A

PROJECT REPORT

ON

"Music Genre Classification System"

SUBMITTED TO

SHIVAJI UNIVERSITY, KOLHAPUR

IN THE PARTIAL FULFILLMENT OF REQUIREMENT FOR THE AWARD OF DEGREE BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY

Sonali Kadolkar	17UCS51017XX
Shrutika Patil	17UIT11045CS
Priyanka Girmal	18UCS71002XX
Abhishek Pandharpatte	18UCS72009XX
Priya Sawant	18UCS71011XX

UNDER THE GUIDANCE OF

Prof. P.M.GAVALI

DKTE

Promoting Excellence in Teaching, Learning & Research

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DKTE SOCIETY'S TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI

(AN AUTONOMOUS INSTITUTE)

ACCREDITED WITH 'A+' GRADE BY NAAC

An ISO 9001: 2015 Certified

SHIVAJI UNIVERSITY KOLHAPUR

2020-2021

D.K.T.E.SOCIETY'S
TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI
(AN AUTONOMOUS INSTITUTE)
ACCREDITED WITH 'A+' GRADE BY NAAC

An ISO 9001: 2015 Certified

SHIVAJI UNIVERSITY KOLHAPUR

2020-2021

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that, project work entitled

"Music Genre Classification System"

Is a bonafied record of project work carried out in this college by

Sonali Kadolkar	**17UCS51017XX**
Shrutika Patil	**17UIT11045CS**
Priyanka Girmal	**18UCS71002XX**
Abhishek Pandharpatte	**18UCS72009XX**
Priya Sawant	**18UCS71011XX**

In the partial fulfillment of award of degree Bachelor in Engineering in Computer Science & Engineering prescribed by Shivaji University, Kolhapur for the academic year 2020-2021.

Prof. P.M.GAVALI
(PROJECT GUIDE)

Prof.(Dr.) D.V.KODAVADE
(HOD CSE DEPT.)

Prof.(Dr.) P.V.KADOLE
(PRINCIPAL)

EXAMINER: ________________

DECLARATION

We hereby declare that, the project work report entitled "**Music Genre Classification System"** which is being submitted to D.K.T.E. Society's Textile and Engineering Institute Ichalkaranji, affiliated to Shivaji University, Kolhapur is in partial fulfillment of degree B.Tech. (CSE). It is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any university or institution for the award of any degree. Further, we declare that we have not violated any of the provisions under Copyright and Piracy / Cyber / IPR Act amended from time to time.

NAME	**Roll no.**	**Signature**
Sonali Kadolkar	17UCS51017XX	
Shrutika Patil	17UIT11045CS	
Priyanka Girmal	18UCS71002XX	
Abhishek Pandharpatte	18UCS72009XX	
Priya Sawant	18UCS71011XX	

ACKNOWLEDGEMENT

With great pleasure we wish to express our deep sense of gratitude to Prof. P.M.Gavali for his valuable guidance, support and encouragement in completion of this project report. Also, we would like to take opportunity to thank our head of department Dr. D. V. Kodavade for his co-operation in preparing this project report.

We feel gratified to record our cordial thanks to other staff members of Computer Science and Engineering Department for their support, help and assistance which they extended as and when required.

Thank you,

Sonali Kadolkar	17UCS51017XX
Shrutika Patil	17UIT11045CS
Priyanka Girmal	18UCS71002XX
Abhishek Pandharpatte	18UCS72009XX
Priya Sawant	18UCS71011XX

ABSTRACT

Categorizing music files according to their genre is a challenging task in the area of music information retrieval (MIR). This project was primarily aimed to create an automated system for classification model for music genres. The first step included finding good features that demarcated genre boundaries clearly. A total of five features, namely MFCC vector, chroma frequencies, spectral rolloff, spectral centroid, zero-crossing rate were used for obtaining feature vectors for the classifiers from the GTZAN genre dataset. Many different classifiers were trained and used to classify, each yielding varying degrees of accuracy in prediction. An ensemble classifier based on majority voting was then created to incorporate all of the classifiers into one.

INDEX

INTRODUCTION

There are number of music streaming services like Apple music, Spotify, Gaana, YouTube etc. These services always try to provide best of the experience to their users by providing amazing playlists that include songs that they like as well as introducing the users to the similar songs that they may have never listened to before. We can view a song as, depending on length, approximately 1.3 million data-points. In order to classify songs as similar we'd like to look at these data-points and it becomes increasingly difficult to try to do this with the raw data especially when new songs are introduced every day. Also, similarity between songs is a difficult thing to define because the parameters won't to describe how similar two songs are inherently subjective and cannot be easily translated into an algorithm.

Music plays a very vital role in people's lives. Music can be said as a thread which binds the like-minded people together and forms different communities. These communities can be identified by the type of songs they compose or even listen to. To identify such communities it is important to be able to identify the similarities in different songs. One way of doing this is to identify genre of songs, which is identified by some characteristics of the music such as rhythmic structure, harmonic content and instrumentation. To make the process easier automatic classification of songs supported genre will be helpful. So that is why in today's world automatic classification of music has become an important aspect. Being able to automatically classify and supply tags to the music present during a user's library, based on genre, would be beneficial for audio streaming services.

A. PROBLEM DEFINITION

Currently there are huge number of songs and they are to be classified manually according to their respective genres. The proposed system is going to classify the songs into correct genre with the help of neural networks.

In this project we are going to create a classifier such that it will be able to classify the music based on its genre. The very popular GTZAN dataset will be used here to train the model. After training, model will be able to predict the genre of the music accurately.

B. AIM & OBJECTIVE OF THE PROJECT

The aim of this project is to implement an algorithm that reduces the amount of data-points we'd like to work with and build a classifier. The classifier should be ready to categorize a given music track into the proper genre supported how similar the extracted features are to the music samples we want to construct the classifier for.

Music classification is a way to accurately perceive the things we hear and describe them to others. Music Genres are important. They represent one among the foremost important and valuable tools we have to appreciate, understand, and communicate with each other about the music we always hear. There is no doubt that this system will be able to classify music according to genre and will be helpful to Music Information Retrieval systems.

Objectives:

1. To build a model that classifies Music into its respective genre based on various different features, instead of manually entering the genres.

2. To develop a model with good accuracy so that it classifies the new music into its genre correctly.

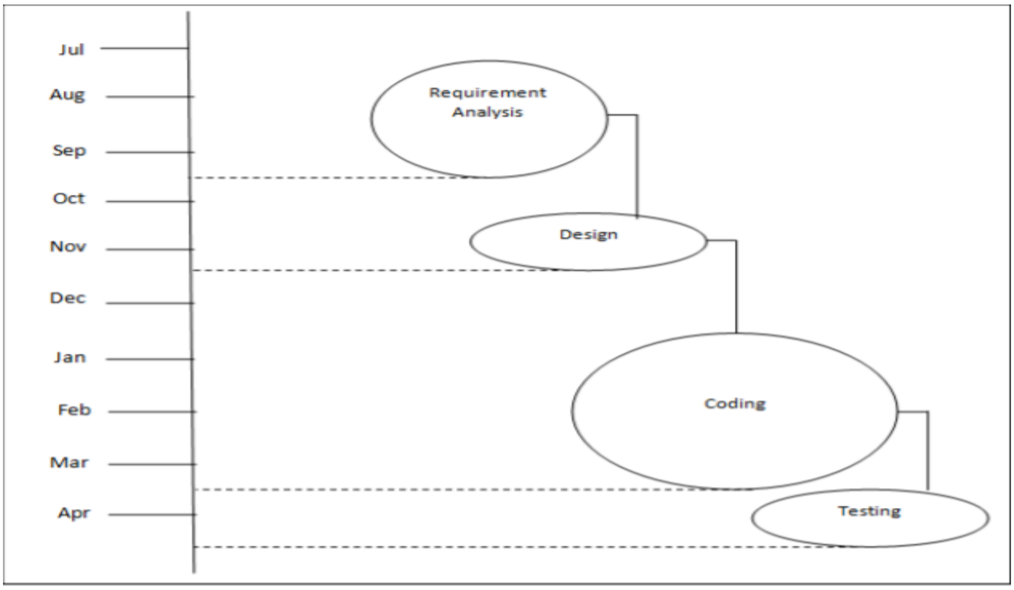
C. SCOPE & LIMITATION OF THE PROJECT

The acquisition of reliable ground truth could also be a key requirement of coaching effective genre classifiers. It’s been suggested that only limited agreement are often achieved among human annotators when classifying music by genre, which such limits impose an unavoidable ceiling on automatic genre classification performance. Individuals can't only differ on how they classify a given recording, but they also are going to be ready to differ in terms of the pool of genre labels from which they choose. only a few genres have clear definitions, and what information is out there out there's usually confusing, ambiguous and inconsistent from source to source. Usually there's significant overlapping between genres. Individual recordings can possibly belong to multiple genres of varying degrees. There are often complex relationships between genres. Some genres are broad while some are narrow. Furthermore, genres usually form multiple different clusters.

