Music Genre Classification Using Convolutional Neural Networks

Tehsin Sherasiya
Department of Computer Science
Birla Institute of Technology and Science, Pilani
Dubai, U.A.E
f20180207@dubai.bits-pilani.ac.in

Meriam Joseph
Department of Computer Science
Birla Institute of Technology and Science, Pilani
Dubai, U.A.E
f20180291@dubai.bits-pilani.ac.in

Shruthi Vandhana V
Department of Computer Science Birla Institute of Technology and Science, Pilani
Dubai, U.A.E
f20180284@dubai.bits-pilani.ac.in

Abstract—**Music genres are categories that help organize artists, albums and songs into groups that share similar musical characteristics. This has been done manually for many years, however, today, automated music genre classification is the norm. Automatic music genre classification is important in the age of digital music and music platforms, as it allows companies to recommend similar music to their customers, and even offer this service as a product. In this paper, we make use of the GTZAN dataset, and digital signal processing techniques like mel spectrograms to extract features of different kinds of music and then classify this music by making use of machine learning and deep learning methods like convolutional neural networks (CNNs). We shall also determine the accuracy measures to evaluate the performance of the CNN.**

Keywords—convolutional neural networks, music, genre, classification, mel spectrogram, GTZAN dataset.

I. Introduction

Music genres are labels used on songs to identify or organize the style of music that the song belongs to. Music Genre Classification was introduced in 2002 by Tzanetakis and Cook as part of a pattern recognition task. [1] It is a topic that comes under Music Information Retrieval and has since been the most popular way to classify digital music. Most research related to this classification has made use of the GTZAN and the FMA dataset, of which we shall make use of the GTZAN dataset. [2]

In the 21st century, due to the widespread usage of internet and smart applications, digital music can now be consumed by the public anywhere and anytime. These apps are attractive, as they minimize the space required to store music, reduce hardware requirements, allow interaction with other users, and allow exploration and discovery of new music. Moreover, most music streaming platforms and applications like Spotify, on top of providing access to billions of songs and playlists, even provide recommendations based on previously listened to music. These recommendation systems work based on a Music Genre Classification (MGC) system and some applications like Shazam, even offer this service as a product. [3] Music Genre Classifiers are also useful in building online libraries to organize large collections of music to access them better. They can also be used to sort any new music into its respective genres and hence is an integral feature of any automated service related to music.

Machine Learning is a field that has grown at a rapid rate in the past years. It is a data analysis method that automates model building, which means that it allows a system to analyze data, identify patterns and make predictions with minimal human intervention. It provides computers the capability to learn and improve without being explicitly programmed. Machine Learning quite often takes inspiration from nature to develop algorithms to train computers. Artificial Neural Networks modeled loosely after the connection between neurons in the human brain is an example of this. Artificial Neural Networks or ANNs, are composed of nodes which are connected to other nodes to send and receive data. These nodes or neurons are commonly aggregated into layers and different layers may perform different functions. One of the many types of neural networks is the Convolutional Neural Networks or CNNs. It is a deep learning algorithm that takes an image input, assigns weights and biases, which are updated through a backpropagation process to various aspects of the image and finally classify the images. We have chosen CNN since the preprocessing required in such a network is much lower compared to other classification algorithms. The architecture for CNNs were inspired by the visual cortex of the human brain. The architecture allows CNN to extract high-level features like edges from the input image. [4]

Most Music Genre Classification Systems work by, firstly, extracting features from audio signals and then performing feature selection, and lastly applying machine learning and deep learning methods for classification. Feature selection is done to reduce the dimensionality of the data so that it can be processed faster and more accurately by the classifier. This is done by mel frequency cepstral coefficients (MFCCs), which are the set of features that help describe the overall shape of a spectral envelope. Therefore, each audio file is converted into visual mel spectrograms and then split into training and testing sets. Since a mel spectrogram is a visual representation of the audio with respect to the frequency and time, Convolutional Neural Networks has the perfect architecture, which is a grid like topology to use these multidimensional vectors such as images, since they have additional layers for edge detection. [5]

DATA COLLECTION

The dataset we have decided to use is the GTZAN dataset. It consists of 10 genres ranging from classical, pop, metal, blues, etc. having 100 audio files (in.wav format) 30 seconds long. All total 1,000 audio files also have their respective visual representations in the form of Mel-spectrograms. There are also 2 .csv files, one of which has the mean and variance of multiple features for each audio file. The other file contains the same data but for all audio files after they get split into 10 3--second-long files, thus having 10,000 files in total. This helps the CNN during feature extraction because the more data it has, the better it can classify.

