A Mini Project Report on

Music Genre Classification

Submitted in partial fulfilment for the
degree of Bachelor of Technology in
Computer Science and Technology Information

Submitted by
Vrushali Ingle (1812014)
Dhanashree Divecha (1812009)
Priyanka Patil (1812032)

Under the guidance of
Ms. Amrapali Patil

Usha Mittal Institute of Technology
S.N.D.T. Women's University
Juhu Tara Road, Santacruz(West),
Mumbai-400049
2020-2021

Certificate

This is to certify that **Vrushali Ingle, Dhanashree Divecha, Priyanka Patil** has completed the Project report on the topic **Music Genre Classification** satisfactorily in partial fulfillment for the **Bachelor's Degree in Computer Science and Technology** under the guidance of **Ms. Amrapali Patil** during the year **2020-2021** as prescribed by **S.N.D.T. Women's University, Mumbai**.

Guide	Head Of Department
Ms. Amrapali Patil	Dr. Sanjay Pawar

Principal
Dr. Sanjay Pawar

Examiner 1	Examiner 2

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

(Name of the student)
Vrushali Ingle (1812014)
Dhanashree Divecha (1812009)
Priyanka Patil (1812032)

Date

Abstract

Music is a form of art that uses sound organized in time. It plays a significant role in everybody's lives. It brings like-minded people together. Communities can be recognized by the type of songs that they compose, or even listen to. The purpose of our project is classifying the music genre using the CNN algorithm. In this project we will train our model using GTZAN dataset to classify the test data (audio). For extracting the features of the audios in the dataset we will be using MFCC.

Categorizing music files according to their genre is a challenging task in the area of music information retrieval (MIR). Music genre classification is important to obtain music from a large collection.Companies nowadays use music classification, either to be able to place recommendations to their customers or simply as a product.It finds applications in the real world in various fields like automatic tagging of unknown piece of music.

Contents

List of Figures

Chapter 1

Introduction

The amount of data that is available to us is rapidly increasing every day, to the point where manual curation is becoming infeasible and classification using automated systems a necessity. The music industry is no exception. Automating the process of music tagging would result in better organization of the data and thereby making further development using this data easier, such as creating themed playlists or recommending songs to users. Machine learning can be used to find the subtle patterns in the data, which would otherwise be very difficult to explicitly code algorithms for. One such case is determining what genre a song belongs to, which is the use case this report will cover. However, finding patterns in audio is not only useful for musical analysis. It is possible that the results of this study could find use within other fields that incorporate audio.

Multimedia databases or file systems can easily have thousands of audio recordings. However, the audio is usually treated as an opaque collection of bytes with only the most primitive fields attached; namely, file format, name, sampling rate, etc. Meaningful information can be extracted from digital audio waveforms in order to compare and classify the data efficiently. When such information is extracted, it can be stored as content description in a compact way. These compact descriptors are of great use not only in audio storage and retrieval applications, but also in efficient content-based segmentation, classification, recognition, indexing and browsing of data. The need to automatically classify, to which class an audio sound belongs, makes audio classification and categorization an emerging and important research area. Automatic music genre classification is an application of artificial intelligence, more specifically machine learning, that builds a system that predicts the genre of a song. It is fairly simple for a human being to identify the genre of a song. One thinks about how fast the beat of the song is, the mood the song, the video of the song, etc. All these helps create a mental picture of the song and thus the genres associated with it are determined.

1.1 Problem Statement

Genre classification is an important task with many real-world applications. As the quantity of music being released on a daily basis continues to sky-rocket, especially on internet platforms. Every day numerous music albums are released. To manage these albums manually is arduous. The need for accurate meta-data required for database management and search or storage purposes therefore climbs in proportion. Being able to instantly classify songs in any given playlist or library by genre is an important functionality for any music streaming or purchasing service, and the capacity for statistical analysis that correct and complete labelling of music and audio provides is essentially limitless.