D. TIMELINE OF THE PROJECT

We have used classic life cycle paradigm also called “Water Fall Model”. Software engineering is a sequential approach to software development that begins at the system level and progress through analysis, design, coding, testing and maintenance. We had completed software requirement analysis by the mid of September 2020 which encompasses both system and software requirement gathering. By the end of December 2020 we had completed project planning and design. On the basis of design prepared in the previous stage by the end of March 2021 we completed coding stage.

After completion of coding stage the important part in the software development which is testing phase carried out in first week of April 2021.Various criteria of testing were taken into account which includes unit testing, integration testing, validation testing and system testing. First, each and every module of the project was tested under the unit testing. After the unit testing, integration testing was carried out by integrating all module tested in unit testing. After unit testing the module prepared was cross checked with the design.



BACKGROUND STUDY AND LITERATURE OVERVIEW

A. LITERATURE OVERVIEW

The problem of Music Genre Classification has been approached in various ways. Tzanetakis, Sam Clark, and other papers have approached this problem specific to certain algorithms. John cast, suggest that features coupled with MFCCs improve classification accuracy. Martin, suggests that the temporal modulations of MFCCs are important for classification and its performance is better in comparison with the Standard low-level feature set because of its increased ability to classify background crowd noise and popular music. C H lee, proposed a novel feature set for music genre classification based on cepstral (MFCC) features which achieves higher classification accuracy. Most work on this problem is done by using specific data preprocessing techniques like FFT or MFCC exclusively to convert audio data and the training of classification algorithms is done based on these data values. We believe there is no prior work analyzing how various algorithms perform using MFCC and FFT data values to the extent of our knowledge. Hence this paper is a study on such an analysis. From the analysis it is clear that using MFCC data values gives better results overall than using FFT values. The Simpler algorithms such as Logistic Regression and Kth Nearest Neighbors did fairly well in comparison to superior algorithms such as Recurrent Neural Networks and Support Vector Machines. The highest accuracy reached was 86% using Neural Networks.

B. INVESTIGATION OF CURRENT PROJECT AND RELATED WORK

The basis of any kind of automatic audio analysis system is that the extraction of feature vectors. A large number of different kind of feature sets, mainly deriving from the world of speech recognition, are proposed to represent audio signals. Typically they're supported some sort of time-frequency representation. Although an entire overview of audio feature extraction is beyond the scope of this paper, some relevant representative audio feature extraction references are provided. Automatic classification of audio has also an extended history originating from speech recognition. Mel-frequency cepstral coefficients (MFCC), are a group of perceptually motivated features that have been widely utilized in speech recognition. These MFCC features provide a brief representation of the spectral envelope, such that most of the signal energy is concentrated within the first coefficients.

More recently, audio classification techniques that include non-speech signals have been proposed and evolved with time. Most of those systems target the classification of broadcast news and video in broad categories like music, speech, and environmental sounds. The problem of discrimination between music and speech has received considerable attention from the first work of Saunders where simple thresholding of the typical zero-crossing rate and energy features is used, to the work of Scheirer and Slaney where multiple features and statistical pattern recognition classifiers are carefully evaluated. In D. Kimber and L. Wilcox, “Acoustic segmentation for audio browsers” project, audio signals are segmented and classified into “music,” “speech,” “laughter,” and non-speech sounds using cepstral coefficients and a hidden Markov model (HMM). A heuristic rule-based system for the segmentation and classification of audio signals from movies or TV programs supported the time-varying properties of simple and straightforward features is proposed in T. Zhang and J. Kuo’s “Audio content analysis for online audio-visual data segmentation and classification”. Signals are classified into two main and broad groups of music and non-music. They are further subdivided into harmonic environmental sound, song, speech with music which is considered into a music category, and pure speech and non-harmonic environmental sound which is considered into non-music category. Berenzweig and Ellis affect the harder problem of locating voice segments in musical signals. In their system, the phoneme activation output of an automatic speech recognition system is employed because the feature vector for classifying singing segments.

Another sort of non-speech audio arrangement involves isolated instrument sounds and sound effects. In the pioneering work of Wold automatic retrieval, classification and clustering of musical instruments, sound effects, and environmental sounds using automatically extracted features is explored. The features utilized in their system are statistics (mean, variance, autocorrelation) over the entire sound file of short time features like pitch, amplitude, brightness, and bandwidth. Using an equivalent dataset various other retrieval and classification approaches are proposed.

REQUIREMENT ANALYSIS

RS1: The hardware requirement:

- Personal computer with standard configuration.
- RAM – Minimum 4 GB.
- Processor – Intel Core i3 or higher.

RS2: The Software requirement:

- Operating system – Windows 8 or higher.
- Python installed on machine or Anaconda Distribution of Python 3.

RS3: The system should have the sufficient amount of dataset.

The sufficient amount of data is required so that the model can be trained with more accuracy.

RS4: The system should have Librosa library for converting audio files into melspectrogram.

Librosa is a Python module to analyze audio signals in general but geared more towards music.

RS5: The system should have other basic libraries for processing of the audio files.

To perform different operations on the extracted features of the data, different modules like numpy, matplotlib, pandas, etc are required.

RS6: The system should have a database to store the Feature Vector.

The system should be able to store the features extracted from each sample of audio so that it will be easier for further processing.

SYSTEM DESIGN

DATA FLOW DIAGRAM

DFD Level 0:

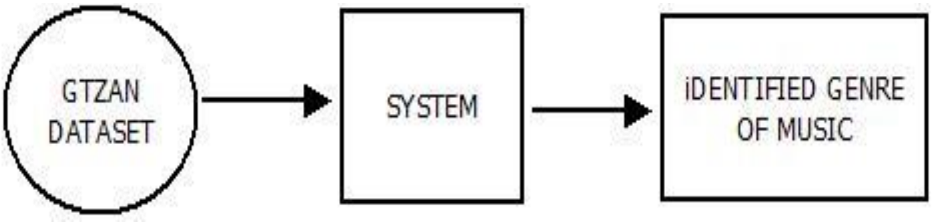


Fig 1: DFD Level 0

DFD Level 1:

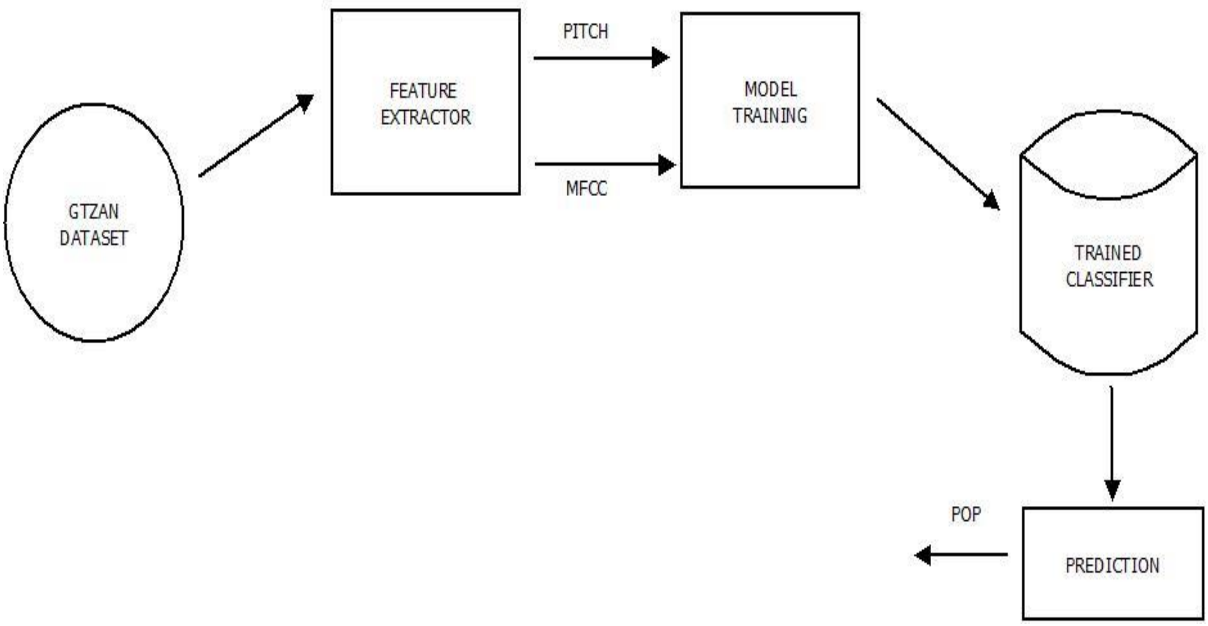


Fig 2: DFD Level 1

FLOWCHART

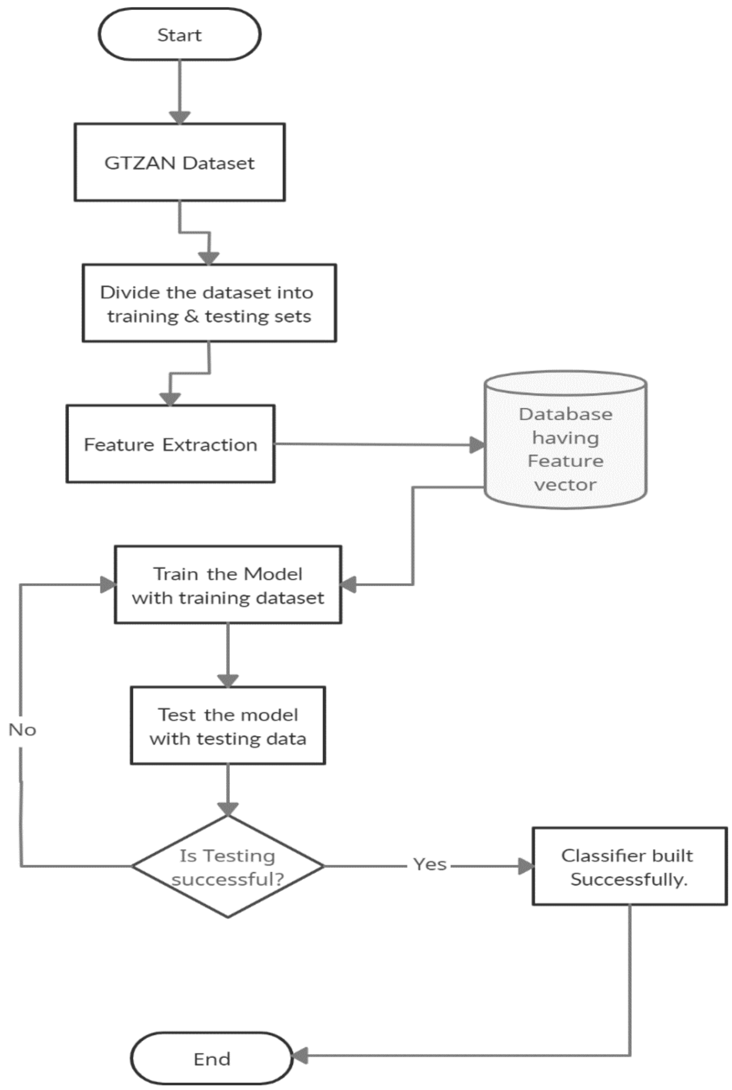


Fig 3: Flowchart

CLASS DIAGRAM

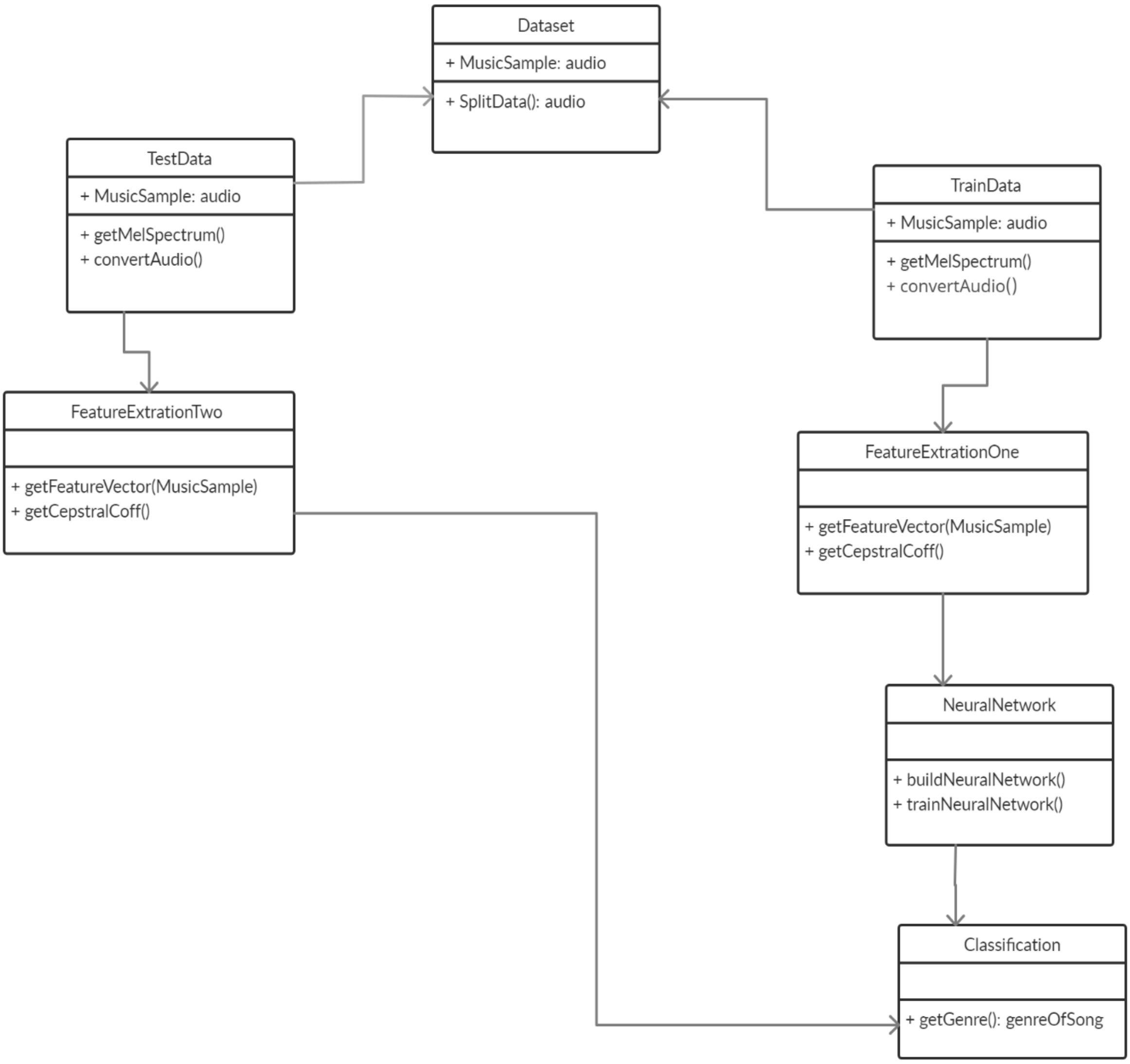


Fig 4: Class Diagram

SEQUENCE DIAGRAM

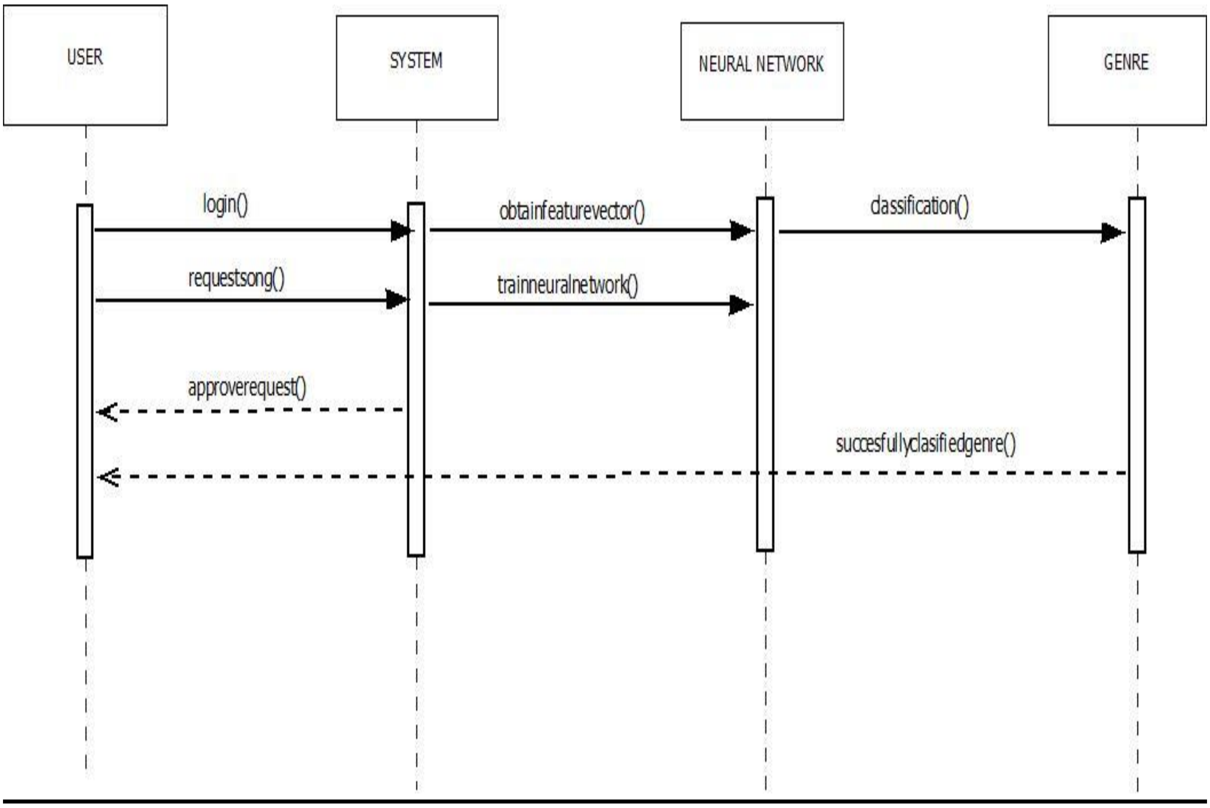


Fig 5: Sequence Diagram.

COLLABORATION DIAGRAM

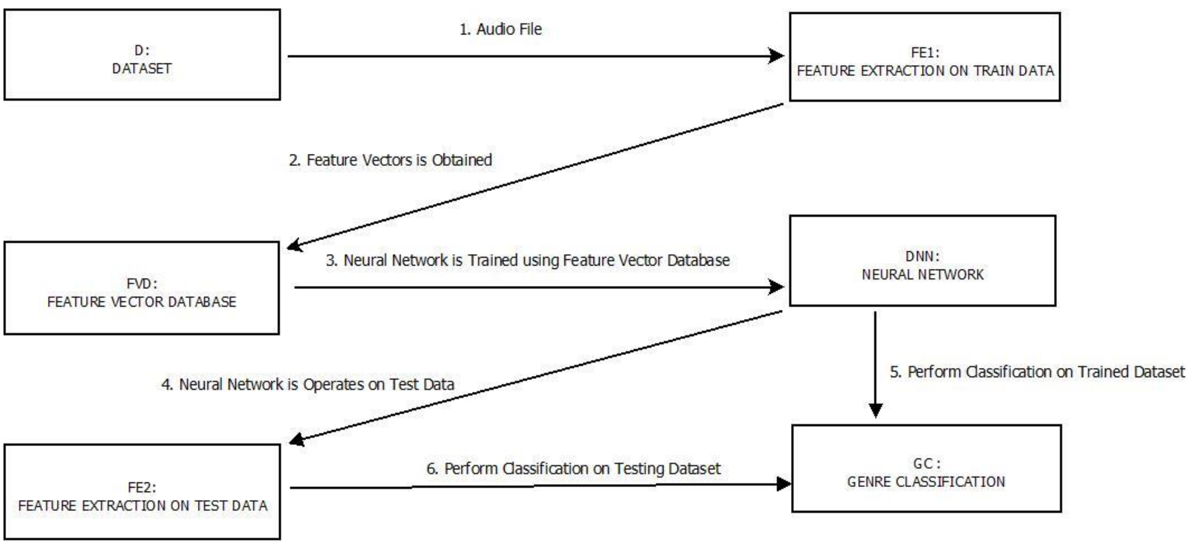


Fig 6: Collaboration Diagram

IMPLEMENTATION

There are different number of approaches to develop the system for classification of Music based on the genre. One of them is:

- The ground-breaking work, based on Naive Bayesian and Neural Network approaches, identifies one out of four styles of a musician improvisation. There work included testing a performer's ability to consistently produce intentional styles which should be different from each other. A database was created to train the classifiers. While classifying among the four styles an accuracy of 98% was achieved. Then while using eight classifiers, trained model used to return "yes" or "no" for eight different styles. They got an overall accuracy of 77–90%.

In this project we will use feature set which will have features like Short-time Fourier Transform (STFT) , Mel-frequency Cepstral Coefficients (MFCCs), etc. These features will help to represent the audio files in our dataset. With the help of these vectors we will train our classifier by using an algorithm like simple guassian, K-nearest neighbor, etc.

- Mel-frequency Cepstral Coefficient (MFCCs):

  Pitch is one among the characteristics of a speech signal and is measured in terms of the frequency of the signal. Mel scale can be said as a scale that relates the perceived frequency of a tone to the particular and actual measured frequency. Cepstrum is nothing but the information of rate of change in spectral bands. MFCCs are nothing but the coefficients that make up or structure the Mel-frequency Cepstrum

- Short-time Fourier Transform (STFT):

  The STFT, is a Fourier-related transform which is used to determine the sinusoidal frequency and phase content of local sections of a signal because it changes over time.

This system will be a classifier which will be used for classifying the different type of music based on its genre. A generalized algorithm for this system can be written as:

1) Input the audio files from the dataset.
2) Preprocess the data.
3) Perform the feature extraction on the audio file (features like MFCC).
4) Train the model with the available dataset.
5) Test the model against new data.
6) Is testing successful?
    a) Yes: Classifier is ready to use in an application.
    b) No: Go to step 4 again.
7) Stop.

INTEGRATION AND TESTING

We will be using GTZAN dataset for training and testing of model. This dataset is considered as the standard for genre classification. This dataset contains 9 music genres, each genre has 100 audio clips in .au format. Each audio clip has a length 30 seconds, are 22050Hz Mono 16-bit files.