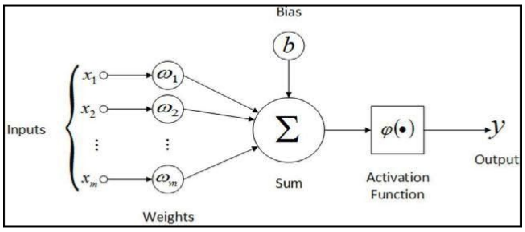
Since the Mel-spectrograms for 10,000 audio files take a day to generate, we will be using one-eight of the dataset. According to other people who have used GTZAN, the jazz genre seems to have error files so we will be skipping that. So, we are left with 9 genres, each having 13 audio files. After being split, we will obtain 130 audio files for each genre, bringing the total to 1,170 files.

II. Background Theory

First and foremost, we need to have a basic understanding of what a Neural Network is, it's composition and the different architectures.

A. Understanding Neural Networks

A Neural Network (NN) acts as the backbone for many pattern detecting and classification applications. It is often said to resemble the workings of neurons in brains. There are 3 necessary components of a NN, namely, neurons, weights and an activation function. Neurons are nodes that receive input and send output to other neurons. Its input can be data like pictures or audio snippets or the output of other neurons from a previous layer. The output of a neuron is simply its weight multiplied by the input value. Finally, the bias term is added to the weighted summation of all neuron inputs.



The neurons are all interconnected, with each line of connection having a weight assigned to it. Weights depict the significance an input will have on the output. A near 0 weight

It also condenses the inputs into a certain range depending on a certain threshold. For example, in tanh activation function, its range is [-1, 1] and in sigmoid function, it's [0, 1]. A NN can also have multiple layers in between the input and output layer. Such a network is called multi layered NN and the extra layers are referred as "hidden" layers. A NN simply just having the input and output layer is called a single layered NN. A multi layered NN introduces more parameters, thus bringing more non-linearity. This means that not only can the network handle more complex functions, but it also shows more accuracy during the training process. [6]

B. Different types of NN

Authors of research papers [7, 8, 9] have experimented with different types of NN to see which one suits music genre classification the best. The NNs discussed are as follows: Convolutional NN (CNN), Recurrent NN (RNN), Long Short-Term Memory RNN (LSTM) and Multi-Layer Perceptron NN (MLPN). RNN are trained via backpropagation but sometimes the gradients obtained during this can either tend to 0 (vanishing gradient) or infinity (exploding gradient). Each of the NN's inputs are to be updated with the gradient with respect to the inputs' original weights. But a gradient of 0 would mean no update and a gradient of infinity would mean an exponentially large update. These issues can be enough to halt the training process altogether which is when LSTM comes in.

means little to no change on the output. Initially during the training process, the weights are chosen at random. They are adjusted later on according to the output they give. The ultimate goal is to choose weights such that there is very little difference between the predicted and actual output. An activation function is a mathematical equation that decides whether to signal (activate) the neuron or not. The LSTM architecture consists of a cell that can retain information at regular intervals and 3 types of cells, input, output and forget. These cells control flow of information to and from the cell. When LSTM units are paired up with RNN, it is able to somewhat avoid the vanishing gradient problem. This is because the derivative of a hidden state in RNN contains a factor responsible for the abnormality. In LSTM, the derivative of the cell state, that factor isn't present and it makes sure to have 1 path where the gradient cannot vanish. However, this fails to counter the exploding gradient problem. [10]

MLP also uses backpropagation for training. Along with the gradient issues discussed above, a good learning rate is also to be considered. High learning rates allows for a faster learning pace at the cost of getting subpar weights. Smaller learning rates can obtain optimal waits but causes the system to take longer to train. [11]

CNN is composed of an input layer, hidden layers and an output layer. The hidden layers include convolution, pooling and receptive layers. In the convolution layers, the input goes through a set of filters and the output is sent to the pooling layers which reduce the dimensionality of the data thus helping in reducing processing time. Lastly, the fully connected layer connects each neuron in a layer to all neurons in another layer. By using regularized weights, they $= \phi$ m

(bias + $\sum w_i x_i$)$_{i=1}$network is able to avoid the gradient issues. [12] Thus, we can safely conclude that CNN is the optimal network for

music genre classification.