Classifying the genre of a song, although an inherently subjective task, comes quite easily to the human ear. Within seconds of hearing a new song one can easily recognize the timbre, distinct instruments, beat, chord progression, lyrics, and genre of the song. For machines on the other hand this is quite a complex and daunting task as the whole "human" experience of listening to a song is transformed into a vector of features about the song. Historically, machines haven't been able to reliably detect many of these musical characteristics that humans recognize in music. Currently, machine learning algorithms haven't been able to surpass the 70 percent testing accuracy. Here Convolution Neural Network is used for training and classification. The proposed system classifies music into various genres by extracting the feature vector. The aim of this project is to improve upon the accuracy of genre classification. We are considering a 10-genre classification problem with the following categories: Blues, Classical, Country, Disco, Hip-hop, Jazz, Metal, Pop, Reggae, and Rock

Chapter 2

Literature Survey

2.1 Review of various papers

[1]. Convolutional Neural Network Achieves Human-level Accuracy in Music Genre Classification [Mingwen Dong Psychology, Rutgers University (New Brunswick)] Dong [2018]

- Purpose:-It proposed a method that combines knowledge of human perception study in music genre classification and the neurophysiology of the auditory system.

- Methodology:-The method works by training a simple convolutional neural network (CNN) to classify a short segment of the music signal. Then, th[2] of a music is determined by splitting it into short segments and then combining CNN's predictions from all short segments. After training, this method achieves human-level (70%) accuracy and the filters learned in the CNN resemble the spectrotemporal receptive field (STRF) in the auditory system.

- Result :-This model achieves human-level (70%) accuracy in the 10-genre classification task. It's 10% higher than that obtained in other methods and classifies 5 more different genres with similar accuracy.

[2]. Explaining deep convolution neural networks on music classification[Keunwoo Choi, Gyorgy Fazekas, Mark Sandler]Choi et al. [2016]

- Purpose: Introduced auralisation of a CNN to understand its underlying mechanism, which is based on a deconvolution procedure to extend understanding of CNNs in music

- Methodology: To examine CNN i)The features in deeper levels are visualised by a method called deconvolution ii) The CNN architecture consists of 5 convolutional layers iii) Spectrograms of deconvolved signal is obtained from all layers

- Result: The high-level features that CNN learnt to classifier genre are robust to the variations of key, chord, and instrument.

[3]. Musical Genre Classification of Audio Signals [George Tzanetakis, Student Member, IEEE, and Perry Cook, Member, IEEE] Tzanetakis and Cook [2002]

- Purpose:- In this paper, the automatic classification of audio signals in to an hierarchy of musical genres is explored. More specifically, three feature sets for representing timbral texture, rhythmic content and pitch content are proposed.This paper classified the music using supervised machine learning approach.

- Methodology:- Gaussian Mixture model and k- nearest neighbour classifiers are the two approaches used here. They introduced 3 sets of features for this task categorized as timbral structure, rhythmic content and pitch content. Hidden Markov Models (HMMs), which have been extensively used for speech recognition tasks, have also been explored for music genre classification

- Result:-Using the proposed feature sets classification of 61% (nonreal time) and 44% (real time), has been achieved in a dataset consisting of ten musical genres.

[4]. Automatic Musical Pattern Feature Extraction Using Convolutional Neural Network[Tom LH. Li, Antoni B. Chan and Andy HW. Chun]Li et al. [2010]

- Purpose: To show that musical data have very similar characteristics to image data so that the variation of musical patterns can be captured using CNN. also to show that the musical pattern features are informative for genre classification tasks

- Methodology: i)Divided dataset in small subset ii)Firstly trained 3 most difficult-to-classify genres i.e. Blues, Metal and Rock on dataset and 4 other after that iii)Extensive experiments are also performed towards the selection of CNN network parameter

- Result: Dividing the dataset into small subsets to train the CNN feature extractors may have the side-effect that features extracted to classify songs within one subset.