Input:

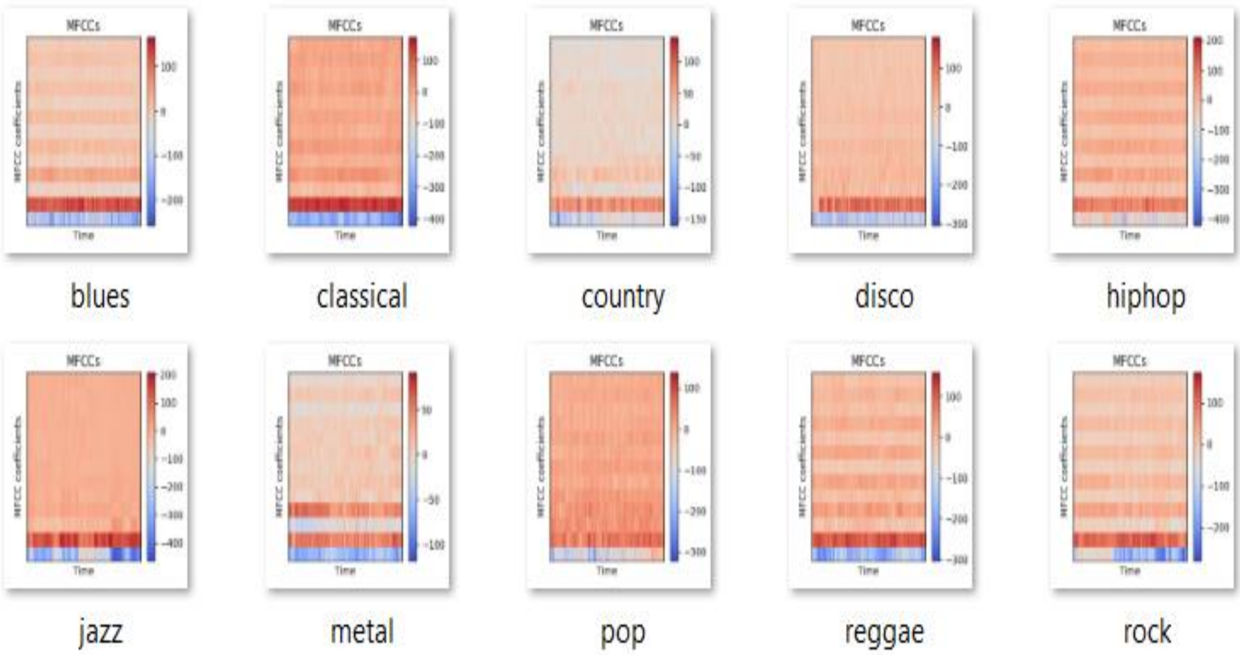
An audio file will be given to the model.

Output:

The model will classify the given audio file into its respective genre.

1. Feature Extraction:

Sample MFCC's 1 audio signal from each music genre:



The features of the audio data will be stored in data.json file which will look like-

```
{
    "mapping": [
            "genre\\blues",
            "genre\\classical",
            "genre\\country",
            …],
    "labels": [
            0,
            1,
            2,
            3,
            …],
    "mfcc": []
}
```

Mapping is a list of genres we have in our dataset.

Labels is a list of labels (in the form of integers) assigned to each of the genre.

Mfcc is a list of Mel-Frequency Cepstral Coefficients of each audio file from the dataset. For each of the audio there will be a list of 13 coefficients. For each genre in the dataset there will be a list of such coefficients for each audio.

2. Model Building:

The model is trained using Convolutional Neural Network (CNN).

The snapshots below show the summary of the model and training of the classifier.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 128, 11, 32)       320
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 64, 6, 32)         0
_________________________________________________________________
batch_normalization (BatchNo (None, 64, 6, 32)         128
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 62, 4, 32)         9248
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 31, 2, 32)         0
_________________________________________________________________
batch_normalization_1 (Batch (None, 31, 2, 32)         128
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 30, 1, 32)         4128
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 15, 1, 32)         0
_________________________________________________________________
batch_normalization_2 (Batch (None, 15, 1, 32)         128
_________________________________________________________________
flatten (Flatten)            (None, 480)               0
_________________________________________________________________
dense (Dense)                (None, 64)                30784
_________________________________________________________________
dropout (Dropout)            (None, 64)                0
_________________________________________________________________
dense_1 (Dense)              (None, 10)                650
=================================================================
Total params: 45,514
Trainable params: 45,322
Non-trainable params: 192
```

C:\Windows\System32\cmd.exe

