

Music Genre Classification with Convolutional Neural Network

by

Masud Chowdhury

17101323

Ibnul Islam Tilok

17201058

Prodipta Das

17201059

Avoy Chowdhury

17101409

MD. Abdullah Al Masum Anas

20141046

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
January 2022

© 2022. Brac University
All rights reserved.

Declaration

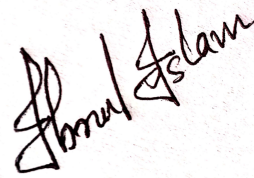
It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:



Masud Chowdhury
17101323



Ibnul Islam Tilok
17201058



Prodipta Das
17201059



Avoy Chowdhury
17101409



MD. Abdullah Al Masum Anas
20141046

Approval

The thesis titled “Music Genre Classification with Deep Learning” submitted by

1. Masud Chowdhury(17101323)
2. Ibnul Islam Tilok(17201058)
3. Prodipta Das(17201059)
4. Avoy Chowdhury(17101409)
5. MD. Abdullah Al Masum Anas(20141046)

Of Fall, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 16, 2022.

Examining Committee:

Supervisor:



Moin Mostakim
Lecturer
Department of Computer Science and Engineering
Brac University

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.

Keywords: Music Genre, CNN, Classification, Feature extraction, Accuracy

Acknowledgement

Firstly, we have completed our thesis without any significant interruption by the Grace of Almighty Allah.

Secondly, Thanks to our Advisor Mr. Moin Mostakim, for his generous support and guidance in our work. He supported us whenever we needed help.

Finally, We would like to acknowledge with gratefulness, The support and love of our Parents throughout the Journey by keeping us in their prayers. With their kind support and prayer, we are now on the verge of graduation.

Table of Contents

Declaration	1
Approval	2
Abstract	3
Acknowledgment	4
Table of Contents	5
List of Figures	7
List of Tables	8
1 Introduction	9
2 Related Work	10
2.1 Literature Review	11
3 Data Pre-processing and Feature Extraction Method	12
3.1 Dataset	12
3.2 Data Prepossessing	12
3.3 Feature Extraction	12
3.4 Feature Extraction Technique	15
4 Convolutinal Neural Network Analysis	17
4.1 CNN Model Architecture	17
5 Model Aanlysis and Classification	19
5.1 Sequential Model and CNN Classification with CSV	19
5.2 Sequential Model and CNN Classification with JSON	20
5.3 K-Means Clustering with PCA	21
5.4 Recurrent Neural Network-Long Short-Term Memory Model	22
6 Result and Analysis	24
6.1 Model Summary	24
6.2 Result Analysis	24

7	Conclusion and Future Work	26
7.1	Conclusion	26
7.2	Future Work	26

List of Figures

3.1	Spectrogram	13
3.2	Mel-Spectrogram	13
3.3	Zero Crossing Rate	14
3.4	Spectral- centroid	14
3.5	MFCC	14
3.6	Spectral Contast	15
3.7	Spectral Bandwidth	15
3.8	Feature Extraction in JSON file	16
3.9	Feature Extraction in CSV file	16
4.1	CNN Architecture	17
4.2	Dropout	18
5.1	Working Convolutional Neural Network Model	19
5.2	Accuracy of the model	20
5.3	Working Convolutional Neural Network Model	20
5.4	Accuracy of the model	21
5.5	K Means Clustering on Genres	22
5.6	Prediction of Genres with K-Means	22
5.7	Accuracy of the RNN-LSTM model	23

List of Tables

6.1	Sequential Model with CNN (CSV File)	24
6.2	Sequential Model with CNN (JSON File)	24

Chapter 1

Introduction

Music segmentation refers to distinct labeling music segments similar to human annotation—for example, the chorus, verse, intro, outro, and bridge in popular music. The boundary between such segments may be due to several factors, such as a change in melody or chord progression, rhythm, instrumentation, dynamics, key, or tempo. This task has generally been evaluated with two classes of metrics. The first class, boundary detection, refers to the ability of the algorithm to locate the locations of such boundaries in time. The second class, segment labeling, refers to segments where two disconnected segments have been labeled as the same or different based on their perceptual similarity. Some methods have been studied to classify music genres. Although these methods can achieve satisfactory performance on small and moderate-sized datasets, they cannot perform well when the music data is increasingly significant. 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 Convolutional Neural Network(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.