[5]. Music Genre Classification using Machine Learning techniques [Hareesh Bahuleyan, (2018)] Bahuleyan [2018]

- Purpose: The work conducted gives an approach to classify music automatically by providing tags to the songs present in the user's library

- Methodology: Two models are described in this paper:
  i) CNN: This model requires only spectrogram as input. It is trained end to end with these spectrograms of audio signal.
  ii) The second approach uses various Machine Learning algorithms like Logistic Regression, Random forest etc, where it uses hand-crafted features from

time domain and frequency domain of the audio signal. iii)The manually extracted features like Mel-Frequency Cepstral Coefficients (MFCC), Chroma Features, Spectral Centroid etc are used to classify the music into its genres using ML algorithms like Logistic Regression, Random Forest, Gradient Boosting (XGB), Support Vector Machines (SVM).

- Results: By comparing the two approaches separately they came to a conclusion that VGG-16 CNN model gave highest accuracy. By constructing ensemble classifier of VGG-16 CNN and XGB the optimised model with 0.894 accuracy was achieved.

Chapter 3

Proposed System

3.1 Dataset

GTZAN from the marsyas.info website is the name of the dataset that we are going to use for the project. It contains 10 genres, namely -Blues, Classical, Country, Disco, Hip-hop, Jazz, Metal, Pop, Reggae, and Rock. Each genre has 100 tracks each of 30 seconds. In all the dataset consists of 1000 tracks. The tracks are all 22050Hz Mono 16-bit audio files in wav format. The dataset incorporates samples from a variety of sources like CDs, radios,microphone recordings, etc. The training,testing sets will be randomly partitioned.

3.2 Pre-processing

We cannot directly use the audio file as an input for our CNN, we need to preprocess it and then we can use the data in the GTZAN dataset. And preprocessing means to extract useful features from the audio signal. So, for Mel-frequency cepstral coefficient (MFCC) are one of the way to extract useful information from the signal because it defines the brightness of a sound. It can also be used to calculate the timbre (quality) of the sound. In this model, we will use the Librosa library to convert the audio file from the GTZAN datasets into MFCC features.

First, we normalized the audio data for each song to remove differences in the baseline volume at which different pieces are recorded, which does not affect their

genres.Raw audio is difficult to work with since it contains too many data points (22,500 per second) to be computationally feasible for most neural networks. It would take too long to train, and the data's detail would make pattern recognition difficult without a prohibitively large model. Thus, we tested two more compact forms of data representation: Fourier-transform coefficients and Mel-frequency cepstral coefficients (MFCC)Performing Fourier transform involves breaking the audio sample into small segments (0.1 seconds), and taking a Fast Fourier Transform (FFT) of each segment. The resulting Fourier coefficients vectors were stacked along the time axis to form a 4 time-series matrix of Fourier coefficients, which can be treated like an "image" when training. The FFT was performed using a Numpy function.
Performing the MFCC transform involves the following steps:

- Take the Fourier transform of each segment of audio
- Map the powers of the spectrum obtained above onto the Mel-scale, using triangular overlapping windows.
- Take the logarithms of the powers at each of the Mel-frequencies.
- Take the discrete cosine-transform of the list of Mel-log powers.
- The MFCCs are the amplitudes of the resulting spectrum This process was implemented using the Librosa library (a toolset designed for sound processing) . We tried MFCC because it leads to significant accuracy improvements

3.3 Feature Extraction using MFCC

It includes identifying the linguistic content and discarding noise. For audio and music feature extraction, mel-frequency cepstral coefficients (MFCCs) are

extracted from the song or audio to train the model and extract only relevant information from audio. We will represent an audio file in digital format, the computer looks at it as a wave with X axis as Time and Y axis as Amplitude.A mel spectrogram is a spectrogram where the frequencies are converted to the mel scale. The normal spectogram can be used for extracting features, but this still contains some amount of additional information which is not required. As the human ear works on logarithmic scale and not linear scale, we use mel spectograms, which convert this spectogram into a logarithmic representation to get the features more accurately by removing or eliminating unwanted features.
The MFCCs uses a mel scale, which is used to extract the features from an audio signal, which when represented as a graph, turns out to be a mel specotgram. So,what we see on a mel spectorgram is the exact features we need for training our model.