```
Epoch 1/30
188/188 [==============================] - 8s 41ms/step - loss: 2.8603 - accuracy: 0.1681 - val_loss: 1.9587 - val_accuracy: 0.3264
Epoch 2/30
188/188 [==============================] - 13s 71ms/step - loss: 2.0592 - accuracy: 0.3092 - val_loss: 1.6030 - val_accuracy: 0.4266
Epoch 3/30
188/188 [==============================] - 9s 46ms/step - loss: 1.7271 - accuracy: 0.3946 - val_loss: 1.4648 - val_accuracy: 0.4793
Epoch 4/30
188/188 [==============================] - 8s 43ms/step - loss: 1.5660 - accuracy: 0.4527 - val_loss: 1.3570 - val_accuracy: 0.5087
Epoch 5/30
188/188 [==============================] - 9s 48ms/step - loss: 1.4864 - accuracy: 0.4687 - val_loss: 1.2872 - val_accuracy: 0.5287
Epoch 6/30
188/188 [==============================] - 8s 42ms/step - loss: 1.4026 - accuracy: 0.4898 - val_loss: 1.2143 - val_accuracy: 0.5708
Epoch 7/30
188/188 [==============================] - 8s 43ms/step - loss: 1.3335 - accuracy: 0.5369 - val_loss: 1.1811 - val_accuracy: 0.5828
Epoch 8/30
188/188 [==============================] - 9s 46ms/step - loss: 1.2829 - accuracy: 0.5282 - val_loss: 1.1321 - val_accuracy: 0.6015
Epoch 9/30
188/188 [==============================] - 8s 44ms/step - loss: 1.2515 - accuracy: 0.5521 - val_loss: 1.0946 - val_accuracy: 0.6175
Epoch 10/30
188/188 [==============================] - 8s 41ms/step - loss: 1.1892 - accuracy: 0.5821 - val_loss: 1.0725 - val_accuracy: 0.6322
Epoch 11/30
188/188 [==============================] - 7s 39ms/step - loss: 1.1122 - accuracy: 0.6119 - val_loss: 1.0609 - val_accuracy: 0.6262
Epoch 12/30
188/188 [==============================] - 8s 43ms/step - loss: 1.1197 - accuracy: 0.6009 - val_loss: 1.0266 - val_accuracy: 0.6395
Epoch 13/30
188/188 [==============================] - 8s 45ms/step - loss: 1.0453 - accuracy: 0.6391 - val_loss: 0.9947 - val_accuracy: 0.6656
Epoch 14/30
188/188 [==============================] - 10s 55ms/step - loss: 1.0058 - accuracy: 0.6487 - val_loss: 0.9802 - val_accuracy: 0.6682
Epoch 15/30
188/188 [==============================] - 9s 50ms/step - loss: 1.0077 - accuracy: 0.6402 - val_loss: 0.9655 - val_accuracy: 0.6642
Epoch 16/30
188/188 [==============================] - 10s 52ms/step - loss: 0.9524 - accuracy: 0.6601 - val_loss: 0.9458 - val_accuracy: 0.6802
Epoch 17/30
188/188 [==============================] - 8s 40ms/step - loss: 0.9508 - accuracy: 0.6617 - val_loss: 0.9453 - val_accuracy: 0.6709
Epoch 18/30
188/188 [==============================] - 7s 39ms/step - loss: 0.9292 - accuracy: 0.6760 - val_loss: 0.9100 - val_accuracy: 0.6909
Epoch 19/30
188/188 [==============================] - 8s 41ms/step - loss: 0.9004 - accuracy: 0.6850 - val_loss: 0.9035 - val_accuracy: 0.6903
Epoch 20/30
188/188 [==============================] - 8s 45ms/step - loss: 0.8524 - accuracy: 0.7024 - val_loss: 0.8939 - val_accuracy: 0.7016
Epoch 21/30
188/188 [==============================] - 9s 48ms/step - loss: 0.8596 - accuracy: 0.6912 - val_loss: 0.9001 - val_accuracy: 0.6856
Epoch 22/30
188/188 [==============================] - 8s 42ms/step - loss: 0.8353 - accuracy: 0.7084 - val_loss: 0.8694 - val_accuracy: 0.7076
Epoch 23/30
188/188 [==============================] - 8s 41ms/step - loss: 0.7878 - accuracy: 0.7351 - val_loss: 0.8567 - val_accuracy: 0.7083
Epoch 24/30
188/188 [==============================] - 7s 39ms/step - loss: 0.8062 - accuracy: 0.7164 - val_loss: 0.8498 - val_accuracy: 0.7049
Epoch 25/30
188/188 [==============================] - 7s 39ms/step - loss: 0.7643 - accuracy: 0.7264 - val_loss: 0.8715 - val_accuracy: 0.7009
```

PERFORMANCE ANALYSIS

Existing system is a manual entry of the music files according to their genres. It will be a tedious job to maintain the record of all the genres and respective songs. The human effort is more here. The retrieval of the knowledge is not quite easy because the records are to be maintained. It also requires correct input into the respective field. Suppose the wrong inputs are entered, it will become more difficult to deal with it.

To overcome the drawbacks of the prevailing system, the proposed system has been developed. This system aims to reduce the manual work and save time to generate accurate results. This proposed system will help to classify the songs into their respective genres. It will also help to classify a new song into its genre accurately.

The development of this system is economically feasible. It is cost effective in the sense that the technical requirement for the system is also economic and it does not use any other additional hardware or software. The working of the system is sort of easy to use and learn. There is no any need of special training for using this system.

The system developed gives the accuracy of 65-70% for training, testing and validation dataset. The snapshots below show the accuracy of the training, validation and testing data.

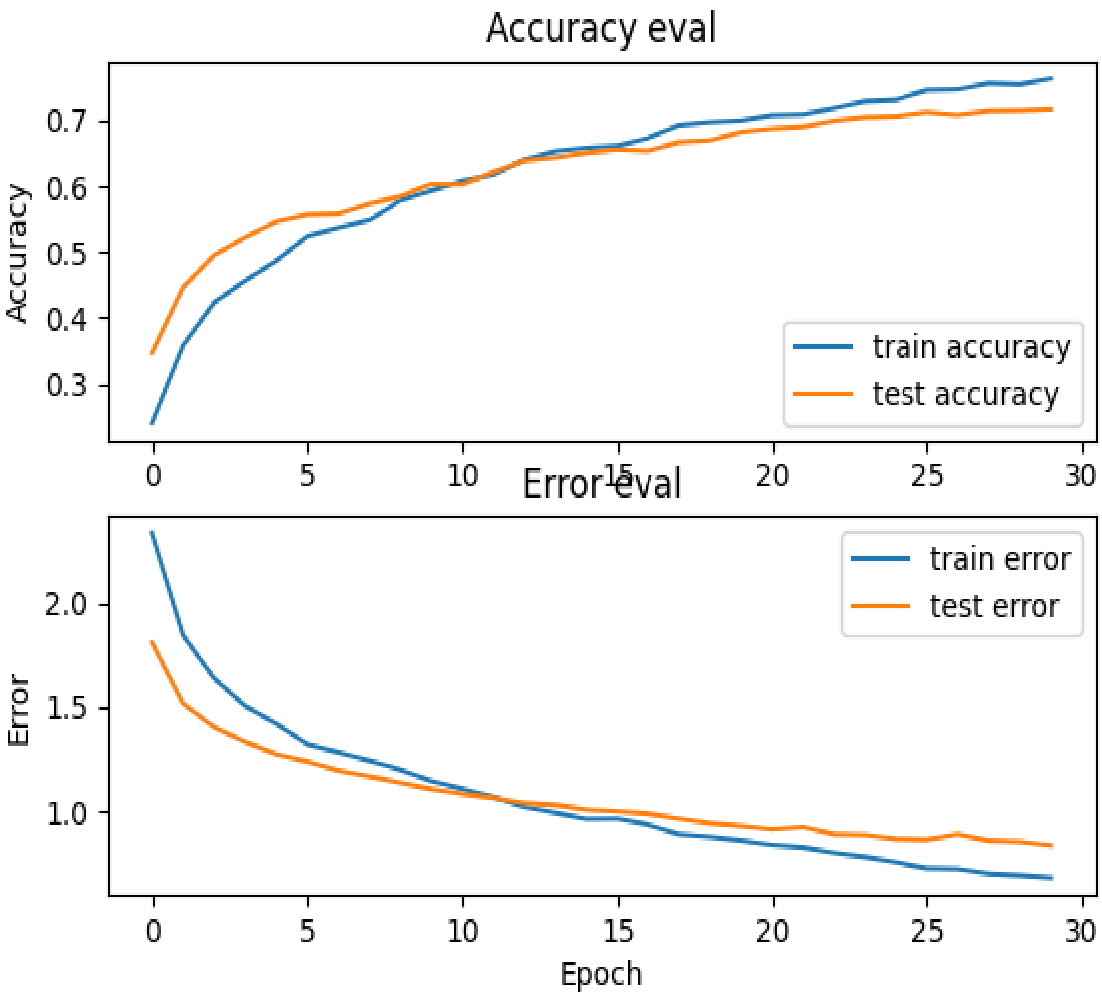
```
Epoch 29/30
188/188 [==============================] - 9s 49ms/step - loss: 0.6964 - accuracy: 0.7574 - val_loss: 0.8563 - val_accuracy: 0.7150
Epoch 30/30
188/188 [==============================] - 9s 50ms/step - loss: 0.6857 - accuracy: 0.7605 - val_loss: 0.8381 - val_accuracy: 0.7170
79/79 - 1s - loss: 0.7802 - accuracy: 0.7257
```

a. Training & Validation accuracy.