Chapter 2

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. Convolutional Neural Network(CNN) is a very popular model for genre classification among deep learning because of its high accuracy. In recent years, CNN has achieved brilliant performance in various fields. In the papers, we studied the author using the RNN-LSTM model for genre classification [8]. According to their evaluation, they found 89% accuracy. Another study presented an end-to-end strategy using 1D CNN [4]. Here the author 80% of the dataset for training for a hundred 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 [10]. Another author suggested classifying music using a different model automatically [3]. 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 due to vanishing and exploding gradients. As a result, it becomes tough for long-term dependencies.

2.1 Literature Review

Music is a significant kind of sound information, and the study of automatic music categorization is a critical field of audio classification. The music genre is among the most well-known ways to express the music substance, and it has extensive uses in music data recovery, music suggestions, and online music access. It is very challenging to overlook the tunes they prefer listening to as there is a noticeable improvement of the music databases online and music substance's easy access. Songs can be organized and classified based on the genre, and the characterization can be done by substance, instrumentation, and rhythmic structure, for recognizing music. From the beginning of modern technology, music genre classification has played a role as a prominent research sector [5]. Moreover, the author used directed machine learning models like Gaussian Blen demonstration and closest neighbor classifiers to address the issue. Three features mentioned for this particular assignment are rhythmic content, timbral structure, and pitch content. A few ways have been studied to classify the music types [11]. Although these algorithms can attain satisfactory performance on a bit of estimate and direct measure datasets, they cannot perform effectively when the amount of music information increases. According to the authors[7], they examine psychoacoustic feature participation in distinguishing music genres, precisely the value of STFT taken on the Bark Scale [9]. Mel-frequency cepstral coefficients (MFCCs), spectral contrast, and spectral roll-off were a few of the properties utilized by the authors [12]. To prepare SVM and AdaBoost classifiers, visual and acoustic features are merged. Due to the enormous volume of music data, the music genre classification using Deep Neural Network(DNN) has become a trend [6]. However, there still exists a problem. DNN or other models prepare the model with free input frames and cannot capture the transitory conditions of the audio sequence. The author stated that there is a high selection rate of the audio inputs; representing audio or music in the time domain for neural network input is not simple [1]. A spectrogram of a movement can capture time and frequency information, which can be used as an alternative representation. For developing Convolutional Neural Networks(CNN), the spectrogram is considered images and utilization of it [2]. A CNN model was constructed to predict the music genre utilizing the raw MFCC framework as input. Our research aims to give a brief comparative analysis among deep learning-based models. For that, we take spectrograms as input, and the classifiers are trained using hand-crafted features.

Chapter 3

Data Pre-processing and Feature Extraction Method

3.1 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.

3.2 Data Preprocessing

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 Comma-Separated Values(CSV) file to use the dataset. CSV files allow us to save the tabular data such as spreadsheets, which is very useful for large datasets. We used our dataset and converted it 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). We split the dataset into the training and test sets because it is necessary and 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.

3.3 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. It will require

an image as input data. Therefore, we must 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.

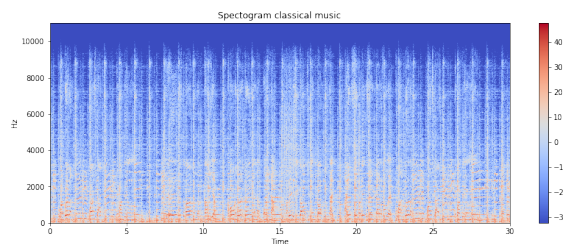


Figure 3.1: Spectrogram

Mel-Spectrogram: The Mel scale is a set of pitches that are comparable in distance to one another and maybe felt by the listener. The Fourier transform is used to convert frequencies to the Mel scale. The Mel spectrum converts frequency into Mel scale by Fourier transform.