CNN: A CNN is composed by three main layers:

- Convolutional Layer:
  The convolutional layer is the one tasked with applying the convolution operation on the input.This is done by passing a filter (or kernel) over the matricial input, computing the convolution value, and using the obtained result as the value of one cell of the output matrix (called feature map); the filter is then shifted by a predefined stride along its dimensions. The filters parameters are trained during the training process.

- Detector layer:
  In the detector layer, the output of the convolution is passed through a non-linear function, usually a ReLU function.

- Pooling layer:
  The pooling layer is meant to reduce the dimensionality of data by combining the output of neuron clusters at one layer into one single neuron in the

subsequent layer. The last layer of the network is a fully connected one (a layer whose units are connected to every single unit from the previous one), which outputs the probability of the input to belong to each of the classes.

Chapter 4

Architectural Overview

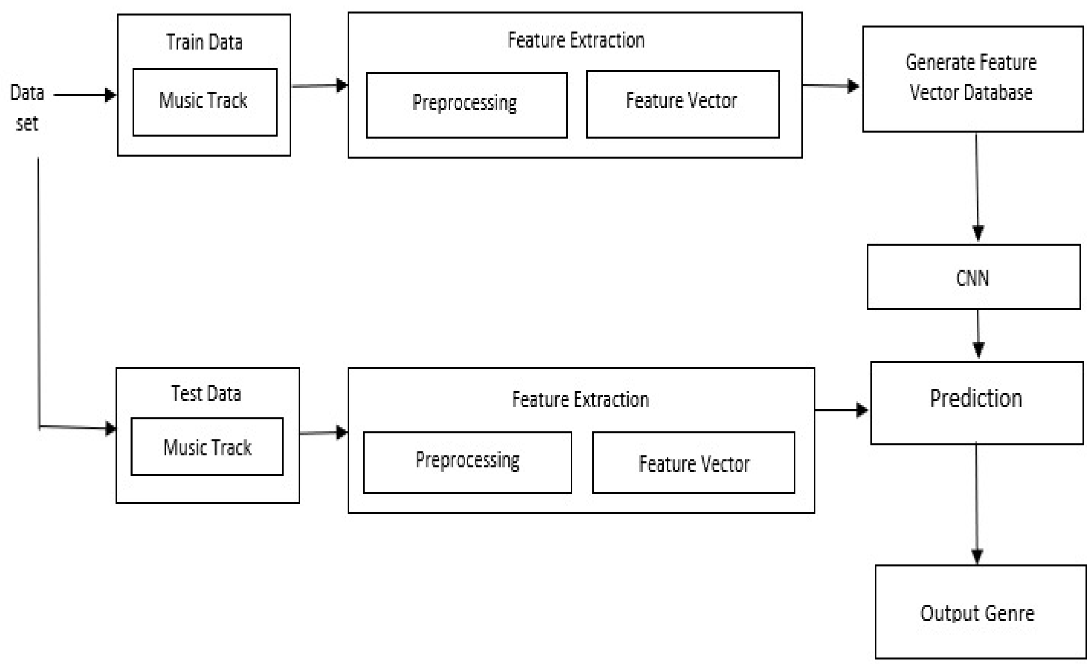

Figure 4.1: Block digram

1. Dataset:

We have randomly splitted our training and testing data from the model selection module from the Sci-kit learn library.

2. Pre-processing:

Each track from the train dataset has been preprocessed and a feature vector is extracted for the same. A Feature Vector Database has been generated from the extracted feature vectors.

3. The CNN model is trained using the obtained feature vector database.

4. Each track from the test dataset has also been preprocessed and a feature vector is extracted for the same.

5. The trained CNN model is operated on the feature vector obtained at the end of step 4 to perform classification on test data.

