Music Genre Classification with Convolutional Neural Network

Ibnul Islam Tilok
Dept. of Computer Science and Engineering
BRAC University
66 Mohakhali, Dhaka
Bangladesh
ibnul.islam.tilok@g.bracu.ac.bd

Masud Chowdhury
Dept. of Computer Science and Engineering
BRAC University
66 Mohakhali, Dhaka
Bangladesh
masud.chowdhury@g,bracu.ac.bd

Prodipta Das
Dept. of Computer Science and Engineering
BRAC University
66 Mohakhali, Dhaka
Bangladesh
prodipta.das@g.bracu.ac.bd

Avoy Chowdhury
Dept. of Computer Science and Engineering
BRAC University
66 Mohakhali, Dhaka
Bangladesh
avoy.chowdhury@g.bracu.ac.bd

Md. Abdullah Al Masum Anas
Dept. of Computer Science and Engineering
BRAC University
66 Mohakhali, Dhaka
Bangladesh
email address or ORCID

***Abstract*—Today, Music is one of the effective forms of entertainment. Everyday new Music is being composed, and the quantity of Music is increasing day by day. So, it is essential to classify or categorize Music into different genre forms accurately. Classification of Music is necessary as it enables us to differentiate the Music based on the genre. The main objective of our thesis is to extract the music feature and classify or categorize Music based on the genre. The aim is to predict the genre with the help of convolutional neural networks. There are many techniques to classify genres, but convolutional neural networks give more accuracy than other techniques. The audio dataset is collected here, and the audio signal has been converted into a spectrogram. After generating a spectrogram, CNN will give predictions based on the sample provided. Our work will give improvement to various audio and music applications. We will train the CNN to provide predictions more accurately by feeding it with huge batches of data samples.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Music genre Classification is classifying the music based on its genre. Usually, there are many types of genres like classical, country, disco, Metal, etc. Music genres can be classified by different kinds of models like L.S.T.M, R.N.N, S.V.M, D.N.N, etc. the main goal of our thesis is to improve the accuracy of the genre. Therefore, we will follow the different models and check which model performs better. Our paper mainly worked with a Convolutional neural network; here, we followed the sequential model. Besides this, we also worked with L.S.T.M and the k-mean clustering algoirithm. The main goal is to compare the model's accuracy and justify which model is suitable for classification. Moreover, we can also check which model performs better with less dataset and which performs better on a large dataset. Also, music genres are hard to coherently and consistently label due to their inherently subjective nature. After a discussion, we fix that if we use CNN for our system, hopefully, we can achieve a better result than others. We need to process our image using CNN, which is very efficient. According to our research, we found that most of the studies use handmade feature extraction techniques. There are too many differences between CNN and handmade techniques. CNN architecture never depends on segmentation which human experts do. Because of millions of learning able parameters, CNN catches more data.

II. Related Work

Presently, we found a massive number of songs in an online music database. So, it is pretty hard to choose your desired song from that list. We built this system to classify different songs based on their genre. Using deep learning,