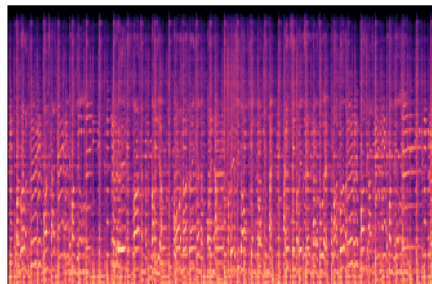


Figure 3.2: Mel-Spectrogram

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.

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

Mel frequency Cepstral Coefficients (MFCCs):The MFCC describes the shape of the spectral. We have extracted the MFCC of the audio data. The MFCC coefficients are represented in the y axis and time in the x-axis. We have used 13 strides

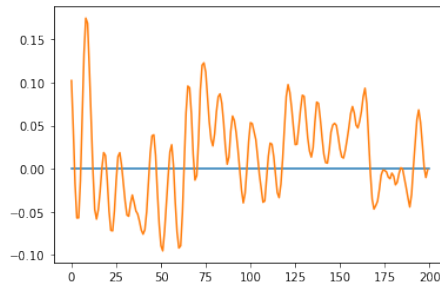


Figure 3.3: Zero Crossing Rate

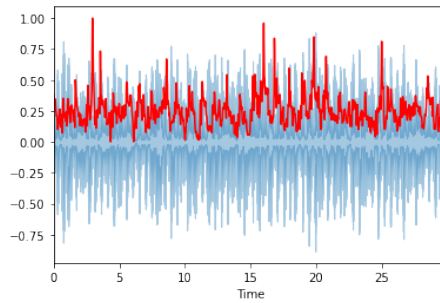


Figure 3.4: Spectral-centroid

for the JSON file. We used `librosa.feature.mfcc` to generate the MFCC.

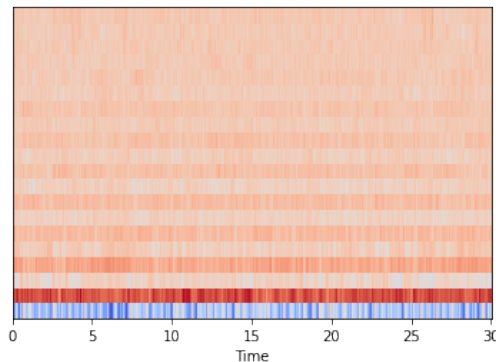


Figure 3.5: MFCC

Rolloff frequency: Spectral Rolloff measures the shape of the signal. Here it is defined as the percentage of power spectral distribution. The ratio is usually 85 – 95%. Below, the roll-off point is concentrated at 85-95% of the magnitude distribution. It is used to differentiate between noisy and harmonic sounds.

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.

Spectral bandwidth: Bandwidth helps to distinguish between the upward and downward frequency. As signals cycle around a point, and if the point is the signal's

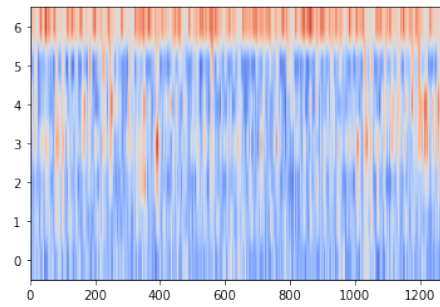


Figure 3.6: Spectral Contast

center, then bandwidth during that time frame may be calculated as the sum of the signal's maximum deviation on both sides of the point.