```
Test accuracy: 0.7256708145141602
Target: 9, Predicted label: [9]
```

b. Testing accuracy.

The figure below is a graphical representation of accuracy and loss of the model.



APPLICATIONS

Music genre labels are useful categories to arrange and classify songs, albums, and artists into broader groups that share similar musical characteristics.

We use this classification for grouping our MP3 downloads. The categories distinguish each sort of music by its form and elegance. This kind of machine classification is not just going to be useful for your own purpose, but it also has its actual use within the music industry.

Many companies nowadays use music classification to put recommendations to their customers for instance Spotify, Soundcloud. These are the online music streaming services which classify and recommend the songs to the user based on their interest. Genre Classification is implemented to perform these tasks.

Music classification can also be used as a product like Shazam. Shazam is an application which can identify music, movies, advertising, and television shows, based on a brief sample played using the microphone on the device.

This system can be further used in music recommendation systems. Most of the online streaming applications may need to recognize the taste of music their users have and recommend them the songs based on their taste. Genre classification might be helpful in such scenarios.

We can use this system to generate a Music Genre Prediction System. Using this classifier we can develop a website/application for prediction of the genre of a given audio file.

INSTALLATION GUIDE AND USER MANUAL

Installation guide is a technical communication document. The intention behind providing such guide is to help people on the way to install a specific program. User manual is nothing but the instructions given to the user about systematic ways of using the project.

This project requires the following python modules to be installed on your machine.

- Tensorflow
- Numpy
- Librosa
- Matplotlib

To install these modules use the command:
pip install <moduleName>

Here we are using the GTZAN dataset for training our model. A short information of the dataset is as follows:

- This dataset was used by G. Tzanetakis and P. Cook for their well-known paper in genre classification named “Musical genre classification of audio signals”.
- The files were collected from a variety of sources including personal CDs, radio, microphone recordings, in order to represent a variety of recording conditions.
- The dataset consists of 1000 audio tracks each of which is 30 seconds long. It contains 10 genres and each one is represented by 100 tracks. The all tracks are in .wav format.
- Official web-page: [marsyas.info](http://marsyas.info/download/data_sets]
  Download size: Approximately 1.2GB
  Download link: http://opihi.cs.uvic.ca/sound/genres.tar.gz

ETHICS

As a Computer Science & Engineering Student, I believe it is unethical to,

1. Surf the internet for personal interest and non-class related purposes during classes.
2. Make a copy of software for personal or commercial use.
3. Make a copy of software for a friend.
4. Loan CDs of software from the internet.
5. Download pirated software from the internet.
6. Distribute pirated software from the internet.
7. Buy software with a single user license and then install it on multiple computers.
8. Share a pirated copy of software.
9. Install a pirated copy of a software.

REFERENCES

- https://www.sciencedirect.com/science/article/pii/S1110866512000151#:~:text=Tests%20were%20carried%20out%20with,set%20of%20parameters%20for%20each

- https://www.irjet.net/archives/V6/i5/IRJET-V6I5174.pdf

- https://cse.iitk.ac.in/users/cs365/2015/_submissions/archit/report.pdf

- https://www.researchgate.net/publication/324218667_Music_Genre_Classification_using_Machine_Learning_Techniques

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	878	Date	May 25,2021
Characters	5537	Excluded URL	

5% Plagiarism	95% Unique	2 Plagiarized Sentences	39 Unique Sentences

Content Checked For Plagiarism

There are number of music streaming services like Apple music, Spotify, Gaana, YouTube etc. These services always try to provide best of the experience to their users by providing amazing playlists that include songs that they like as well as introducing the users to the similar songs that they may have never listened to before. We can view a song as, depending on length, approximately 1.3 million data-points. In order to classify songs as similar we'd like to look at these data-points and it becomes increasingly difficult to try to do this with the raw data especially when new songs are introduced every day. Also, similarity between songs is a difficult thing to define because the parameters won't to describe how similar two songs are inherently subjective and cannot be easily translated into an algorithm.
Music plays a very vital role in people's lives. Music can be said as a thread which binds the like-minded people together and forms different communities. These communities can be identified by the type of songs they compose or even listen to. To identify such communities it is important to be able to identify the similarities in different songs. One way of doing this is to identify genre of songs, which is identified by some characteristics of the music such as rhythmic structure, harmonic content and instrumentation. To make the process easier automatic classification of songs supported genre will be helpful. So that is why in today's world automatic classification of music has become an important aspect. Being able to automatically classify and supply tags to the music present during a user's library, based on genre, would be beneficial for audio streaming services.
A. PROBLEM DEFINITION
Currently there are huge number of songs and they are to be classified manually according to their respective genres. The proposed system is going to classify the songs into correct genre with the help of neural networks.
In this project we are going to create a classifier such that it will be able to classify the music based on its genre. The very popular GTZAN dataset will be used here to train the model. After training, model will be able to predict the genre of the music accurately.
B. AIM & OBJECTIVE OF THE PROJECT
The aim of this project is to implement an algorithm that reduces the amount of data-points we'd like to work with and build a classifier. The classifier should be ready to categorize a given music track into the proper genre supported how similar the extracted features are to the music samples we want to construct the classifier for.
Music classification is a way to accurately perceive the things we hear and describe them to others. Music Genres are important. They represent one among the foremost important and valuable tools we have to appreciate, understand, and communicate with each other about the music we always hear. There is no doubt that this system will be able to classify music according to genre and will be helpful to Music Information Retrieval systems.
Objectives:
1. To build a model that classifies Music into its respective genre based on various different features, instead of manually entering the genres.
2. To develop a model with good accuracy so that it classifies the new music into its genre correctly.
C. SCOPE & LIMITATION OF THE PROJECT
The acquisition of reliable ground truth could also be a key requirement of coaching effective genre classifiers. It's been suggested that only limited agreement are often achieved among human annotators when classifying music by genre, which such limits impose an unavoidable ceiling on automatic genre classification performance. Individuals can't only differ on how they classify a given recording, but they also are going to be ready to differ in terms of the pool of genre labels from which they choose. only a few genres have clear definitions, and what information is out there out there's

usually confusing, ambiguous and inconsistent from source to source. Usually there's significant overlapping between genres. Individual recordings can possibly belong to multiple genres of varying degrees. There are often complex relationships between genres. Some genres are broad while some are narrow. Furthermore, genres usually form multiple different clusters.

D. TIMELINE OF THE PROJECT

We have used classic life cycle paradigm also called “Water Fall Model”. Software engineering is a sequential approach to software development that begins at the system level and progress through analysis, design, coding, testing and maintenance. We had completed software requirement analysis by the mid of September 2020 which encompasses both system and software requirement gathering. By the end of December 2020 we had completed project planning and design. On the basis of design prepared in the previous stage by the end of March 2021 we completed coding stage.

After completion of coding stage the important part in the software development which is testing phase carried out in first week of April 2021.Various criteria of testing were taken into account which includes unit testing, integration testing, validation testing and system testing. First, each and every module of the project was tested under the unit testing. After the unit testing, integration testing was carried out by integrating all module tested in unit testing. After unit testing the module prepared was cross checked with the design.

Sources	Similarity
(PDF) Online Distributor System Using Android OS ... We have used classic life cycle paradigm also called "Water Fall Model". For software engineering which is sequential approach to software development that begins at the system level and progress through Analysis, Design, Coding, Testing and Maintenance. Fig. 1 Architecture of the system Here in this architecture, system consists of 3 phases: registration phase, login phase and verification phase. During registration, distributor registers with his new ID and … https://www.academia.edu/9220960/Online_Distributor_System_Using_Android_OS	4%
Linear sequential model for software engineering is also ... Linear sequential model for software engineering is also known as waterfall model. The linear sequential model suggests a systematic sequential approach to software development that begins at the system level and progresses through analysis, design, coding, testing, and support. https://www.toppr.com/ask/question/linear-sequential-model-for-software-engineering-is-also-known-as/	3%

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	855	Date	May 25,2021
Characters	5930	Excluded URL	

2% Plagiarism	98% Unique	1 Plagiarized Sentences	41 Unique Sentences

Content Checked For Plagiarism

The problem of Music Genre Classification has been approached in various ways. Tzanetakis, Sam Clark, and other papers have approached this problem specific to certain algorithms. John cast, suggest that features coupled with MFCCs improve classification accuracy. Martin, suggests that the temporal modulations of MFCCs are important for classification and its performance is better in comparison with the Standard low-level feature set because of its increased ability to classify background crowd noise and popular music. C H lee, proposed a novel feature set for music genre classification based on cepstral (MFCC) features which achieves higher classification accuracy. Most work on this problem is done by using specific data preprocessing techniques like FFT or MFCC exclusively to convert audio data and the training of classification algorithms is done based on these data values. We believe there is no prior work analyzing how various algorithms perform using MFCC and FFT data values to the extent of our knowledge. Hence this paper is a study on such an analysis. From the analysis it is clear that using MFCC data values gives better results overall than using FFT values. The Simpler algorithms such as Logistic Regression and Kth Nearest Neighbors did fairly well in comparison to superior algorithms such as Recurrent Neural Networks and Support Vector Machines. The highest accuracy reached was 86% using Neural Networks.
B. INVESTIGATION OF CURRENT PROJECT AND RELATED WORK
The basis of any kind of automatic audio analysis system is that the extraction of feature vectors. A large number of different kind of feature sets, mainly deriving from the world of speech recognition, are proposed to represent audio signals. Typically they're supported some sort of time-frequency representation. Although an entire overview of audio feature extraction is beyond the scope of this paper, some relevant representative audio feature extraction references are provided. Automatic classification of audio has also an extended history originating from speech recognition. Mel-frequency cepstral coefficients (MFCC), are a group of perceptually motivated features that have been widely utilized in speech recognition. These MFCC features provide a brief representation of the spectral envelope, such that most of the signal energy is concentrated within the first coefficients.