III. Issues Involved in Existing Work

Genre is defined as a type of music that has a certain style. However, with the rise of many new genres, its definition has become vague and inconsistent. This is because there are genres that have evolved, genres whose features tend to overlap with one other and genres that vary by just a small margin. Moreover, there is no classification system up-to-date that encompasses all of them. Most people developing or experimenting about such a system only account for a few popular ones like Pop, Hip-Hop, R&B, Rock, etc. Their lesser-known counterparts or genres exclusive to a certain country don't get considered. This means that sometimes, the system cannot classify a song properly and might be forced to classify it based on its closest similarity with the genres it knows. There also comes the issue that some misclassifications do more damage than others. For example, classifying Jazz as Blues is still better than classifying it as R&B. These varying degrees of misclassification needs to be accounted for via penalization schemes during the training process in order to better the system's quality. These constraints greatly reduce the system's accuracy and limit its potential. The most overall successful system had a 75% accuracy when classifying 10 genres. [13]

Most published works also make the assumption that the entirety of the song belongs to 1 genre. Sometimes, the sections of a song differ from the genre it is labelled as. The band Queen's Bohemian Rhapsody is one such example.

While it’s labelled as Rock, it has a Ballad intro, an Opera passage and a Hard Rock coda. Therefore, a better approach would be to consider the structure of the song and classify that section accordingly. In this way, sections like the intro, hook, chorus, bridge, etc. can be tagged correctly resulting in multiple genres. But then comes another problem. Trying to average all these genres can be tricky because sometimes they can be so distinct from each other that hardly any of their features overlap. [14]

Another issue among previous works [7, 15, 16] is their input audio length. The sample lengths range from 1, 3, 5, 30 seconds to the entirety of the song itself. This brings about an inconsistency in the input parameter and can thus affect the final result. A necessary step genre during classification is feature extraction, a process where the input data is combed thoroughly to extract only its key features. Naturally, a system will be able to conduct efficient feature extraction from a longer sample than a shorter one. We have already discussed the possibilities of misclassification due to multiple genres’ similarities. So, if a system has only a short 1 second sample length, the chances of misclassifying it can be very high. [7] has that sample length and the highest accuracy it could from that even after applying the best pre-processing methods was a mere 48%

However, there is also an inconsistency about the number and type of features considered for the feature extraction process. For [8], there are 8 in total: Beats per Minute (bpm), Discrete Wavelet Transform (DWT), valence, loudness, acoustic-ness, danceability, energy and voice content. [16] also has 6: 0 crossing rate, spectral bandwidth, centroid, contrast, roll-off and Mel Frequency Cepstral Coefficient (MFCC). Whereas in [15], there are only 5: timbral texture, timbral texture vector, rhythmic contents, pitch content and whole file/real-time features. The system uses the features extracted from this process as a basis for evaluation during the training process. This means that the system is trained differently based on the kind of features and how many there are. Obviously, more features mean more parameters the system can work with, meaning more accurate results. Moreover, one combination of features might be able to train the system better than the others. And so, there needs to be more research done on this so that future systems can have an improved training process.

The more technical types of issues are described as follows:

`1) Training and Testing Process

[15] put aside 90% of the data for training and 10% for testing. Whereas, [7] used only 50% for training, 20% for testing and 30% for holdout. There are usually 2 problems to look out for, that you neither have too little training or testing data. The former means that the parameter estimates have a larger variance and the latter means the performance statistic will have larger variance. So, special care needs to be taken to ensure there is a good training-testing split.

2) Types of Classifiers

Some papers tested different types of classifiers to see which one gave the highest accuracy. In [17], those are J48, 3NN (Nearest Neighbor), Support Vector Machine (SVM), Multi-Layer Perceptron (MLP) and Naïve Bayes. [16] chose k-NN, Naïve Bayes, SVM, Random Forest and Decision Tree. While both papers found SVM to be the most accurate, their feature extractions did differ. If they had the same feature extraction procedure, there could be some change in the accuracy of the other classifiers.

3) Performance Metric

Most authors [7, 17] have used only 1 metric for measuring performance, namely accuracy. [15] used accuracy mean and standard deviation and [8] used a confusion matrix. In [16], the authors took to 4 metrics, accuracy, precision, recall and F-Measure. Having more than 1 metric will help us in understanding the performance of a system better and find out in which areas it falls short of because different metrics depend upon different factors to calculate performance.

IV. Literature Survey

Many researchers have worked on musical parameters and the ways to classify them, out of which some of the works have been discussed in this section. S. Vishnupriya and K. Meenakshi [18] classified music using Convolutional Neural Network (CNN) They used the Million Song Dataset (MSD) in their music genre classification model. After creating the database, they used the librosa package in python for feature extraction. The extracted feature is called as MFCCs which encodes the timbral properties of the music signal. They apply the Fourier Transform to get the frequency spectrum of the music signal. The three layers used are Convolutional Layer, Pooling Layer and Fully connected Layer. Using MFCCs and Mel spectrum the classification is promising even for a large database.

