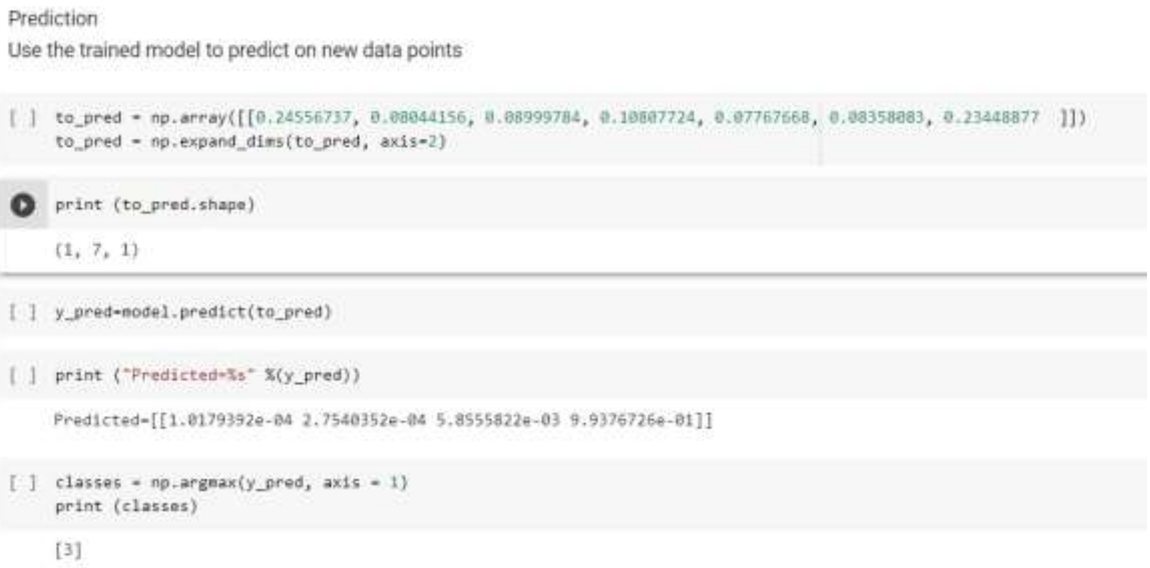
4.6 Results and Discussion

From the results shown in figure, 1D CNN gave an excellent accuracy result of 92.5%, precision score of 92.75%, recall of 92.5% and f1 score of 92.5%. Another insight into the result is that a good model gives a high training and validation accuracy values alongside a low training and validation loss values with an optimal curve (cite). Our model displayed this as seen in figures 8 and 9

Furthermore, a slight difference between the training and validation accuracy and loss curves showed the model learned appropriately.

To further test the accuracy of the model in classifying appropriately, a fuji song (Nigerian traditional genre) using seven preprocessed features was passed into the model to predict the genre of the song. The model predicted the song correctly by returning three as seen in figure 5which is the encoded label for fuji genre in this study.

1D Convolutional Neural Network has proven to be a good classifier for music genre classification with content based features.



Prediction
Use the trained model to predict on new data points

```
[ ] to_pred = np.array([[0.24556737, 0.08044156, 0.08999784, 0.10807724, 0.07767668, 0.08358083, 0.23448877 ]])
    to_pred = np.expand_dims(to_pred, axis=2)

    print (to_pred.shape)

    (1, 7, 1)

[ ] y_pred=model.predict(to_pred)

[ ] print ("Predicted=%s" %(y_pred))

    Predicted=[[1.0179392e-04 2.7540352e-04 5.8555822e-03 9.9376726e-01]]

[ ] classes = np.argmax(y_pred, axis = 1)
    print (classes)

    [3]
```

Figure 5 Prediction on new data point

5 Conclusion

In this study, we have successfully carried out the classification of Nigerian songs into their genres using content based features and 1D Convolutional Neural Network (CNN) being the classifier.

This study was carried out on a dataset which consisted of Nigerian songs of four different genres; Afro, Apala, Fuji and Juju. This dataset was used in order to test the algorithm with the genres as several genres across the world have been widely tested leaving genres peculiar to certain regions untested.

It can be concluded that content based features that were extracted from the songs are excellent features that can be used to classify songs into their genre. Also, 1D Convolutional Neural Network (1D CNN) has proven to be a good classifier for music genre classification from the results delivered from this work.