6. Finally, output is the genre of the music track.

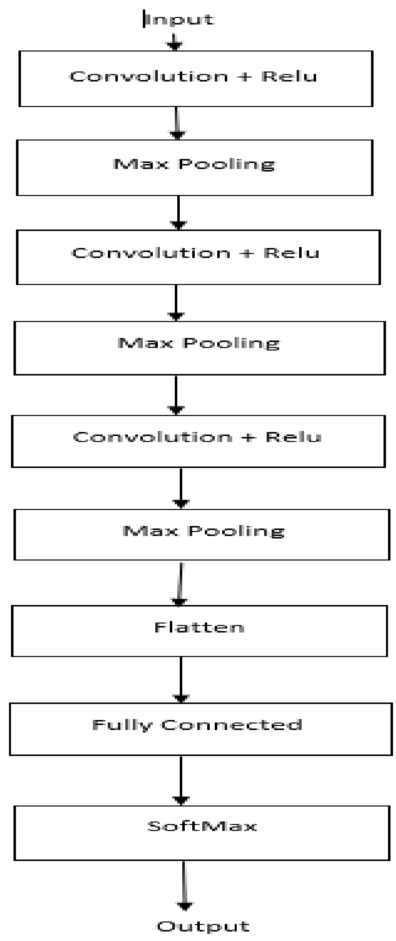

Figure 4.2: Flowchart of CNN

4.1 Hardware & Software requirement

1. Hardware requirement: 64 bit Operating System, x64-based processor

2. Software requirement: Python version 3.8 and above

3. Source code editors: Visual Studio Code, Jupyter, Pycharm (any one of these can be used)

Python libraries:
1. NumPy: It is a Python library used for working with arrays. It is a mathematical model for scientific computing

2. Pandas: It is an open source Python package that is most widely used for data science/data analysis and machine learning tasks.

3. Seaborn: It is a library for making statistical graphics in Python.

4. Matplotlib: It is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy.

5. Librosa: It is a Python package for music and audio analysis. It is speech processing library to extract features from songs.

6. Tensorflow: It is a free and open-source software library for machine learning. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.

7. Sklearn: The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.
8. Keras: It is an open-source software library that provides a Python interface for artificial neural networks.Keras acts as an interface for the TensorFlow library.

4.2 Train, Test and Validation

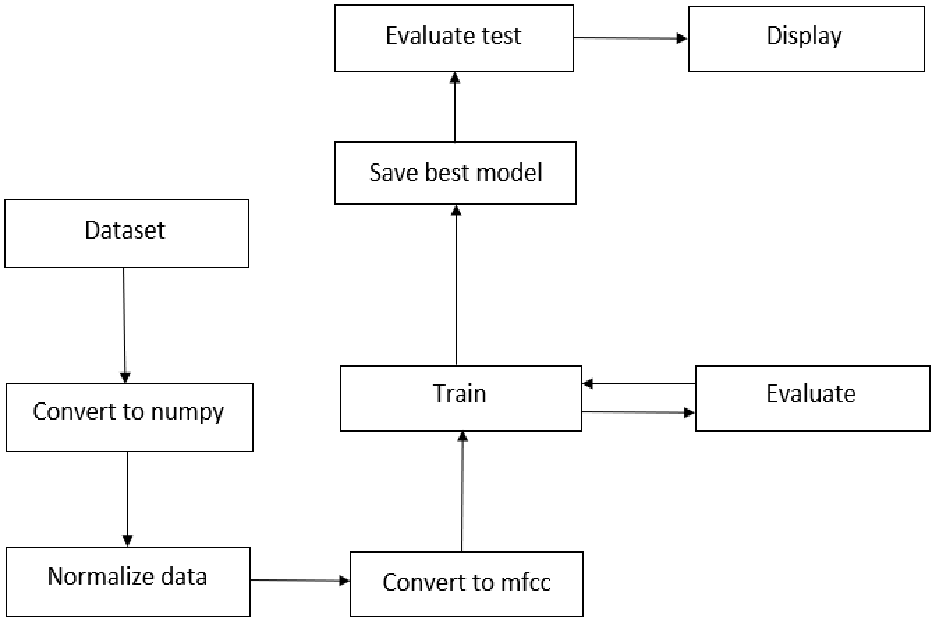

Figure 4.3: Train test validation

- Download the GTZAN dataset
- Each sample is of 30-seconds so convert them to Numpy arrays.
- Normalize each array
- Convert raw audio arrays into time series of Mel-frequency cepstral coefficients
- Split data into training, validation and test sets
- Execute training loop with periodic evaluations of validation accuracy
- Save the model with highest validation accuracy
- Load best validation accuracy model, predict genres of test data.