In [19], Spectrograms and Convolutional Neural Networks (CNN) are used for the classification. To show that the Time Frequency analysis gives better results than the use of Hand-Crafted features, a comparative study is done on three datasets namely a collection of western music (ISMIR), a collection of ethnic African music and a collection of Latin music (LMD). The results obtained from the study show that CNN works better for western and Latin music datasets with a recognition rate of 92%.

Rajanna et al. [20], compared the classification accuracy rate of Deep Neural Networks (DNN) with Support Vector Machines (SVM), logistic regression and hand-crafted features. A two-layer neural network has been used for the comparison and the results show that DNN works as good as the other well-known machine learning models when represented in a rich feature space.

In [21], authors classified the GTZAN data set by using two CNNs. The first CNN is fed a Fourier Transform of the audio signal and the output is fed into another CNN.

The network topologies of the two CNN's are analyzed in order to arrive at the results. They combined max- and average pooling to improve the performance. The use of shortcut connections in their work helped in decreasing the number of hidden layers in the neural network. However, the work is not completely based on CNNs because they have also used spectrogram which is a hand-crafted feature.

Another such work done by Senac and Christine [22] analyzes the relationship between spectrogram and Artificial Neural Network (ANN). Works done in the field shows that an ANN learns in the presence of a spectrogram. To avoid the use of spectrograms they do the classification based on tonality, dynamics, and timbre for a smaller database, and check for the ANNs classification accuracy rate. In the absence of spectrogram an ANN does perform better with just 8 music features and shows 91% accuracy for the GTZAN dataset.

To improve the music feature learning rate, Sigtia, Siddharth, and Simon Dixon [23] proposed three methods in their paper. They also did a comparative of neural networks, sigmoid networks and hand-crafted features. Rectified Linear Units (ReLUs) are applied instead of the traditional sigmoid networks, Hessian Free (HFF) for training the network and Dropout for regularization. The research paper claims that ReLUs learning rate is better than sigmoid networks.

Yang et al. [24] proposed a CNN model with more duplicated convolutional layers. The purpose of doing so is to increase the efficiency of classification. The results show that duplication of layers in the CNN behaves differently for every class of genres which implies that network architectures play an important role in classification. However, the reason behind the behavior of CNN in classification of genres has not been studied in the research work. An assisting Recurrent Neural Network (RNN) is needed for end-to-end learning to avoid the use of a spectrogram for feature extraction.

In [25], a parallel recurrent convolutional neural network (PRCNN) is used for music genre classification in mobile phones. This is an end-to-end learning network that combines time series models and feature extraction. The temporal frame orders are designed using a Bi-directional Recurrent Neural Network (Bi-RNN). The lost features in CNN are obtained through Bi-RNN. SoftMax function is used for the classification of data and an assisting RNN is also used. This method outperforms all the traditional models used for music genre classification. It is also found that CNN has better efficiency in the presence of an assisting 1-layer RNN instead of a 2-layer RNN. The performance of PRCNN is verified by implementing it on the GTZAN and Extended Ballroom datasets.

Wu, Wenli, et al. [26] proposed a model that classifies music genres using Independent Recurrent Neural Network (IndRNN). This training model is useful in case of Long-term relationship as compared to Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM). Scattering Transform is used to pre-process the raw data and to control the feature loss. The computational efficiency of IndRNN is proven to be better than the state of art models.

Kour, Gursimran, and Neha Mehan [27] combined the features of Mel-frequency Cepstral Coefficients (MFCC), Support Vector Machine (SVM) and Back Propagation Neural Network (BPNN). Acoustic features are extracted from the audio signal and MFCC identifies the structure and frequency of these signals. Then, the data is classified using SVM and BPNN. The accuracy rate of these classifiers is found by simulating the model in MATLAB. BPNN shows an accuracy rate of 95% while SVM's accuracy rate is 83%.

In [28], authors use a Bottom-up Broadcast Neural Network that identifies the genre of the unknown audio signal with the help of multi-scale time-frequency information. The datasets used are Ballroom, Extended Ballroom and GTZAN with Extended Ballroom having the highest classification accuracy rate of 97.2% followed by Ballroom and GTZAN with 96% and 93% respectively. The proposed model only works for devices with limited computational resources.

