Music Genre Classifier

A Project Report Submitted

for the Course of

Minor Project – 2

In

Third year – Sixth Semester of

Bachelor of Technology

In

Computer Science Engineering

With Specialization

In

Artificial Intelligence and Machine Learning

Under

Dr. Piyush Chauhan
Assistant Professor - Senior Scale
Department of Informatics
UPES, Dehradun

Prepared by

Name	SAP ID	Specialization
Abhijeet Jain	500075391	AI & ML
Aryan Verma	500076058	AI & ML
Wamiq Zafar	500075916	IT Infrastructure

DEPARTMENT OF INFORMATICS
SCHOOL OF COMPUTER SCIENCE
UNIVERSITY OF PETROLEUM AND ENERGY STUDIES, BIDHOLI,
DEHRADUN, UTTRAKHAND, INDIA

Table of Contents

1. INTRODUCTION

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 such as Soundcloud and Spotify – a 2016 number suggests that tens of thousands of songs were released every month on Spotify – the need for accurate meta-data required for database management and search/storage purposes 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/purchasing service, and the capacity for statistical analysis that correct and complete labeling of music and audio provides is essentially limitless.

1.1 Purpose of the Project

The objective of automating the music classification is to make the selection of songs quick and less cumbersome. If one has to manually classify the songs or music, one has to listen to a whole lot of songs and then select the genre. This is not only time-consuming but also difficult. Automating music classification can help to find valuable data such as trends, popular genres, and artists easily. Determining music genres is the very first step towards this direction.

1.2 Target Beneficiary

Classification results can be used to support sociological and psychological research into how humans construct the notion of musical similarity and form musical groupings, and how this compares to the —objective“ truth produced by computerized classifiers.

1.3 Project Scope

An automatic genre classification algorithm could greatly increase efficiency for music databases such as AllMusic. It could also help music recommender systems and playlist generators that companies like Spotify and Pandora use.

1.4 References

[1] https://medium.com/bisa-ai/music-genre-classification-using-convolutional-neural-network-7109508ced47

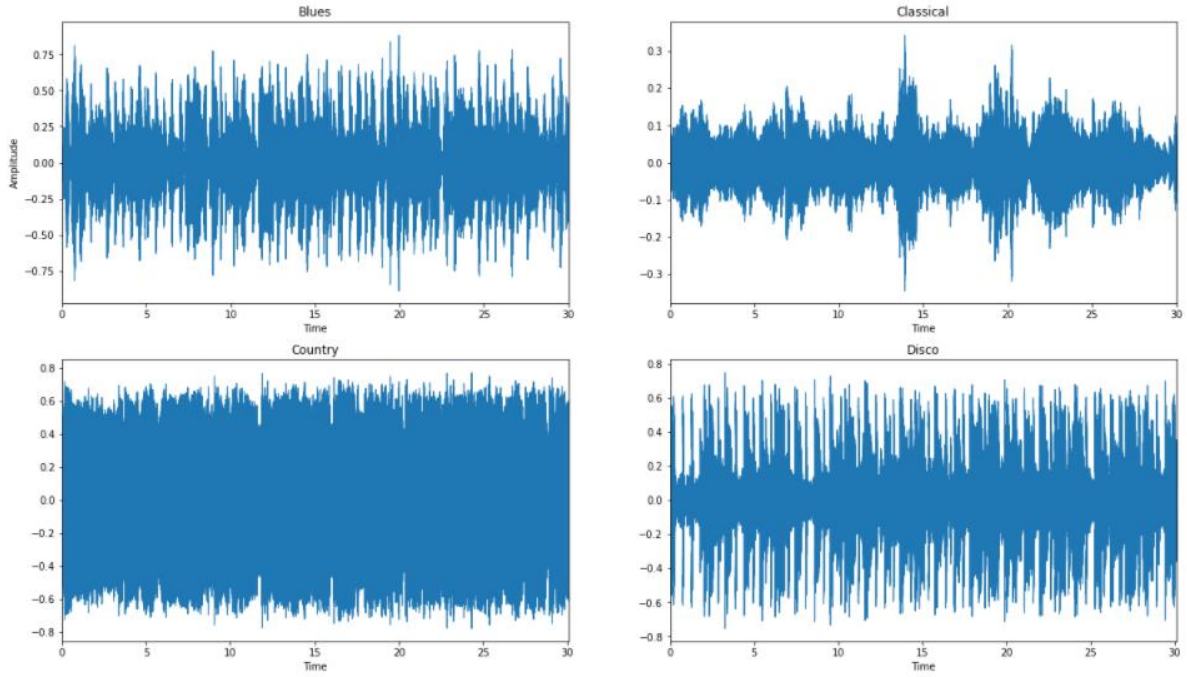
[2] http://cs229.stanford.edu/proj2018/report/21.pdf