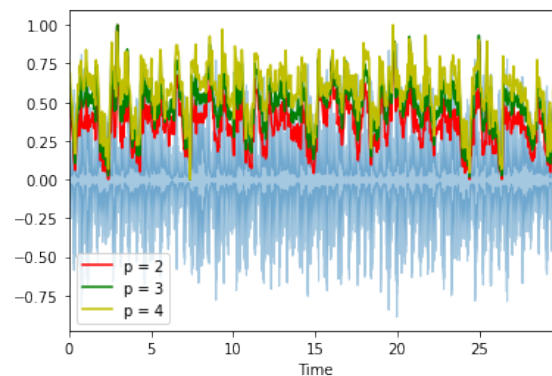


Figure 3.7: Spectral Bandwidth

Principal Component Analysis(PCA): PCA is a feature extraction technique meant to reduce the dimensions of our dataset. PCA can also be used to save computational time. The main goal of principle component analysis (PCA) is to minimize the complexity of a data set thought up of many connected variables while retaining as much variation as possible. Similarly, the variables are transformed into a new collection of variables known as the principal components. PCA is a technique for determining the relationships and dependencies between the features in a data collection. A covariance matrix shows the relationship between the various variables in the data set. Heavily dependent variables must be identified since they include biased and redundant information, lowering the model's overall performance.

3.4 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 22050. 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.

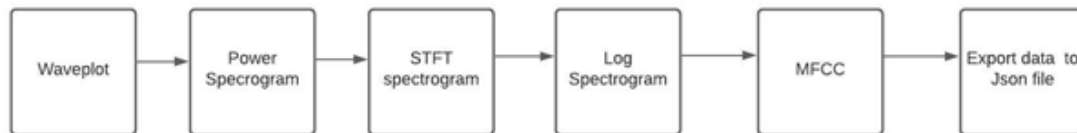


Figure 3.8: Feature Extraction in JSON file

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.

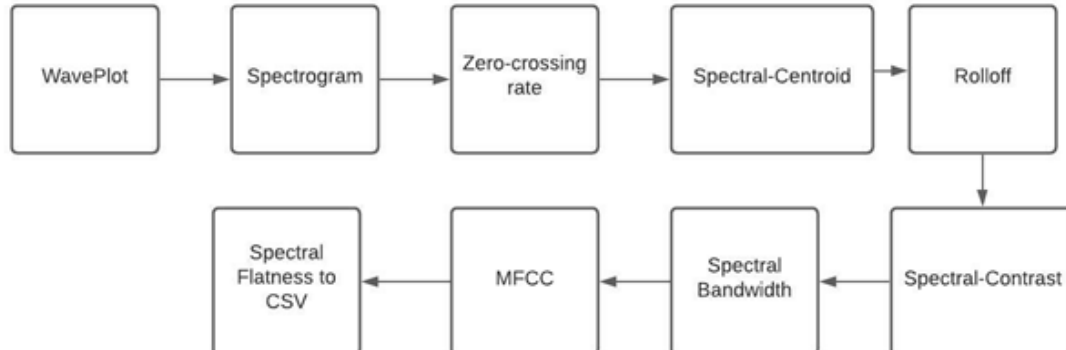


Figure 3.9: Feature Extraction in CSV file

Chapter 4

Convolutinal Neural Network Analysis

4.1 CNN Model Architecture

A convolution tool extracts and identifies the different features of 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 comprise 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.

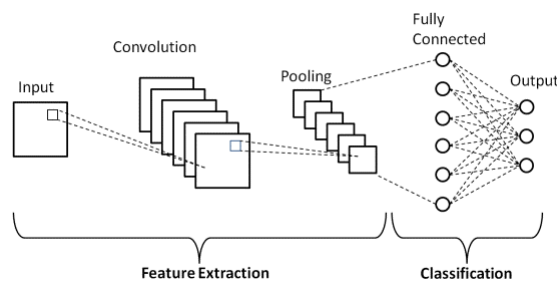


Figure 4.1: CNN Architecture

Three types of layers make up the CNN: the convolutional layers, pooling layers, and fully-connected (FC) layers. When these layers are stacked, a CNN architecture will form. In addition to these three layers: the dropout layer and the activation function.

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 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.

4. Dropout: We use dropout to prevent the overfitting while training model. What dropout does is it randomly drop neurons while training. By doing that, it helps to increase the network robustness.