Ghosal et al. [29] proposed a novel approach that combines transfer learning and Convolutional Long Short-Term Memory based Neural Network (CNN LSTM) that recognizes the different music genres. Averaged Signal Vector and Music Transfer Learning Vector are used for summary statistics of the extracted features. This recognition model is basically a voting system that assigns genres according to the frequency and is finalized using the SoftMax function. The model achieves a near perfect score for the accuracy rate.

Ghosal, Soumya Suvra, and Indranil Sarkar [30] using the temporal features with the help of CNN and LSTM to sequence the autoencoders. They also introduced Clustering Augmented Learning Method Classifier (CALM) that aggregates the data having similar features by which music genres classification is achieved. The proposed model has an accuracy rate of 95%.

In [31], a comparison between traditional and relational approaches to music genre classification is done. The research work claims that the structural features offer more statistical information than the automated music genre classifiers and that relational models can work on graph models.

Lee et al. [32] classified the audio signals into 8 genres. A Time Delay Neural Network (TDNN) which takes Fourier Transformed vectors of the signal is designed. The features are extracted by the sound of a snare drum. However, the TDNN model proposed has a long training time but a short execution time and is promising for very large datasets.

VI. DISCUSSION

Firstly, music genre classification is done using CNNs and MFCC (Mel Frequency Cepstral Coefficients). Convolutional Neural Network is a deep learning algorithm which can take in an input image, assign weights and biases to various aspects in the image and be able to differentiate one image from the other. We have chosen Convolutional Neural Network, because it requires much lower pre-processing than other classification algorithms. Mel frequency, on the other hand, is a measure of the height of sound obtained from human perception characteristics. MFCCs are all the points that collectively make up an MFC.

To implement this, firstly, as per the file extract_data.py, we define a high level function *save_mfcc(),* which is used to extract the MFCCs from the music dataset and save them into a json file. Then, we build a dictionary to store data such as mapping, MFCC vectors for each of the audio segments, and the labels (or the target output) for the MFCCs. We then loop through all the genre folders and analyse each song one by one. The *os.walk()* method is used to do this.

In the mappings, we have to save the path of the current directory, specifically the name of the genre folder. Then, we go through all the files in the specific genre folder and load the audio files in it, *librosa.load()* is used to load the sample file. We then process the segments by extracting the MFCCs by going through all the segments. We get the MFCC by using the *librosa.feature.mfcc()* function.

The number of samples per track is given by the sample rate multiplied by the duration of each audio file. The number of samples for each segment is given by the number of samples per track divided by the number of segments. This is the logic that is used to define the start sample and the finish sample for extracting the MFCC. Then, we store the MFCC for the segment if it has the expected length of vectors. This expected length of vectors per segment is calculated by dividing the number of samples per segment by hop length. Thus we store the MFCC and labels for each segment. We finally save all the dictionary data (mapping, labels, MFCC) into the json file.

In the file cnn_genre_classfier.py, the first thing we do is load the data from lists to numpy arrays. The inputs for the CNN are the MFCCs and the target is the labels, which is the genres. The next thing we do is to split the data into training, validation and testing sets, using the *prepare_dataset()* function where the test size is 0.1 and validation size is 0.2. Then, we convert all the X training, testing and validation sets to a 4D array, since for a CNN, Tensorflow expects a 4D array to use *model.predict()* method. Then, we build the CNN network with three ReLu Convolutional layers, flatten the output, feed this to the dense layer and then add an output layer or Softmax. We then compile the neural network with a learning rate of 0.0005 and train the CNN by fitting the training and validation data with a batch size of 32 and Epochs of 30. Then we evaluate the CNN on the test set using *model.evaluate()* method. Finally, we make predictions on a simple sample and plot the accuracy and error graphs. We have obtained an accuracy of 88%.

For CNN implementation with Mel-spectrograms, we have made use of the python addon named kaggle. This allows us to download and unzip the GTZAN dataset and arrange the dataset's contents as files.without us having to manually download and categorize them. Then, we define the genre list and use that to make genre folders under the *audio3sec* directory. We make the *spectrograms3sec* directory where we will be storing all our spectrograms later on . Since we have only 13 audio files to work with from each genre, we decided to split it into 10 parts each, thus each genre now has 130 audio files which are 3 seconds long. All such files are stored under each genre folder in the *audio3sec* directory. Next, we define train and test folders under *spectrograms3sec* and make all 9 genre folders under both. We make Mel-spectrograms of each audio file present in all genre folders in *audio3sec* directory and store them in train folder under *spectrograms3sec* directory. After that, we randomly choose 13 spectrograms from each genre and place them in the test folder under the same genre. *ImageDataGenerator()* is a function which takes the spectrograms and rescales them all to the same range so that it could lead to low loss. *GenreModel()* builds the CNN model where each layer's activation function is Rectified Linear Unit (ReLU) and the final flattened layer has softmax activation function. In order to train the model, we require F1-score metrics. *get_f1()* measures true positives, precision and recall and substitutes them in the F1-score equation and returns that value. We fix the learning rate and performance metrics to be accuracy and F1-score. Lastly, *GenreModel()* is called and we configure the model to show accuracy, loss and F1-score for training and validation data for 30 epochs. We got 91.21% accuracy, which is an improvement to the previous attempt using MFCCs.