Future studies should include other untested genres across the world into the dataset in order to have a robust dataset. Some genres may perform differently with certain classifiers due to their structure. It is therefore important to carry out research on all untested genres across the world. Also, other features other than spectrogram and content based features should be researched on for more features to be used for music genre classification.

References

1. Weibin Zhang, Wenkang Lei, Xiangmin Xu, Xiaofeng Xing (2016). Improved Music Genre Classification with Convolutional Neural Network. *Interspeech 2016* .http://dx.doi.org/10.21437/Interspeech.2016-1236. pp. 3304-3308

2. N. Auguin, S. Huang, and P. Fung (2013) "Identification of live or studio versions of a song via supervised learning*," in Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, 2013 Asia-Pacific. IEEE, 2013, pp. 1–4.

3. J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. K´egl (2006) *"Aggregate features and adaboost for music classification," Machine learning*, vol. 65, no. 2-3, pp. 473–484, 2006.

4. S. Sigtia and S. Dixon (2014).*"Improved music feature learning with deep neural networks," in Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on.*IEEE, 2014, pp. 6959–6963.

5. Jia Dai, Wenju Liu, Hao Zheng, Wei Xue, and Chongjia Ni (2016). Semi-supervised Learning

of Bottleneck Feature for Music Genre Classification, *T. Tan et al. (Eds.): CCPR 2016,* Part II, CCIS 663, pp. 552–562.

6. Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, Daniel J. Inman (2020). 1D convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing 151 (2021) 107398, Elsevier,* pp. 1-21

7. Nils (2018). Introduction to 1D Convolutional Neural Network, blog.goodaudience.com/.

8. A. Elbir✉ and N. Aydin. Music genre classification and music recommendation by using deep learning (2020*).Electronics Letters.* Vol. 56, No. 12, pp. 627–629

9. Soumya Suvra Ghosal and Indranil Sarkar. Novel Approach to Music Genre Classification using Clustering Augmented Learning Method (CALM).*Proceedings of the AAAI 2020 Spring Symposium on Combining Machine Learning and Knowledge Engineering in Practice (AAAI-MAKE 2020).*Vol. 2600

10. Snigdha Chillara, Kavitha A S, Shwetha A Neginhal, Shreya Haldia and Vidyullatha K S. (2019). Music Genre Classification using Machine Learning Algorithms: A comparison. *International Research Journal of Engineering and Technology (IRJET)*. Volume: 06 Issue: 05, pp. 851-858

11. Hareesh Bahuleyan (2018). Music Genre Classification using Machine Learning Techniques. arXIV: 1804.01149v1[cs.sd], https://www.researchgate.net/publication/324218667. Accesed in October 2020

12. Vishnupriya S, K. Meenakshi (2018). Automatic Music Genre Classification using Convolution Neural Network.*2018 International Conference on Computer Communication and Informatics (ICCCI - 2017), Coimbatore,* INDIA.IEEE

13. Nikki Pelchat, Craig M Gelowitz. Neural Network Music Genre Classification (2019). *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*

14. Vinicius D. Valerio, Rodolfo M. Pereira, Yandre M. G. Costa, Diego Bertolini and Carlos N. Silla Jr. (2018). A Resampling Approach for Imbalanceness on Music Genre Classification Using Spectrograms. *Association for the Advancement of Artificial Intelligence* (www.aaai.org).pp 500-505.

15. George Tzanetakis and Perry Cook (2002).Musical Genre Classification of Audio Signals, *IEEE Transactions on speech and audio processing,* VOL. 10, NO. 5, pp. 293-302

16. Carlos N. Silla Jr., Celso A. A. Kaestner, Alessandro L. Koerich (2007), Automatic Music Genre Classification Using Ensemble of Classifiers. *XXXIII Seminario Integrado de Software e Hardware.*DOI: 10.1109/ICSMC.2007.4414136

17. Nicolas Scaringella, Giorgio Zoia, and Daniel Mlynek (2006). IEEE Signal Processing Magazine, pp. 133-141

18. Carmine-Emanuele Cella (2015).An Introduction to Audio Features. Coservatorio di Padova. www.carminecella.com. Accessed October, 2019.

19. Zhouyu Fu, Guojun Lu, Kai Ming Ting, and Dengsheng Zhang (2011).A Survey of Audio-Based Music Classification and Annotation. *IEEE Transactions on Multimedia 13(2):*pp: 303 - 319

20. Juan Pablo Bello (2007). Low-level features and timbre. EL9173 Selected Topics in Signal Processing: Audio Content Analysis, NYU Poly, https://s18798.pcdn.co/. Accessed January, 2020.