we can easily classify music rather than handmade features. Today, Deep learning plays a vital role in the Classification of data. CNN is a very popular model for genre classification among deep learning because of its high accuracy. According to the article, CNN has achieved brilliant performance in various fields. In papers [10], we studied the author using the RNN-LSTM model for genre classification. According to their evaluation, they found 89hundred epochs. The accuracy achieved for 1D CNN with four layers was 69 percent, 75 percent, 79 percent, and 80 percent on the test set. Acoustic scene categorization uses a solid deep feature extraction approach with an accuracy of 81.9 percent on the ESC-50 dataset. Another author [5] suggested classifying music using a different model automatically. The author uses a spectrogram to train the CNN model from start to finish, whereas the second approach uses ML algorithms such as Logistic Regression, Random Forest, and others. VGG-16 with an accuracy of 89 percent, the CNN model was the most accurate. On the other hand, we introduced CNN for our system and MFCC and spectrograms. Spectrogram represents signal strength using visual ways. We can use spectrograms in different ways, which carry Fourier transform, wavelet transform, and band-pass filter. After a discussion, we fix that if we use CNN for our system, hopefully, we can achieve a better result than others. We need to process our image using CNN, which is very efficient. According to our research, we found that most of the studies use handmade feature extraction techniques. There are too many differences between CNN and handmade techniques. CNN architecture never depends on segmentation which human experts do. Because of millions of learning-able parameters, CNN catches more data. The music genre is commonly classified in two ways nowadays. Firstly, the features extracted from the dataset and the group of feature tables have been built. Secondly, algorithms like SVM, KNN, LSTM, and many other models were used.DNN has become famous for training large numbers of music or data samples. The problem with DNN is it cannot detect the audio sequence. Therefore, frame-based training. The model will skip the sequence information while training. The sequence information is essential for categorizing the dataset. The recurrent neural network has been used for sequence labeling tasks, but it lacks its limited storage as it has to deal with long sequences 3 due to vanishing and exploding gradients. As a result, it becomes tough for longterm dependencies.

III. Related Work

A. Dataset

We have used the GTZAN dataset, which is very popular for retrieving information. Moreover, the dataset contains a thousand audio songs, each 30 seconds duration. Also, our dataset has ten genres; each of the genres contains hundred audio tracks. The genres are Reggae, hip-hop, jazz, pop, rock, classical, country, disco, metal, and blue.

B. Data Prepossessing

Preprocessing is a way of making raw data suitable for machine learning models. It is essential as a dataset may contain noise, missing value, an unstable format, which cannot be used for a machine learning model. Therefore, the data should be preprocessed to increase the model's accuracy and efficiency. We usually convert the dataset into a CSV file to use the dataset. CSV(Comma-Separated Values) files allow us to save the tabular data such as spreadsheets, which is very useful for large datasets. Also, we used our dataset and converted them into a JSON file to use as an input file. To perform preprocessing, we have to import some libraries like NumPy (used for mathematical operation or scientific Calculation), matplotlib (which is a plotting library) pandas (used for importing and managing the dataset). After importing the dataset, we need to set a directory. Splitting the dataset into the training and test sets is also essential because it enhances the machine learning model's performance. Moreover, Feature scaling is also significant as it ensures all the data are scaled and standardize the independent variable of the dataset in a specific range. In feature scaling, all the variables are kept in the same scale and range so that no variable can dominate other variables.

C. Feature Extraction

Our paper applied two feature extraction techniques to know the result and accuracy of different techniques. We are using a convolutional neural network, We will require an image as input data. Therefore, we have to extract the following feature and save it as an input file. Therefore, we are using python library librosa to extract features of audio signals. Then we are using librosa.display to generate Spectral Centroid, Chroma features, Spectral Bandwidth, Zero Crossing Rate, MFCC, Spectrogram, etc., from the audio data in a different format.

Spectrogram- In a spectrogram, the frequency is displayed on the y-axis and time on the x-axis. It is the visual representation of loudness over different frequencies at a particular time. Moreover, the heat map is used for denoting spectrogram. The data is converted in the short-term Fourier transform to know the amplitude of the frequency.

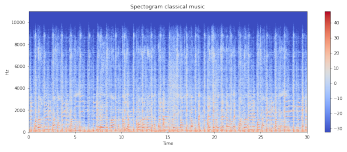

Fig. 1. Spectrogram

Mel-Spectrogram- The Mel scale is a set of pitches that are comparable in the distance to one another and maybe felt by the listener. The Mel spectrogram is created by converting frequencies to the Mel scale. The Fourier transform is used to

convert frequencies to the Mel scale. Basically, the Mel spectrum converts frequency into Mel scale by Fourier transform.