VII RESULTS:

In this paper, we've presented two models that classify the music into the best possible genres. The first model uses the GTZAN database to train 30% of the data and test the rest 70% of the data using Mel Frequency Cepstral Coefficients(MFCC) and build a CNN model. The second model makes use of the mel spectrogram and a CNN model is built later to classify the audio files into different genres.

We've chosen 1/8th of the dataset to make the training process faster. The accuracy of both the models are analysed.

The accuracy and validation accuracy for the MFCC model is presented in Table 1.

0.1771	0.15171
0.5616	0.1754
0.6933	0.1991
0.7246	0.2701
0.7896	0.4692
0.8371	0.6066
0.8637	0.7536
0.8648	0.7441
0.8838	0.7915
0.8899	0.8057
0.9028	0.8294
0.9301	0.8246
0.9282	0.8246
0.9350	0.8341
0.9271	0.8626
0.9464	0.8673

0.9514	0.8341
0.9516	0.8294
0.9697	0.8531
0.9660	0.8578
0.9712	0.8436
0.9566	0.8531
0.9658	0.8341
0.9619	0.8389
0.9810	0.8531
0.9873	0.8626
0.9603	0.8531
0.9710	0.8626

Table 1. Accuracy and Validation accuracy for MFCC model

The overall accuracy obtained from this model is 88.03% , validation accuracy is 86.26%.

The accuracy and validation accuracy for the Mel Spectrogram CNN model is presented in Table 2.

Accuracy	Val Accuracy
0.1061	0.1111
0.1198	0.1111
0.1783	0.1282
0.2553	0.1197
0.3603	0.1111
0.3379	0.1111
0.3910	0.1111
0.4518	0.1111
0.4647	0.1709
0.4781	0.1282
0.5382	0.1880
0.4951	0.1111
0.5690	0.1111
0.5651	0.1966
0.5972	0.1111
0.5935	0.1111
0.6437	0.1111
0.6867	0.1111
0.7286	0.1111
0.7391	0.3248
0.7799	0.2906
0.7434	0.1709
0.7536	0.1111
0.7887	0.1624
0.7850	0.1795
0.8362	0.2222
0.8506	0.1197
0.8707	0.1538
0.8276	0.2051

Table 2. Accuracy and Validation accuracy for the Mel Spectrogram model

The overall accuracy and validation accuracy by the above model is 91.21% and 23.93% respectively. From the above results we can see that the accuracy has improved by 3.18% .

To get a better idea about the models we plotted the accuracy and the error rate graph

for the MFCC model. Figure 1 shows the accuracy rate for MFCC model.

Figure 1. Graph for accuracy rate comparison for the MFCC model.

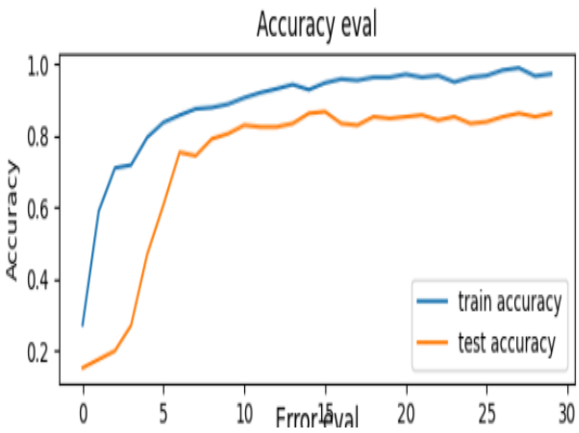
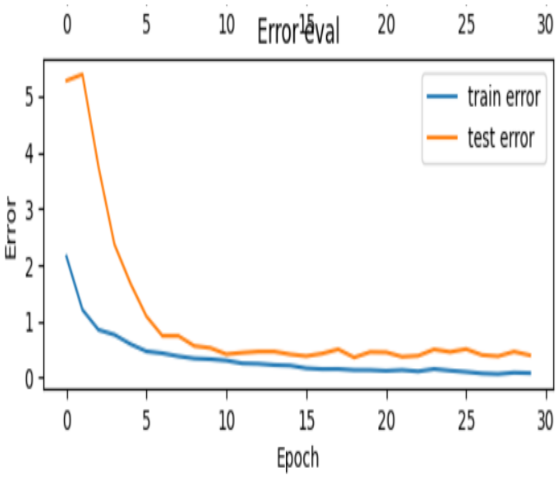


Figure 2. Graph for error rate comparison of the test and train data of the MFCC model.