21. Markus Schedl, Arthur Flexer, JuliánUrbano (2013).The neglected user in music information retrieval research. *Journal of Intelligent Information, volume 41,* pp. 523–539

22. Li D, Sethi I, Dimitrova N, McGee T (2001) Classification of general audio data for Content based retrieval. *Pattern Recognition Letters 22:533–544*

23. Claus Weihs, Uwe Ligges, Fabian M¨orchen and Daniel M¨ullensiefen (2007). Classification in Music Research, *Article in Advances in Data Analysis and Classification*, pp. 1-36

24. Dan Ellis. 2007. Chroma feature analysis and synthesis. Resources of Laboratory for the Recognition and Organization of Speech and Audio LabROSA. https://www.ee.columbia.edu/~dpwe/resources/matlab/chroma-ansyn/. Accessed October, 2019

25. Signal Analysis and Feature Extraction, https://www.musicinformationretrieval.com/, Accessed November, 2019

26. Steve Tjoa. 2017. Music information retrieval. https://musicinformationretrieval.com/mfcc.html. Accessed February, 2020

27. Fabien Guoyon, Franchois Pachet and Olivier Delerue (2000). On The Use of Zero-Crossing Rate For Application of Classification of Percussive Sounds. *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-00),* Verona, Italy, December 7-9, 2000.

28. Yangqiu Song, Changshui Zhang and Shiming Xiang (2007). Semi-Supervised Music Genre Classification. *ICASSP 2007*. pp. 729-732.

29. Music Structure Analysis, https://www.musicinformationretrieval.com/, Accessed December, 2019

30. Archit Rathore and Margaux Dorido (2015).Music Genre Classification. *Department of Computer Science and Engineering. Indian Institute of Technology, Kanpur*.www.semanticsscholar.org/paper/MusicGenreClassificationRathore/ Accessed January, 2020.

31. W. Chai (2006). “Semantic segmentation and summarization of music,” IEEE Signal Processing Mag., vol. 23, no. 2, pp. 124–132,

32. YI-Hsuan Yang and Homer H. Chen (2012). Machine Recognition of Music Emotion: A Review. ACM Transactions on Intelligent Systems and Technology, Vol 3, No 3, Article40, pp: 1-40.

33. LU, L., L IU, D., AND Z HANG, H. 2006. Automatic Mood Detection And Tracking Of Music Audio Signals. IEEE Trans. Audio, Speech Lang. Process. 14, 1, 5–18.

34. Klapuri, A. (1999). Sound Onset Detection By Applying Psychoacoustic Knowledge. In Proc. IEEE International Conference on Acoustics, Speech and Signal Processing

35. Olivier Lartillot, Petri Toivainen (2007). A Matlab toolbox for musical feature extraction from audio. International conference on digital audi effects. pp. 237-244

36. Roy (2020). All about feature scaling, https:towardsdatascience.com/, Accessed April, 2020

37. MinMax Scaling, https://rajeshmahajan.com/.Accessed December, 2019

38. Zakaria Jaadi (2019).Everything You Need To Know About Interpreting Correlations, https://towardsdatascience.com/Accessed June, 2019.

39. Jason Brownlee, (2019). How To Calculate Correlation Between Variables in Python. https://machinelearningmastery.com/how-to-use-correlation-to-understand-the-relationship-between-variables/. Accessed January, 2020.

40. Nils (2018). Introduction to 1D Convolutional Neural Network, blog.goodaudience.com/.

41. Convolutional Neural Network, https://en.m.wikipedia.org/. Accessed December, 2019

42. Ravisutha Sakrepatna Srinivasamurthy (2018). Understanding 1D Convolutional Neural

Network Using Multiclass Time-Varying Signals. All Theses 2911. https://tigerprints.clemson.edu/all_theses/2911

43. Max-pooling/Pooling, http://www.computersciencewiki.org/. Accessed November, 2019

44. Jason Brownlee, (2019). A gentle introduction to pooling layers for convolutional neural networks, https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/. Accessed January,2020

45. Activation Functions, www.365datascience.com. Accessed April, 2020

46. Sanket Doshi (2019). Various Optimization Algorithms For Training Neural Network. https://towardsdatascience.com/. Accessed March, 2020

47. Sagar Sharma (2017). Epoch vs Batch Size vs Iterations. https://towardsdatascience.com/. Accessed March, 2020

48. Vinita (2019). How to interpret loss and accuracy for a machine learning model, https://intellipaat.com/. Accessed December, 2020