More recently, audio classification techniques that include non-speech signals have been proposed and evolved with time. Most of those systems target the classification of broadcast news and video in broad categories like music, speech, and environmental sounds. The problem of discrimination between music and speech has received considerable attention from the first work of Saunders where simple thresholding of the typical zero-crossing rate and energy features is used, to the work of Scheirer and Slaney where multiple features and statistical pattern recognition classifiers are carefully evaluated. In D. Kimber and L. Wilcox, “Acoustic segmentation for audio browsers” project, audio signals are segmented and classified into “music,” “speech,” “laughter,” and non-speech sounds using cepstral coefficients and a hidden Markov model (HMM). A heuristic rule-based system for the segmentation and classification of audio signals from movies or TV programs supported the time-varying properties of simple and straightforward features is proposed in T. Zhang and J. Kuo’s “Audio content analysis for online audio-visual data segmentation and classification”. Signals are classified into two main and broad groups of music and non-music. They are further subdivided into harmonic environmental sound, song, speech with music which is considered into a music category, and pure speech and non-harmonic environmental sound which is considered into non-music category. Berenzweig and Ellis affect the harder problem of locating voice segments in musical signals. In their system, the phoneme activation output of an automatic speech recognition system is employed because the feature vector for classifying singing segments.
Another sort of non-speech audio arrangement involves isolated instrument sounds and sound effects. In the

pioneering work of Wold automatic retrieval, classification and clustering of musical instruments, sound effects, and environmental sounds using automatically extracted features is explored. The features utilized in their system are statistics (mean, variance, autocorrelation) over the entire sound file of short time features like pitch, amplitude, brightness, and bandwidth. Using an equivalent dataset various other retrieval and classification approaches are proposed.

REQUIREMENT ANALYSIS

RS1: The hardware requirement:

• Personal computer with standard configuration.
• RAM – Minimum 4 GB.
• Processor – Intel Core i3 or higher.

RS2: The Software requirement:

• Operating system – Windows 8 or higher.
• Python installed on machine or Anaconda Distribution of Python 3.

RS3: The system should have the sufficient amount of dataset.

The sufficient amount of data is required so that the model can be trained with more accuracy.

RS4: The system should have Librosa library for converting audio files into melspectrogram.

Librosa is a Python module to analyze audio signals in general but geared more towards music.

RS5: The system should have other basic libraries for processing of the audio files.

To perform different operations on the extracted features of the data, different modules like numpy, matplotlib, pandas, etc are required.

RS6: The system should have a database to store the Feature Vector.

The system should be able to store the features extracted from each sample of audio so that it will be easier for further processing.

Sources	Similarity
(PDF) Emotion Recognition from Speech Using the Bag-of-Visual … 2019. 2. 4. · Zhang, T.; Kuo, C.C.J. Audio content analysis for online audiovisual data segmentation and classification. https://www.researchgate.net/publication/330892119_Emotion_Recognition_from_Speech_Using_the_Bag-of-Visual_Words_on_Audio_Segment_Spectrograms	4%

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	1000	Date	May 25,2021
Characters	6445	Excluded URL	

4% Plagiarism	96% Unique	2 Plagiarized Sentences	54 Unique Sentences

Content Checked For Plagiarism

There are different number of approaches to develop the system for classification of Music based on the genre. One of them is:
• The ground-breaking work, based on Naive Bayesian and Neural Network approaches, identifies one out of four styles of a musician improvisation. There work included testing a performer's ability to consistently produce intentional styles which should be different from each other. A database was created to train the classifiers. While classifying among the four styles an accuracy of 98% was achieved. Then while using eight classifiers, trained model used to return "yes" or "no" for eight different styles. They got an overall accuracy of 77–90%.
In this project we will use feature set which will have features like Short-time Fourier Transform (STFT) , Mel-frequency Cepstral Coefficients (MFCCs), etc. These features will help to represent the audio files in our dataset. With the help of these vectors we will train our classifier by using an algorithm like simple guassian, K-nearest neighbor, etc.
• Mel-frequency Cepstral Coefficient (MFCCs):
Pitch is one among the characteristics of a speech signal and is measured in terms of the frequency of the signal. Mel scale can be said as a scale that relates the perceived frequency of a tone to the particular and actual measured frequency. Cepstrum is nothing but the information of rate of change in spectral bands. MFCCs are nothing but the coefficients that make up or structure the Mel-frequency Cepstrum
• Short-time Fourier Transform (STFT):
The STFT, is a Fourier-related transform which is used to determine the sinusoidal frequency and phase content of local sections of a signal because it changes over time.
This system will be a classifier which will be used for classifying the different type of music based on its genre. A generalized algorithm for this system can be written as:
1) Input the audio files from the dataset.
2) Preprocess the data.
3) Perform the feature extraction on the audio file (features like MFCC).
4) Train the model with the available dataset.
5) Test the model against new data.
6) Is testing successful?
a) Yes: Classifier is ready to use in an application.
b) No: Go to step 4 again.
7) Stop.
We will be using GTZAN dataset for training and testing of model. This dataset is considered as the standard for genre classification. This dataset contains 9 music genres, each genre has 100 audio clips in .au format. Each audio clip has a length 30 seconds, are 22050Hz Mono 16-bit files.
Input:
An audio file will be given to the model.
Output:
The model will classify the given audio file into its respective genre.
1. Feature Extraction:
The features of the audio data will be stored in data.json file which will look like-
{

"mapping": [
"genre\\blues",
"genre\\classical",
"genre\\country",
…],
"labels": [
0,
1,
2,
3,
…],
"mfcc": []
}

Mapping is a list of genres we have in our dataset.
Labels is a list of labels (in the form of integers) assigned to each of the genre.
Mfcc is a list of Mel-Frequency Cepstral Coefficients of each audio file from the dataset. For each of the audio there will be a list of 13 coefficients. For each genre in the dataset there will be a list of such coefficients for each audio.
The model is trained using Convolutional Neural Network (CNN).
Existing system is a manual entry of the music files according to their genres. It will be a tedious job to maintain the record of all the genres and respective songs. The human effort is more here. The retrieval of the knowledge is not quite easy because the records are to be maintained. It also requires correct input into the respective field. Suppose the wrong inputs are entered, it will become more difficult to deal with it.
To overcome the drawbacks of the prevailing system, the proposed system has been developed. This system aims to reduce the manual work and save time to generate accurate results. This proposed system will help to classify the songs into their respective genres. It will also help to classify a new song into its genre accurately.
The development of this system is economically feasible. It is cost effective in the sense that the technical requirement for the system is also economic and it does not use any other additional hardware or software. The working of the system is sort of easy to use and learn. There is no any need of special training for using this system.
The system developed gives the accuracy of 65-70% for training, testing and validation dataset. The snapshots below show the accuracy of the training, validation and testing data.
Music genre labels are useful categories to arrange and classify songs, albums, and artists into broader groups that share similar musical characteristics.
We use this classification for grouping our MP3 downloads. The categories distinguish each sort of music by its form and elegance. This kind of machine classification is not just going to be useful for your own purpose, but it also has its actual use within the music industry.
Many companies nowadays use music classification to put recommendations to their customers for instance Spotify, Soundcloud. These are the online music streaming services which classify and recommend the songs to the user based on their interest. Genre Classification is implemented to perform these tasks.
Music classification can also be used as a product like Shazam. Shazam is an application which can identify music, movies, advertising, and television shows, based on a brief sample played using the microphone on the device.
This system can be further used in music recommendation systems. Most of the online streaming applications may need to recognize the taste of music their users have and recommend them the songs based on their taste. Genre classification might be helpful in such scenarios.
We can use this system to generate a Music Genre Prediction System. Using this classifier we can develop a website/application for prediction of the genre of a given audio file.

Sources	Similarity
Short-time Fourier transform - Wikipedia The Short-time Fourier transform (STFT), is a Fourier-related transform used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over ... https://en.wikipedia.org/wiki/Short-time_Fourier_transform	7%
Shazam — Google Arts & Culture Shazam is an application that can identify music, movies, advertising, and television shows, based on a short sample played and using the microphone on the... https://artsandculture.google.com/entity/shazam/m0406_1w?hl=en	2%

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	289	Date	May 25,2021
Characters	2008	Excluded URL	

0% Plagiarism	100% Unique	0 Plagiarized Sentences	14 Unique Sentences

Content Checked For Plagiarism

Installation guide is a technical communication document. The intention behind providing such guide is to help people on the way to install a specific program. User manual is nothing but the instructions given to the user about systematic ways of using the project.
This project requires the following python modules to be installed on your machine.
• Tensorflow
• Numpy
• Librosa
• Matplotlib
To install these modules use the command:
pip install
Here we are using the GTZAN dataset for training our model. A short information of the dataset is as follows :
• This dataset was used by G. Tzanetakis and P. Cook for their well-known paper in genre classification named "Musical genre classification of audio signals".
• The files were collected from a variety of sources including personal CDs, radio, microphone recordings, in order to represent a variety of recording conditions.
• The dataset consists of 1000 audio tracks each of which is 30 seconds long. It contains 10 genres and each one is represented by 100 tracks. The all tracks are in .wav format.
• Official web-page: [marsyas.info](http://marsyas.info/download/data_sets]
Download size: Approximately 1.2GB
Download link: http://opihi.cs.uvic.ca/sound/genres.tar.gz

ETHICS
As a Computer Science & Engineering Student, I believe it is unethical to,
1. Surf the internet for personal interest and non-class related purposes during classes.
2. Make a copy of software for personal or commercial use.
3. Make a copy of software for a friend.
4. Loan CDs of software from the internet.
5. Download pirated software from the internet.
6. Distribute pirated software from the internet.
7. Buy software with a single user license and then install it on multiple computers.
8. Share a pirated copy of software.
9. Install a pirated copy of a software.

Sources	Similarity