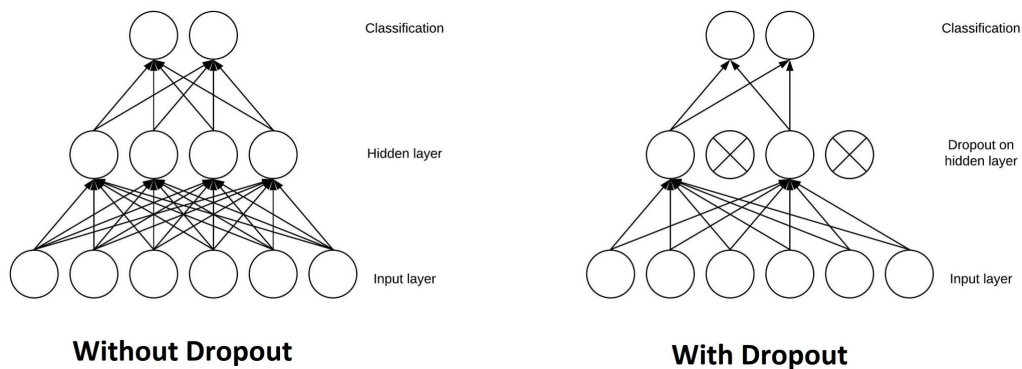


Figure 4.2: Dropout

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.

Chapter 5

Model Analysis and Classification

5.1 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.

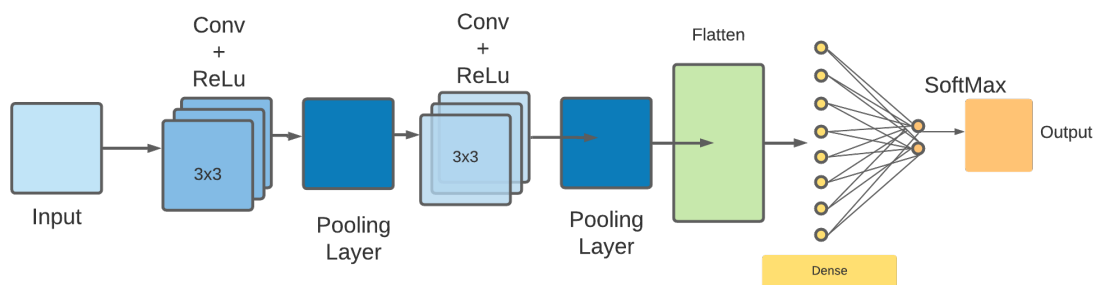


Figure 5.1: Working Convolutional Neural Network Model

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% which is the best we got compared to other models.

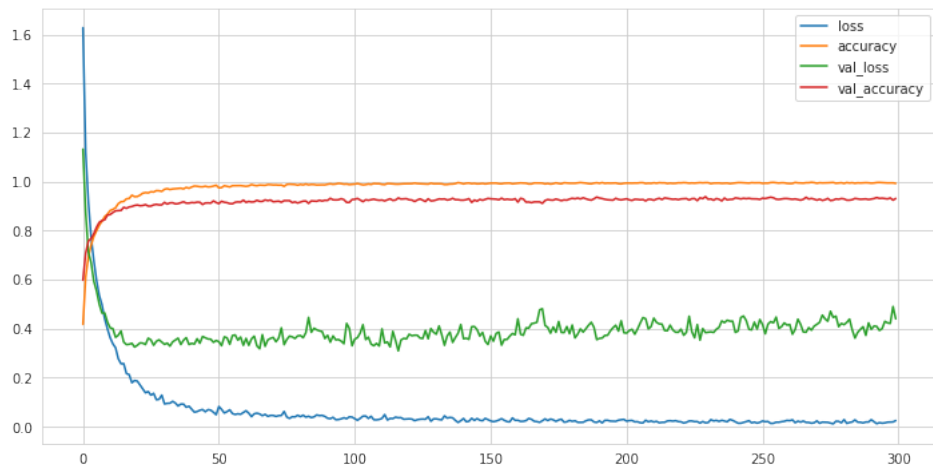


Figure 5.2: Accuracy of the model

5.2 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.