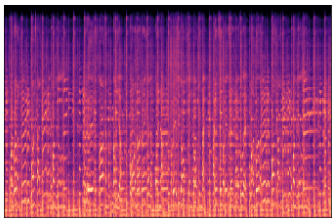

Fig. 2. Mel-Spectrogram

Zero-crossing rate- It is the number of times audio waves cross zero. Zero crossing is mainly used for detecting sound or audio. We used the librosa.zerocrossing for extracting the feature.

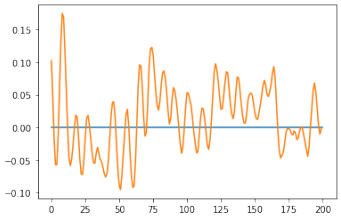

Fig. 3. Zero Crossing Rate

Spectral- centroid- We calculated spectral- centroid as the mean weighted of frequency located in the signal, which is determined by using a Fourier transform. We used the librosa.feature.spectralcentroid.

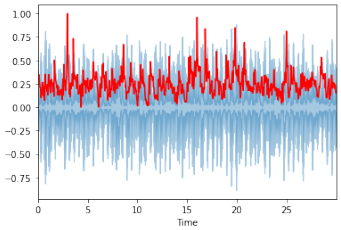

Fig. 4. Spectral- centroid

Mel frequency Cepstral Coefficients (MFCCs)-The MFCC basically describes the shape of the spectral. we have extracted the MFCC of the audio data. Here in the MFCC coefficients are represented in the y axis and time in the x-axis. We have used 13 strides for the JSON file. We used librosa.feature.mfcc to generate the MFCC.

Rolloff frequency- The shape of the signal is measured by spectral Rolloff. Here it is defined as the percentage of power spectral distribution. The ratio is usually $85 - 95\%$. The roll-off point below is concentrated at 85-95 percent of the magnitude distribution. It is used to differentiate between noisy and harmonic sounds.

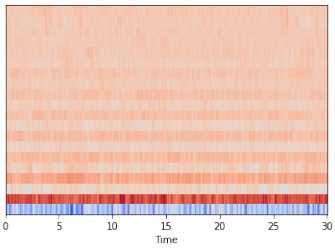

Fig. 5. MFCC

Spectral Contrast- - Spectral contrast considers the spectral peak, the spectral valley, and the difference in each frequency sub-band. Here the power of frequency defers over time, and measuring energy gets difficult.

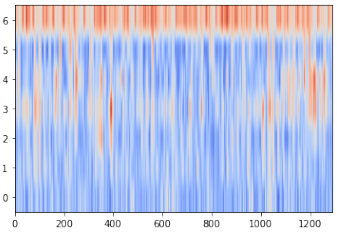

Fig. 6. Spectral Contrast

Spectral bandwidth- Bandwidth help to distinguish between the upward and downward frequency.

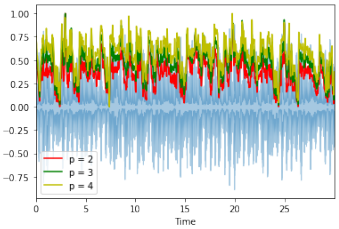

Fig. 7. Spectral Bandwidth

D. Feature Extraction Technique

We have used the librosa library to display the spectrogram. At first, we have extracted the waveform where amplitude was displayed in the y-axis and time in the x-axis. We also displayed the waveform with librosa.display.waveplot. Here the sample rate was 2250. Then we have extracted the power spectrum, which displays magnitude in the y-axis and frequency in the x-axis. After this, by using short term Fourier transform, we have displayed the spectrogram by librosa.core.stft where the frequency is in the y axis and time in the x-axis. Here the number of samples per fast Fourier transform is 2048. Moreover, we used 512 hop lengths. After performing log spectrogram then we extract MFCC. To extract MFCC we used librosa.feature.mfcc. Here we can see how we have extracted a signal from a wave file then performed

a Fourier transform, how to arrive at a power spectrum, spectrogram and log spectrogram, and, importantly, mfccs. Finally, we will export the data to a JSON file. This is all the data we need for preprocessing audio data to export into JSON files.