2. PROJECT DESCRIPTION

2.1 Reference Algorithm

Procedure:
Visualizing Audio Files : The following methods are used for visualizing the audio files we were using Librosa library :

1. Plot Raw Wav Files



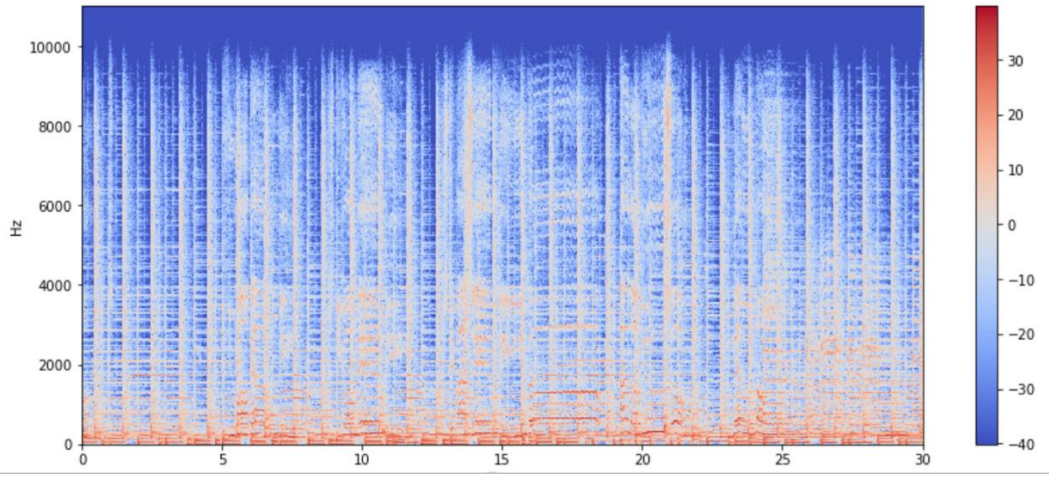
Waveforms are visual representations of sound as time on the x-axis and amplitude on the y-axis.

2. Spectrograms

A spectrogram is a visual way of representing the signal loudness of a signal over time at various frequencies present in a particular waveform. Not only can one see whether there is more or less energy at, for example, 2 Hz vs 10 Hz, but one can also see how energy levels vary over time.

```
[37]: stft = librosa.stft(data)
      stft_db = librosa.amplitude_to_db(abs(stft))
      plt.figure(figsize=(14, 6))
      librosa.display.specshow(stft_db, sr=sr, x_axis='time', y_axis='hz')
      plt.colorbar()
```

```
[37]: <matplotlib.colorbar.Colorbar at 0x7fa44f047160>
```