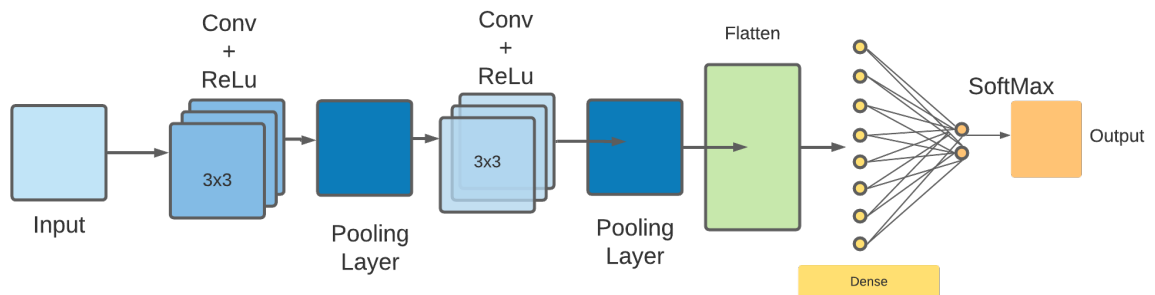


Figure 5.3: Working Convolutional Neural Network Model

The model used in the above figure is the sequential convolutional neural network architecture model. The input data is the JSON file we extracted(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 primary 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 primary 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 30% 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% accuracy for the test set.

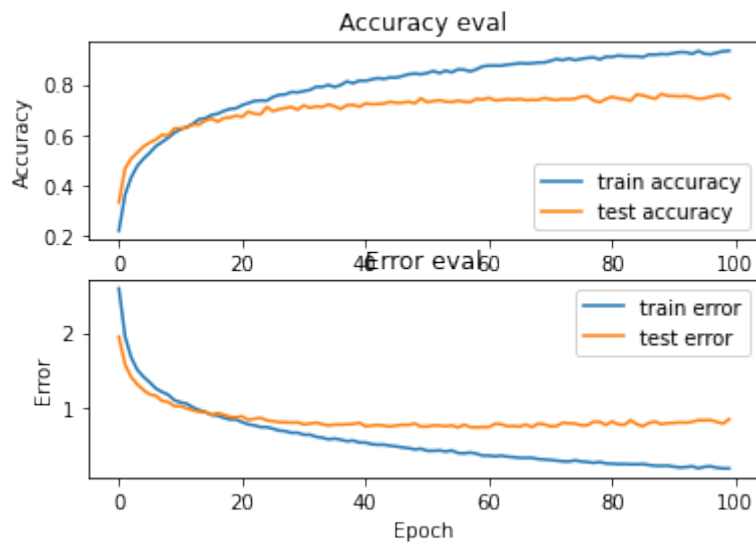


Figure 5.4: Accuracy of the model

5.3 K-Means Clustering with PCA

Clustering is a data analysis method used to intuition the data structure. It is used to identify or distinguish subgroups in a dataset so that the same subset is very similar while the data points in different clusters are different. Clustering is an unsupervised ML process used to identify subgroups of similar or dissimilar data points from a large dataset. It mainly determines the intrinsic cluster or group from the existing unlabeled data. It is used for homogeneous groups, reducing data, describing unknown properties, finding functional grouping, unusual data objects, or outlier detection. K means Clustering an unsupervised learning algorithm where the dataset will be given with certain features, and the task is to categorize the item into a group or cluster.

This model extracted our features from the audio dataset and exported them into a CSV file. Standard Scalar is performed for scaling in preprocessing. Standard Scalar is done for scaling the data so that no data can dominate other data. Then, using Label Encoder to convert the labels into a numeric form to convert them into readable to train our model. In this model, we used PCA to scale our data. PCA is a reduction method used to minimize or eliminate the dimensionality of large data

sets by transforming a large group of variables into smaller ones despite containing a maximum of information. Here due to the reduction of the data, the accuracy may decrease, but little or small data sets are very easy to explore and analyze, which makes the algorithm much faster. Therefore, in short, PCA reduces the number of dataset variables while storing the maximum data. Here we have taken three components