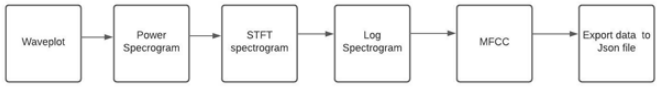

Fig. 8. Feature Extraction in JSON file

In this feature extraction Figure 3.9, we used the same dataset to extract all these features. First, we took a single audio file and loaded it using the Python Librosa module. Then we extracted all spectrograms, MFCC, and all these features are shown in the feature extraction section and exported all the data to a CSV file to train and test our data. We will be using Keras Sequential and K-Mean clustering model to get our accuracy and result.

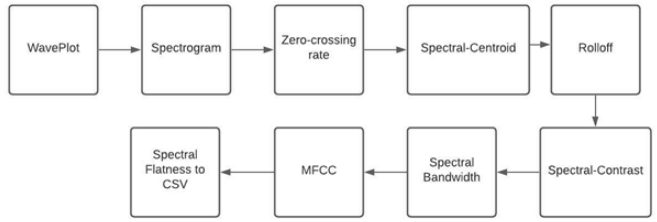

Fig. 9. Feature Extraction in CSV file

IV. CNN Analysis

A. CNN Model Architecture

A convolution tool extracts and identifies the different features of a data for analysis in a process known as Feature Extraction. A fully connected layer that uses the convolutional output to forecast the image's class based on the characteristics retrieved earlier in the algorithm. Convolutional, pooling, and fully-connected layers that make up the Convolutional Neural Network architecture. A CNN architecture will emerge when these layers are added. The dropout layer and the activation function, in addition to these three layers.

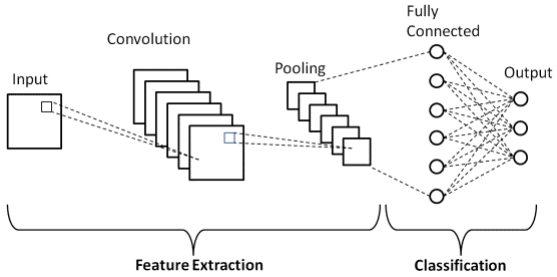

Fig. 10. CNN Architecture

1. Convolutional Layer: The most important layer in CNN is convolutional layer. This layer performs convolution among the input data and a filter of a certain grid. In our model we are using 3x3 filter. It works by placing a filter or kernel over an array of image pixels. The filter then slide in the image data and generate an output. This creates what is called the feature, and it contains information about the input data. Then, the feature map is given to other layers to learn more input features. This layer helps network to learn to extract different types of features.

2. Pooling Layer: What pooling does is downsize the feature map. It shrinks the image data. There is two commonly used pooling layer which is max-pooling and average pooling. However, the max-pooling layer is the one that is used primarily. We are using the max-pooling layer in our model. Max pool reduces the size of a sample by taking the highest value and logging it to the output. This makes processing much faster as it reduces the number of parameters.

3. Fully Connected Layer: The Fully Connected layer is the last few layers of a CNN architecture initiated prior to the output layer and connects the neurons between two separate layers. In this stage, the classification process starts with the input image being flattened and fed to the FC layer.

5. Activation function: Activation functions are a key part of the CNN model architecture. The activation function used After doing the internal processing of each node, the activation function is used. Besides, it shows the flow of the information in both forward direction and end direction at the network's end. The ReLU function and Softmax functions are often used in activation functions.

V. Model Aanlysis and Classification

A. Sequential Model and CNN Classification with CSV

We are using CNN algorithm for training our model for our paper. We preferred this method because various forms of research display it to have the best results for this problem. We used the sequential model because it is easier to build a model in Keras.