Chapter 5

Implementation

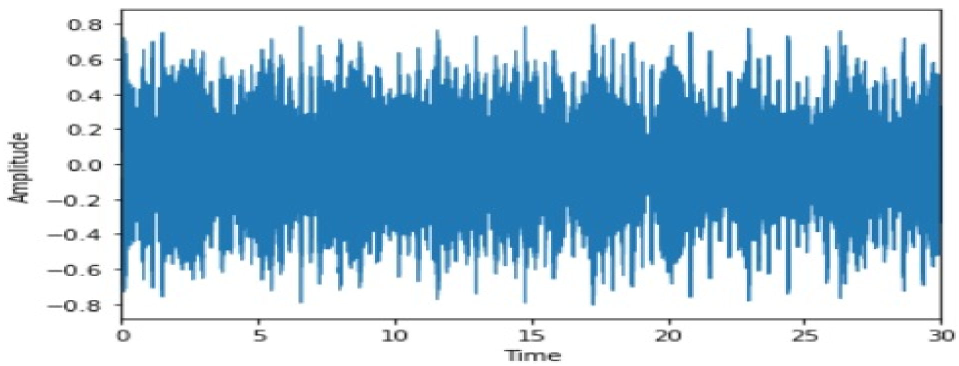

Figure 5.1: Waveform

Waveforms are visual representations of sound as time on the x-axis and amplitude on the y-axis. Waveforms are great because they allow you to quickly scan your audio data and visually compare and contrast which genres might be more similar than others.

Librosa also allows you to separate out the harmonic and percussive signals from your audio data. Harmonic sounds contain information regarding the pitch whereas percussive sounds are more perceived as two objects colliding to create certain noise in patterns. For the purpose of this analysis, it is sufficient to note that the separation of harmonic and percussive signals is only important as far as feature extractions are concerned as certain feature extractions within Librosa require harmonic signals as inputs.

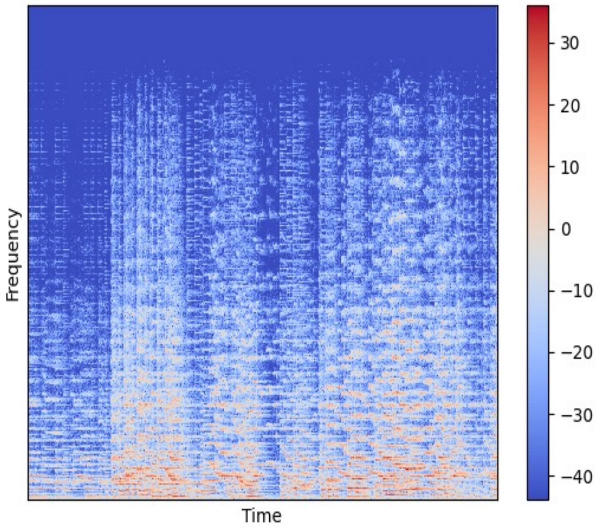

Figure 5.2: Mel spectogram

This mel spectrogram can be thought of as a visual representation of an audio signal. Specifically, it represents how the spectrum of frequencies vary over time

