

Genre Classification

Aanya Verma-200050001

Durgam Latha-200050037

Kumari Pragati Gupta-200050063

Mukkara Gracy paul-200050079

Date : 24/11/2022

References

- Dataset :

GTZAN, <https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification?resource=download-directory>

- Models

- CNN, <https://www.clairvoyant.ai/blog/music-genre-classification-using-cnn>
- KNN, <https://www.analyticsvidhya.com/blog/2022/03/music-genre-classification-project-using-machine-learning-techniques/>

Problem Statement

Music genre classification forms a basic step for building strong recommendation systems and many others. So here, we aim to build a machine learning model, which classifies music samples into different genres. We will take input both as audio and image.

Expected Result

- Label music genre (in total of 10)
Blues, Classical, Country, Disco, Hip-hop, ... so on.
- Compared Implementation on KNN and CNN

Major Features

- Spectrograms
- Spectral Rolloff
- Chroma feature
- Zero Crossing rate

CNN

- Data: Audio + Image (using feature_3_sec, csv file)
- Librosa, load and decode the audio as a time series y .
Sr = sampling rate
- 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. So, bit intuitive to get good result.

CNN model

Model: "sequential_7"

Layer (type)	Output Shape	Param #
dense_35 (Dense)	(None, 512)	30208
dropout_28 (Dropout)	(None, 512)	0
dense_36 (Dense)	(None, 256)	131328
dropout_29 (Dropout)	(None, 256)	0
dense_37 (Dense)	(None, 128)	32896
dropout_30 (Dropout)	(None, 128)	0
dense_38 (Dense)	(None, 64)	8256
dropout_31 (Dropout)	(None, 64)	0
dense_39 (Dense)	(None, 10)	650

Total params: 203,338
Trainable params: 203,338
Non-trainable params: 0

KNN

- Data: Audio files
- We will use the python speech feature library to extract features
- Extract features from each audio file and save the extracted features in binary form in DAT extension format
- Extracted MFCC feature

Model Comparision

CNN

- Train : 99.89%
- Test : 92.93%

KNN

- Train : 100%
- Test : 67.6%