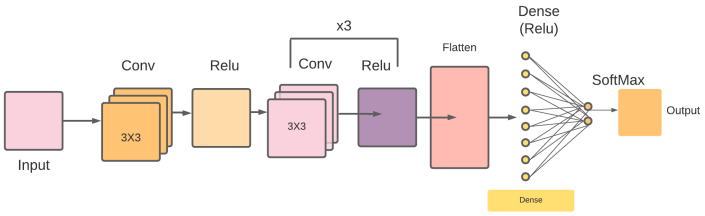

Fig. 11. Working Convolutional Neural Network Model 1

Here, we are using the CSV file we extracted, which contains features of the audio files. As you can see, we specify our Convolutional layers in sequential order. Here, we also used PCA to scale our data. The first step is to standardize normalization. It means the feature is a separate

random variable, and the standard scalar has been used to rescale the value into the same unit. Standard scalar will be responsible for converting the data where the mean will be zero, and the standard deviation will be 1. Then perform fit on the data frame and apply to transform, and it will help all the values transform into the same scale value. We use Flatten and have two dense layers to generate the classification. Moreover, the Adam optimizer is used. Initially, we set the epoch for the training model 100 and started to exceed 300 gradually. We have used the rectified linear unit as the activation. Rectified linear unit or ReLu is used as it makes it optimizable. The loss has been calculated using the sparse categorical cross-entropy function. Here, the flatten layer's task is to flatten the output into the dense layer. We have also used dropout to solve overfitting by dropping the neuron randomly. We selected the Adam optimizer because it delivered the best results after evaluating other optimizers. After that, the accuracy we achieved for the test set is 92.05 percent which is the best we got compared to other models.

B. Sequential Model and CNN Classification with JSON

Here, we are also using CNN to train our model. We preferred this method because various forms of research display it to have the best results for this problem. We used the sequential model because it is easier to build a model in Keras.

The model used in the above figure is the sequential model of convolutional neural network architecture. We can see in the above figure the input data is the JSON file which we extracted from figure 3.8. we have passed the input JSON file into the convolutional layer. Here the number of filters is 32, and we follow the 3X3 kernel for the first layer. Moreover, the rectified linear unit will be used. The primary function of the rectified linear unit is to maintain the dimensionality of the layer. Here, if we get any negative value,, the rectified linear unit will convert the data to zero to maintain the layer's dimensionality. Moreover, we have also used max-pooling in the process. The main function of max-pooling is to shrink the image. There will be three convolutional layers, and the above process will be repeated every time. Then, data will be flattened to a dense layer. The main function of the flatten layer is to convert the two-dimension output into one dimension output. The next important step is to feed the 1-dimension output into the dense layer. We have done dropout in order so that the robustness of the training process increases. Besides this, the dropout plays a vital role in solving the overfitting. Here the dropout drops the neuron randomly. In here, we dropped out thirty percent of the neuron. We have also used batch normalization to increase the speed of the training process. SoftMax is also used last for classification at last. It is mainly used for multiclass classification. After following the step, we achieved 74.96 percent accuracy for the test set.

VI. Result Summery and Analysis

A.

Summery	Params	Size	Result
Params	Total Params Trainable Params Non-Trainable Params	202,826 202,826 0	
Epochs	300		
Result	loss accuracy val_loss val_accuracy	0.0241 0.9920 0.4393 0.9307	92.05%

TABLE I
SEQUENTIAL MODEL 1 WITH CNN (CSV FILE)

Table I shows the result of Keras sequential model followed by a convolution neural network. Here the input data is the CSV file. In figure 3.9, it is clearly shown how we have collected the CSV file. Here, the total trainable parameter is over 200k, and all of the data are trainable. There is zero non-trainable data. Here we started with 70 epochs as training time, got 87 percent accuracy, increased the epoch to 300, and found out the loss was reduced, and accuracy increased to 92 percent. We got 92.05 percent accuracy from the CSV file. This is because the CSV file consists of lots of data, and the CSV contains lots of features has been like spectrogram, zero-crossing rate, spectral- centroid, Rolloff, spectral-Contrast, Spectral Bandwidth, MFCC, etc. Due to the vast of data, the accuracy achieved is 92%.