The Fourier transform is a mathematical formula that allows us to convert an audio signal into the frequency domain. It gives the amplitude at each frequency, and we call it the spectrum. Since frequency content typically varies over time, we performed the Fourier transform on overlapping windowed segments of the signal to get a visual of the spectrum of frequencies over time. That is called the spectrogram. Finally, since humans do not perceive frequency on a linear scale, we map the frequencies to the mel scale (a measure of pitch), which makes it so that equal distances in pitch sound equally distant to the human ear.What we have essentially done is turned the problem into an image classification task As we know that CNN is basically used to classify images, by getting spectrograms we make it easy for our program to classify genres.

```
genres\rock\rock.00096.wav, segment:0
genres\rock\rock.00096.wav, segment:1
genres\rock\rock.00096.wav, segment:2
genres\rock\rock.00096.wav, segment:3
genres\rock\rock.00096.wav, segment:4
genres\rock\rock.00097.wav, segment:0
genres\rock\rock.00097.wav, segment:1
genres\rock\rock.00097.wav, segment:2
genres\rock\rock.00097.wav, segment:3
genres\rock\rock.00097.wav, segment:4
genres\rock\rock.00098.wav, segment:0
genres\rock\rock.00098.wav, segment:1
genres\rock\rock.00098.wav, segment:2
genres\rock\rock.00098.wav, segment:3
genres\rock\rock.00098.wav, segment:4
genres\rock\rock.00099.wav, segment:0
genres\rock\rock.00099.wav, segment:1
genres\rock\rock.00099.wav, segment:2
genres\rock\rock.00099.wav, segment:3
genres\rock\rock.00099.wav, segment:4
```

Figure 5.3: Preprocessing

Preprocessing of data is required before we finally train the data. We focused on column that is 'label' and encoded it with the function LabelEncoder() of sklearn.preprocessing.

We can't have audio file in our data if we're going to run any kind of model on it. So before we can run a model, we need to make this data ready for the model. To convert this kind of categorical audio data into model-understandable numerical data, we use the Label Encoder class.For more accuracy of our model we segmented the sudio file in 5 segments and then the labeled each songs file. Also extracted the Mel-spectrogram in numarical form.

```
Epoch 25/30
94/94 [==============================] - 9s 97ms/step - loss: 0.6351 - accuracy: 0.7795 - val_loss: 1.0600 - val_accuracy: 0.6
347
Epoch 26/30
94/94 [==============================] - 8s 90ms/step - loss: 0.6289 - accuracy: 0.7789 - val_loss: 1.0660 - val_accuracy: 0.6
400
Epoch 27/30
94/94 [==============================] - 9s 96ms/step - loss: 0.6072 - accuracy: 0.8012 - val_loss: 1.0433 - val_accuracy: 0.6
333
Epoch 28/30
94/94 [==============================] - 9s 97ms/step - loss: 0.5591 - accuracy: 0.8075 - val_loss: 1.0384 - val_accuracy: 0.6
413
Epoch 29/30
94/94 [==============================] - 9s 98ms/step - loss: 0.5533 - accuracy: 0.8122 - val_loss: 1.0419 - val_accuracy: 0.6
360
Epoch 30/30
94/94 [==============================] - 9s 96ms/step - loss: 0.5399 - accuracy: 0.8266 - val_loss: 1.0231 - val_accuracy: 0.6
507
40/40 [==============================] - 1s 22ms/step - loss: 1.0185 - accuracy: 0.6504
Accuracy on test set is: 0.6503999829292297
```

Figure 5.4: Epoch

We have set the number of epochs equals to 30. And the accuracy that we get is 65%.


```
{
    "mapping": [
        "C:\\Users\\priya\\Desktop\\genres\\blues",
        "C:\\Users\\priya\\Desktop\\genres\\classical",
        "C:\\Users\\priya\\Desktop\\genres\\country",
        "C:\\Users\\priya\\Desktop\\genres\\disco",
        "C:\\Users\\priya\\Desktop\\genres\\hiphop",
        "C:\\Users\\priya\\Desktop\\genres\\jazz",
        "C:\\Users\\priya\\Desktop\\genres\\metal",
        "C:\\Users\\priya\\Desktop\\genres\\pop",
        "C:\\Users\\priya\\Desktop\\genres\\reggae",
        "C:\\Users\\priya\\Desktop\\genres\\rock"
    ],
    "labels": [
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
```