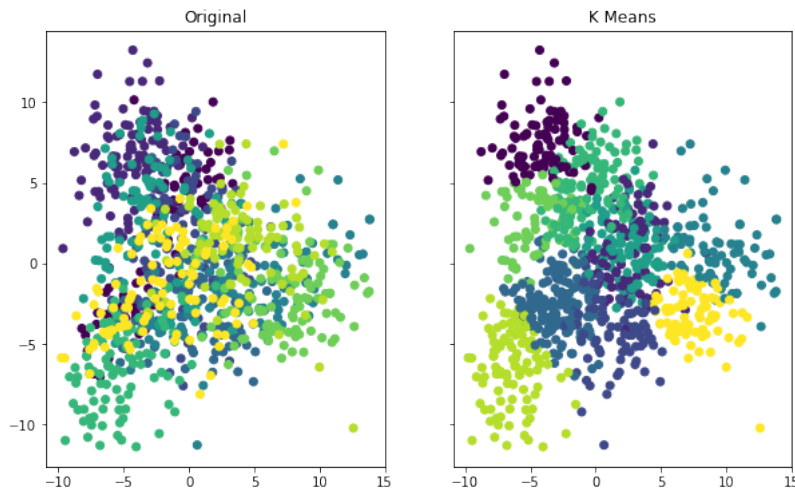


Figure 5.5: K Means Clustering on Genres

The above figure shows that the original cluster was not formed, and the plots are scattered. But, after clustering, we have found that a subgroup of clusters was formed of a different color to represent each of the individual music genres.

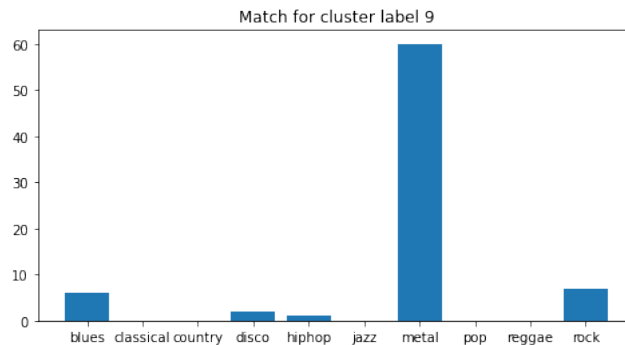


Figure 5.6: Prediction of Genres with K-Means

After applying the K-Mean algorithm on our trained dataset, our cluster model could label and predict our genre. The above figure shows that our model distinguishes metal among other found genres in clusters with an accuracy of 78.9%

5.4 Recurrent Neural Network-Long Short-Term Memory Model

In this model, we are using the JSON file we extracted, which contains features of the audio files. We are also using the same sequential model here. However, this

time we are using RNN-LSTM architecture. We are using R.N.N.- L.S.T.M. architecture. Moreover, we have also used rectified linear unit as an activation layer. We also use SoftMax at the fully connected layer. Initially, we started with 30 epochs for training the data, and we got a test accuracy of 60%, and loss was 0.854. We achieved a classification accuracy of around 55%-60%. After increasing our epochs to 50, the accuracy we achieved for the test set was 66.72%. Here, we got less accuracy, whereas we got much better accuracy with CNN because CNN has more learning capability.

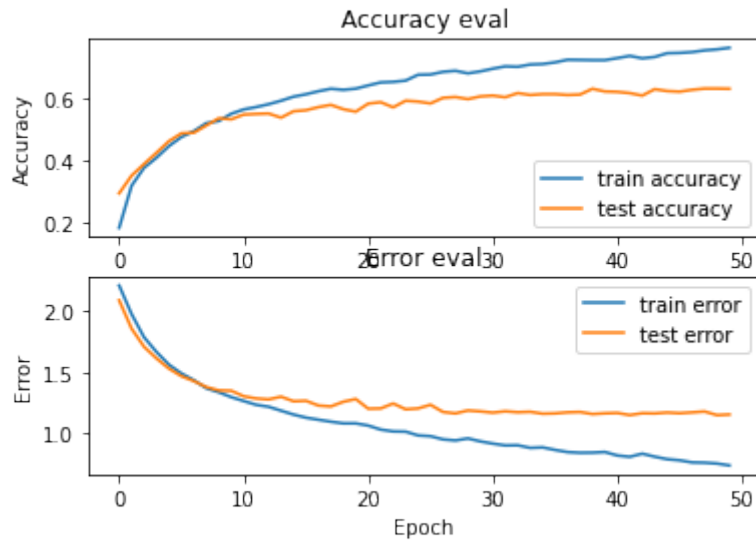


Figure 5.7: Accuracy of the RNN-LSTM model

Chapter 6

Result and Analysis

6.1 Model Summary

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

Table 6.1: Sequential Model with CNN (CSV File)