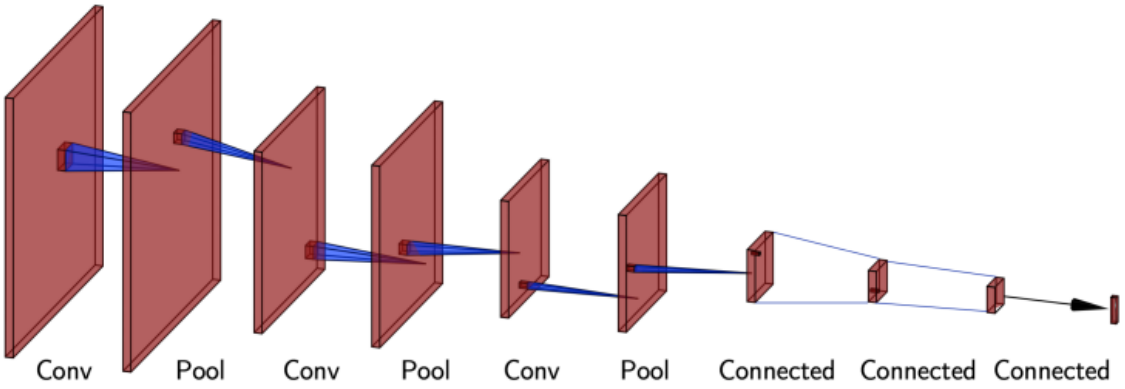
Feature Extraction: Preprocessing of data is required before we finally train the data. We will try to focus on the last column that is 'label' and will encode it with the function LabelEncoder() of sklearn.preprocessing.

Scaling the Features : Standard scaler is used to standardize features by removing the mean and scaling to unit variance.

The standard score of sample x is calculated as:

$$z = (x - u) / s$$

Building the Model : CNN, using 3 convolution layers, each with its own max pool and regularization, feeding into 3 fully connected layers with ReLU activation, softmax output, and cross entropy loss. Most of the equations can be found above, and our architecture is visually presented below: This approach involves convolution windows that scan over the input data and output the sum of the elements within the window. This then gets fed into a max pool layer that selects the maximum element from another window.This was implemented with TensorFlow and Keras.



2.2 Characteristics of Data

For this project, the dataset that we will be working with is GTZAN Genre Classification dataset which consists of 1,000 audio tracks, each 30 seconds long. It contains 10 genres, each represented by 100 tracks.

The dataset has the following folders:

- **Genres original** — A collection of 10 genres with 100 audio files each, all having a length of 30 seconds (the famous GTZAN dataset, the MNIST of sounds)
- **Images original** — A visual representation for each audio file. One way to classify data is through neural networks because NN's usually take in some sort of image representation.
- **2 CSV files** — Containing features of the audio files. One file has for each song (30 seconds long) a mean and variance computed over multiple features that can be extracted from an audio file. The other file has the same structure, but the songs are split before into 3 seconds audio files.

2.3 SWOT Analysis

- Strength – The model accuracy can be increased by further increasing the epochs but after a certain period, we may achieve a threshold, so the value should be determined accordingly. The accuracy we achieved for the test set is 92.93 percent which is very decent.

- Weakness – There is no dedicated GUI for our project as of now which makes it not useful for lay men also training takes too long on low end devices .

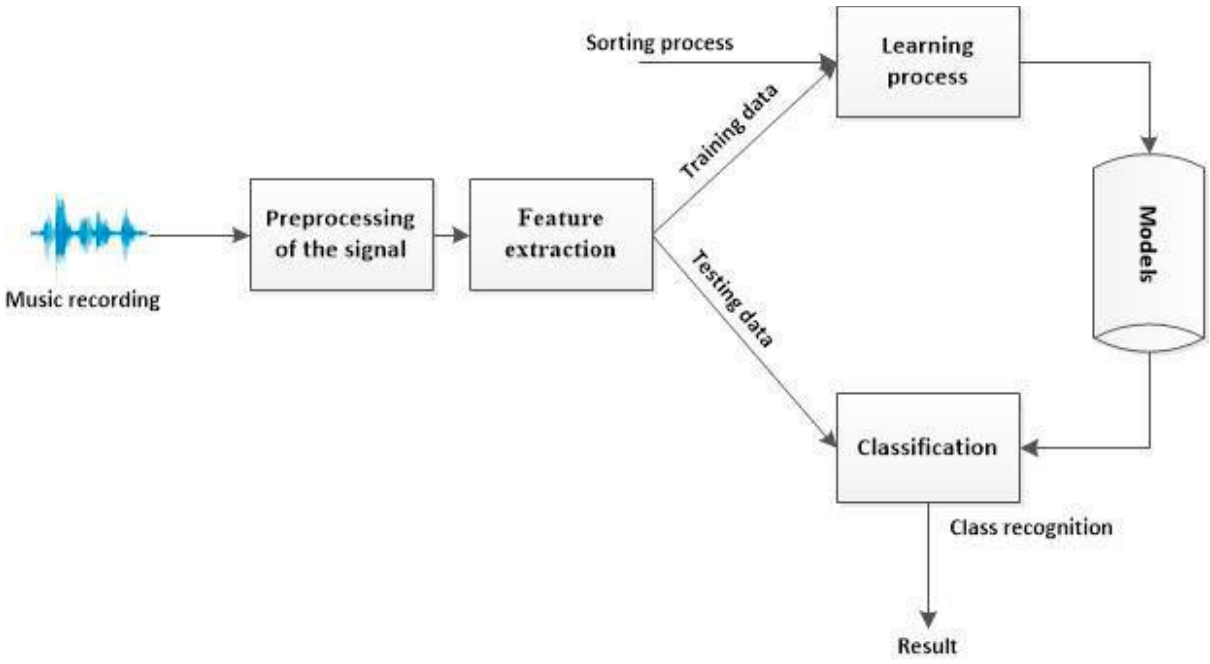
- Opportunities – In our project, we are working on Convolutional Networks and handling audio files with python which can be used in automated music composing applications in the near future .