Figure 5.5: Json file content

After the preprocessing step we got the json file.


```
            [
                -323.90240478515625,
                142.16656494140625,
                -7.789546012878418,
                14.89661979675293,
                25.607173919677734,
                -5.6631622314453125,
                4.887836933135986,
                6.7422075271606445,
                -4.853984832763672,
                4.2482218742370605,
                6.575334548950195,
                3.8598549365997314,
                6.312807083129883
            ],
            [
                -313.2419128417969,
                140.0521240234375,
                -12.904825210571289,
                8.352254867553711,
                23.357006072998047,
                3.479233503341675,
                7.604862213134766,
                15.01456069946289,
                -1.179320216178894,
                2.057659149169922,
                1.8722058534622192,
                1.9015816450119019,
                4.994598865509033
            ],
```


Figure 5.6: Json file contents

```
#make prediction on a sample
X = X_test[100]
y=y_test[100]

predict(model, X, y)

#index 0: blues
#index 1:classical
#index 2:country
#index 3:disco
#index 4:hiphop
#index 5:jazz
#index 6:metal
#index 7:pop
#index 8:reggae
#index 9:rock
```

```
Expected index: 7, Predicted index: [7]
```

Figure 5.7: Output

The model has correctly predicted the genre (Pop) of the song as expected genre and predicted genre is same.

Chapter 6

Advantages and Applications

6.1 Advantages

1. Convenient to use - This system is easy to use and is very simple to handle.
2. Effective - CNN is very effective as it helps the user to predict the genre easily.
3. Flexible - It is quite flexible and can be run on any system.

6.2 Applications

Music genre classification provides the following features/functions.
1. Conversion of music file - The input music file gets converted into statistical data using Librosa library.
2. Detection of genre - The main function of the application is to detect the genre of the song/music file inputted.

Chapter 7

Future scope

Our project makes a basic attack on the music genre classification problem, but could be extended in several ways. Our work doesn't give a completely fair comparison between learning techniques for music genre classification. Adding a validation step to the DAG SVM would help determine which learning technique is superior in this application. We used a single feature (MFCCs) throughout this project. Although this gives a fair comparison of learning algorithms, exploring the effectiveness of different features would help to determine which machine learning stack does best in music classification.

Since genre classification between fairly different genres is quite successful, it makes sense to attempt finer classifications. The exact same techniques used in this project could be easily extended to classify music based on any other labelling, such as artist. In addition, including additional metadata text features such as album, song title, or lyrics could allow us to extend this to music mood classification as well.We may also include features that enable music recording and using to detect the genre. This can be useful in identifying the genre without using file selection. We can implement song recognition by adding extra features for lyrics detection.The current computer application can be converted into an android application and can be deployed in Google Play Store.

References

Hareesh Bahuleyan. Music genre classification using machine learning techniques. *arXiv preprint arXiv:1804.01149*, 2018.

Keunwoo Choi, George Fazekas, and Mark Sandler. Explaining deep convolutional neural networks on music classification. *arXiv preprint arXiv:1607.02444*, 2016.

Mingwen Dong. Convolutional neural network achieves human-level accuracy in music genre classification. *arXiv preprint arXiv:1802.09697*, 2018.

Tom LH Li, Antoni B Chan, and Andy HW Chun. Automatic musical pattern feature extraction using convolutional neural network. *Genre*, 10:1x1, 2010.

George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10(5):293–302, 2002.

Acknowledgement

Foremost, I would like to express my sincere gratitude to Professor Dr. Sanjay Pawar, Head of Department (Computer Science and Technology) and my guide, Ms. Amrapali Patil for her is valuable guidance and continuous support during the project; her patience, motivation, enthusiasm, and immense knowledge. Her direction and mentoring helped me to work successfully on the project topic.

My sincere gratitude to Dr. Sanjay Pawar, Principal (Usha Mittal Institute of Technology) for her valuable encouragement and insightful comments

I would also like to thank to all the teaching and non-teaching staff for their valuable support.

Last but not the least I would like to thank to my parents and friends.

Date:

Name of Candidate
Vrushali Ingle (1812014)
Dhanashree Divecha (1812009)
Priyanka Patil (1812032)