The overall loss for the MFCC and Mel Spectrogram models are 0.0793 and 0.3764.

VIII. Conclusion

This paper presented the survey of different Neural Networks (NN) used by researchers for Music Genre Classification. Most of the work done uses the GTZAN and Extended Ballroom datasets. Convolutional Neural Networks (CNNs) provide the base for the music genre classification in most of the cases. It is understood that CNNs alone cannot give a good accuracy rate but can achieve a near perfect score when used along with other neural networks. Some of the models that help these CNNs to perform better are Recurrent Neural Networks (RNNs), Transfer Learning, Long Short-Term Memory (LSTM) and Spectrogram. It is evident that the accuracy rate is higher for a smaller database but can be improved with the help of multi-layer neural networks.

At first, the code made use of the CNN model with MFCCs. Since we have considered only 13 audio files, the model has very little data to work with and the accuracy could worsen. To combat that, we have modified the code to split each audio file into 10 parts. This will give us 130 audio files for each genre and thus, more data. After training the model, setting the appropriate parameters and metrics, the accuracy obtained was 88.03%. In order to gain a better value, we thought about feeding Mel-spectrograms of the audio files as the CNN's input. We made sure that we took the same parameters as before and trained the model with the new method, gaining an improved accuracy score of 91.21%. Thus, we can conclude that MFCC provides a compressed representation whereas Mel-spectrograms work better for large datasets and strong classifiers like CNN.

IX. .CONTRIBUTION

Topic with authors' names and affiliations, Abstract, Keywords, Introduction, Discussion, the code for CNN with MFCC, video explanation and part of the References done by Meriam Joseph. Data Collection, Background Theory, Issues Involved in Existing Works, part of Conclusion, part of the code for CNN with Mel-spectrograms, video explanation and part of the References done by Tehsin Sherasiya. Literature Survey, Results, part of Conclusion, part of the code for CNN with Mel-spectrograms, video explanation and part of the References done by Shruthi Vandhana V.

REFERENCES

1. Senac, C., Pellegrini, T., Mouret, F., & Pinquier, J. (2017). Music feature maps with convolutional neural networks for music genre classification. Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing. doi:10.1145/3095713.3095733

2. A. Ghildiyal, K. Singh and S. Sharma, "Music Genre Classification using Machine Learning," 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2020, pp. 1368-1372, doi: 10.1109/ICECA49313.2020.9297444.

3. N. Pelchat and C. M. Gelowitz, "Neural Network Music Genre Classification," in Canadian Journal of Electrical and Computer Engineering, vol. 43, no. 3, pp. 170-173, Summer 2020, doi: 10.1109/CJECE.2020.2970144.

4. Saha, Sumit. "A Comprehensive Guide to Convolutional Neural Networks — the ELI5 Way." Medium, 15 Oct. 2020, towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a55.

5. Roberts, Leland. "Musical Genre Classification with Convolutional Neural Networks." Medium, 16 Mar. 2020, towardsdatascience.com/musical-genre-classification-with-convolutional-neural-networks-ff04f9601a74.

6. Baeldung. "Advantages and Disadvantages of Neural Networks." Baeldung on Computer Science, 26 June 2020, www.baeldung.com/cs/neural-net-advantages-disadvantages.

13. https://www.music-ir.org/mirex/abstracts/2005/lidy.pdf Accessed 3 Mar. 2021

7. Jingqing, Yang. "Music Genre Classification With Neural Networks: An Examination Of Several Impactful Variables." Digital Commons, May 2018, http://digitalcommons.trinity.edu/cgi/viewcontent.cgi?article =1044&context=compsci_honors. Accessed 3 Mar. 2021

8. Gessle, Gabriel, Åkesson, Simon. "A Comparative Analysis of CNN and LSTM for Music Genre Classification." DiVa, June 2019, www.diva-portal.org/smash/get/diva2:1354738/FULLTEXT01.pdf. Accessed 3 Mar. 2021

9. Masood, Sarfaraz. "Genre Classification of Songs Using Neural Network." Research Gate, Sept. 2014, www.researchgate.net/publication/280565926_Genre_classification_of_songs_using_neural_network. Accessed 3 Mar. 2021

10. Sutskever, Ilya. "Sequence to Sequence Learning with Neural Networks." ArXiv, 10 Sept. 2014, arxiv.org/abs/1409.3215. Accessed 3 Mar. 2021

11. "Why MultiLayer Perceptron/Neural Network?" MIT Media Lab, 3 Mar. 2021.