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

Table 6.2: Sequential Model with CNN (JSON File)

6.2 Result Analysis

Table 6.1 shows the result of Keras's 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 data are trainable. There is zero non-trainable data. Here we

started with 70 epochs as training time, got 87% accuracy, increased the epoch to 300, and found out the loss was reduced, and accuracy increased to 92%. We got 92.05% 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, and others. Due to the vast of data, the accuracy achieved is 92% (Figure 5.2).

Table 6.2 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% (Figure 5.4). 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.

In Figure 5.6, we also tested other algorithms like K-Means Clustering, where we distinguished genres in clusters with an accuracy of 70%-80%. We also tested the RNN-LSTM model with an accuracy of 66.72%, shown in Figure 5.7.

Chapter 7

Conclusion and Future Work

7.1 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 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 files and CSV files. We found out from our thesis that the sequential model gives the best accuracy with the CSV file as input data.

7.2 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%. 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 problems like overfitting to get better accuracy.

Bibliography

- [1] A. v. d. Oord, S. Dieleman, H. Zen, *et al.*, *Wavenet: A generative model for raw audio*, Sep. 2016. [Online]. Available: <https://arxiv.org/abs/1609.03499>.
- [2] L. Wyse, *Audio spectrogram representations for processing with convolutional neural networks*, Jun. 2017. [Online]. Available: <https://arxiv.org/abs/1706.09559>.
- [3] H. Bahuleyan, *Music genre classification using machine learning techniques*, Apr. 2018. [Online]. Available: <https://arxiv.org/abs/1804.01149>.
- [4] S. Abdoli, P. Cardinal, and A. L. Koerich, *End-to-end environmental sound classification using a 1d convolutional neural network*, Jun. 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S0957417%20419304403>.
- [5] [Online]. Available: https://www.researchgate.net/profile/George-Tzanetakis/publication/3333877_Musical_Genre_Classification_of_Audio_Signals/links/0912f50f7239509ac3000000/Musical-Genre-Classification-of-Audio-Signals.pdf.
- [6] *Improved music feature learning with deep neural networks*. [Online]. Available: <https://ieeexplore.ieee.org/document/6854949>.
- [7] T. Lidy and A. Rauber, *Evaluation of feature extractors and psycho-acoustic transformations for music genre classification: Semantic scholar*. [Online]. Available: https://www.researchgate.net/publication/220723155_Evaluation_of_Feature_Extractors_and_Psycho-Acoustic_Transformations_for_Music_Genre_Classification.
- [8] *Long short-term memory recurrent neural network based segment features for music genre classification*. [Online]. Available: <https://ieeexplore.ieee.org/document/7918369>.
- [9] *Psychoacoustics*. [Online]. Available: https://books.google.com.bd/books/about/Psychoacoustics.html?id=xlNRAAAAMAAJ&redir_esc=y.
- [10] *Robust deep feature extraction method for acoustic scene classification*. [Online]. Available: <https://ieeexplore.ieee.org/document/8947252/>.
- [11] Song, Zhang, D. Re, *et al.*, *Content-based information fusion for semi-supervised music genre classification*. [Online]. Available: <https://ur.booksc.me/book/30660078/3f1fe3>.
- [12] G. Tzanetakis and P. Cook, *[pdf] musical genre classification of audio signals: Semantic scholar*. [Online]. Available: <https://www.semanticscholar.org/paper/Musical-genre-classification-of-audio-signals-Tzanetakis-Cook/7ab881283270e427b05c6e9469562ff39dd6282a>.