- Threats – The image data is quite large so while training the data the kernel might stop working.

2.4 Project Features

- Forms a basic step for a strong music recommendation system .
- The accuracy we achieved for the test set is 92.93 %
- This project forms the basis for higher research on audio files using CNN.

2.4 Design Diagrams



3. SYSTEM REQUIREMENTS

3.1 User Interface

- There is no integrated user interface .

3.2 Software Interface

- Windows OS
- Python 3.8
- Google Colab and Jupyter Notebook

3.3 Database Interface

- Refer section 2.2

3.4 Protocols

- No protocols have been used

4. NON-FUNCTIONAL REQUIREMENTS

4.1 Performance requirements

- The proposed mechanism should be able to interface with different operating systems.
- The proposed technique should be accurate.
- The proposed technique should be better than the existing system.

4.2 Security requirements

- There are no specific security requirements for our proposed technique.

4.3 Software Quality Attributes

- The proposed technique should not crash under any circumstances.

5. RESULTS

- For the CNN model, we had used the Adam optimizer for training the model. The epoch that was chosen for the training model is 600.
- All of the hidden layers are using the RELU activation function and the output layer uses the softmax function. The loss is calculated using the sparse_categorical_crossentropy function.
- Dropout is used to prevent overfitting.
- We chose the Adam optimizer because it gave us the best results after evaluating other optimizers.
- The model accuracy can be increased by further increasing the epochs but after a certain period, we may achieve a threshold, so the value should be determined accordingly.
- The accuracy we achieved for the test set is 92.93 percent which is very decent.

Model Evaluation

```
test_loss,test_acc=model.evaluate(X_test,y_test,batch_size=128)
print("The test loss is ",test_loss)
print("The best accuracy is: ",test_acc*100)
```

```
26/26 [==============================] - 0s 2ms/step - loss: 0.6207 - accuracy: 0.9160
The test loss is  0.6207043528556824
The best accuracy is:  91.59842133522034
```

- So we come to the conclusion that Neural Networks are very effective in machine learning models. Tensorflow is very useful in implementing Convolutional Neural Network (CNN) that helps in the classifying process.

APPENDIX A: GLOSSARY

- **Librosa library :** Librosa is a python package for music and audio analysis. It provides the building blocks necessary to create music information retrieval systems.

- **Feature Extraction :** We then extracted meaningful features from audio files. We will choose five features to classify our audio clips, i.e., Mel-Frequency Cepstral Coefficients, Spectral Centroid, Zero Crossing Rate, Chroma Frequencies, Spectral Roll-off. All the features are then appended into a .csv file so that classification algorithms can be used.

- **Classification :** Once the features have been extracted, we can use existing classification algorithms to classify the songs into different genres. You can either use the spectrogram images directly for classification or can extract the features and use the classification models on them.

CODE WITH OUTPUT