12. Glorot Xavier, Bengio Yoshua. "Understanding the Difficulty of Training Deep Feedforward Neural Networks." Proceedings of Machine Learning Research, proceedings.mlr.press/v9/glorot10a/glorot10a.pdf. Accessed 3 Mar. 2021.

13. Lidy, Thomas and Rauber, Andreas. "COMBINED FLUCTUATION FEATURES FOR MUSIC GENRE CLASSIFICATION." MIREX, 2005, www.music-ir.org/mirex/abstracts/2005/lidy.pdf. Accessed 3 Mar. 2021.

14. Fujinaga, Ichiro. "Musical Genre Classification: Is It Worth Pursuing and How Can It Be Improved?" Research Gate, Jan. 2006, www.researchgate.net/publication/200688634_Musical_Genre_Classification_Is_It_Worth_Pursuing_and_How_Can_It_be_Improved.

15. Tzanetakis, George and Cook, Perry. "Musical Genre Classification of Audio Signals." Research Gate, Aug. 2002, www.researchgate.net/publication/3333877_Musical_Genre_Classification_of_Audio_Signals.

16. Elbir, Ahmet and Bilal Hilmi. "Music Genre Classification and Recommendation by Using Machine Learning Techniques." Research Gate, Oct. 2018, www.researchgate.net/publication/329396097_Music_Genre Classification_and_Recommendation_by_Using_Machine Learning_Techniques.

17. Silla, Carlos and Kaestner, Celso. "Automatic Music Genre Classification Using Ensemble of Classifiers." CORE, core.ac.uk/download/pdf/13671.pdf. Accessed 3 Mar. 2021.

18. S. Vishnupriya and K. Meenakshi, "Automatic Music Genre Classification using Convolutional Neural Network," 2018 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2018, pp. 1-4, doi: 10.1109/ICCCI.2018.8441340.

19. Costa, Yandre MG, Luiz S. Oliveira, and Carlos N. Silla Jr. "An evaluation of convolutional neural networks for music classification using spectrograms." Applied soft computing 52 (2017): 28-38.

20. Rajanna, Arjun Raj, et al. "Deep neural networks: A case study for music genre classification." 2015

IEEE 14th international conference on machine learning and applications (ICMLA). IEEE, 2015.

21. Zhang, Weibin, et al. "Improved Music Genre Classification with Convolutional Neural Networks." Interspeech. 2016.

22. Senac, Christine, et al. "Music feature maps

with convolutional neural networks for music genre classification." Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing. 2017.

23. Sigtia, Siddharth, and Simon Dixon. "Improved music feature learning with deep neural networks." 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2014.

24. Yang, Hansi, and Wei-Qiang Zhang. "Music Genre Classification Using Duplicated Convolutional Layers in Neural Networks." INTERSPEECH. 2019.

25. Yang, Rui, et al. "Parallel recurrent convolutional neural networks-based music genre classification method for mobile devices." IEEE Access 8 (2020): 19629-19637.

26. Wu, Wenli, et al. "Music genre classification using independent recurrent neural networks." 2018 Chinese Automation Congress (CAC). IEEE, 2018.

27. Kour, Gursimran, and Neha Mehan. "Music genre classification using MFCC, SVM and BPNN." International Journal of Computer Applications 112.6 (2015).

28. Liu, Caifeng, et al. "Bottom-up broadcast neural network for music genre classification." Multimedia Tools and Applications (2020): 1-19.

29. Ghosal, Deepanway, and Maheshkumar H. Kolekar. "Music Genre Recognition Using Deep Neural Networks and Transfer Learning." Interspeech. 2018.

30. Ghosal, Soumya Suvra, and Indranil Sarkar. "Novel Approach to Music Genre Classification using Clustering Augmented Learning Method (CALM)." AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering (1). 2020.

31. Valverde-Rebaza, Jorge, et al. "Music genre classification using traditional and relational approaches." 2014 Brazilian Conference on Intelligent Systems. IEEE, 2014.

32. Lee, Jae-Won, Soo-Beom Park, and Sang-Kyoon Kim. "Music genre classification using a time-delay neural network." International Symposium on Neural Networks. Springer, Berlin, Heidelberg, 2006