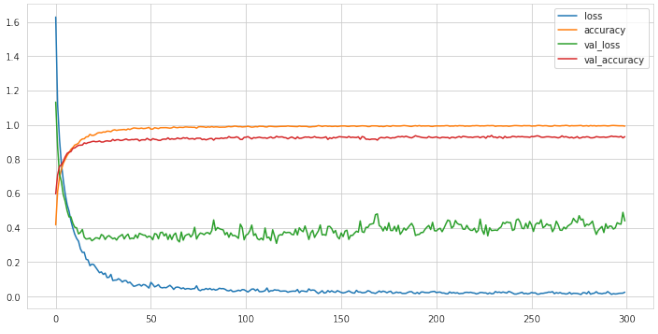

Fig. 12. Accuracy of the model

B.

Summery	Params	Size	Result
Params	Total Params Trainable Params Non-Trainable Params	45,644 45,452 192	
Epochs	100		
Result	loss accuracy val_loss val_accuracy	0.2056 0.9327 0.8575 0.7437	74.97%

TABLE II
SEQUENTIAL MODEL 2 WITH CNN (JSON FILE)

Table II shows us that here JSON file has been taken as input. From our model, we got the epoch was 100, and the

accuracy was 74.97 percent. We used the sequential model and CNN architecture here. However, as input files, we used the JSON file we extracted, which contains features of the audio files. This is why we were getting a lower accuracy here.

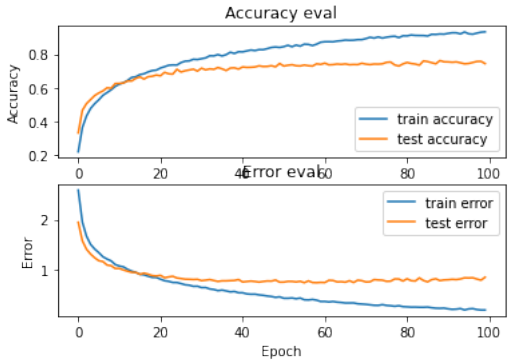

Fig. 13. Accuracy of the model

VII. Conclusion

Music is a very recognized form of entertainment in our society. Every day the number of songs is increasing. Therefore, it has become essential for the Classification of music for the user to categorize the song based on the genre easily. Besides this, the Classification of the genre accurately and categorization will help the user organize the music accordingly. CNN gives better accuracy because the model can work with fewer parameters. The CNN is trained by feeding the dataset to make the prediction accurate. To conclude, we can say CNN performs better than other models like LSTM and K-mean clustering. In short, we have used LSTM, k- mean clustering and sequential model for both JSON file and CSV file. We found out from our thesis that the sequential model gives the best accuracy with the CSV file as input data.

VIII. Future Work

We have tested the accuracy of the genre with other algorithms like RNN-LSTM, K- Means clustering and found out that the convolutional neural network gives the highest accuracy, which is 92 percent. We can also work with a different model like DNN to judge whether it performs better than a CNN or not. Besides this, we will test more models, and overfitting needs to be solved to get the highest accuracy. We believe more data can give better accuracy. So, we may work or modify the dataset to see a better result. In addition, we can also work with the big or different datasets to check if we get better results or not. We should also focus on the problem like overfitting in order to get better accuracy.

IX. Acknowledgment

We want to acknowledge our Advisor Mr. Moin Mostakim sir, for his generous support and guidance in our work. He supported us whenever we needed help. Also, we would like to acknowledge with gratefulness, The support and love of our Parents throughout the Journey by keeping us in their prayers.