Final Report

Project Name: Top-n Music Genre Classification Neural Network

Team Members: Cheng-Ying Wu, Sophie Zhao

Course: CS 467 Online Capstone Project, Summer 2022

Date: August 10, 2022

1. Introduction:

The Top-n Music Genre Classification Neural Network Team developed a program that allows users to enter a song clip, and receive a formatted top-n list of genres sorted by confidence value in descending order. The prediction is made by the pre-trained Convolutional Neural Network (CNN) model designed by our project team.

2. User's Perspective:

By using our program, end users can obtain detailed music genre information, which may or may not be different from the original genre that the music labeled. Such information can be refreshing. One type of potential user is a music streaming service that dispatches music lists based on genre preferences.

3. Development Efforts:

Original Project Plan

Task	Time Estimated(hrs)
Week 3 - Review ML Knowledge, Study Relevant Works, & Work on EDA • Review Machine Learning Knowledge • Explore & Study Relevant Works • Data overview • Signal patterns visualization • Basic statistics of the dataset	10
Week 4 - Work on Dataset Preprocessing and Splitting • Learn how to use Librosa for audio preprocessing and finish data preprocessing • Split the dataset into training, validation, and testing subsets	12
Week 5 - Machine Learning Algorithms Modeling & Individual Project Demo Video • Use PyTorch to construct neural networks models like CNN • Create an individual project demo video showing my main contributions to the project • (If time allows) Start to build up the main.py to do training and evaluation	15
Week 6 - Training & Evaluation • Determine the training details like epoch numbers & learning rates, etc. • Determine the performance metrics used • Finish the main program file containing functions such as load_data, train, evaluate, etc.	13
Week 7 - Hyperparameters Tuning & Create Poster • Tune the hyperparameters by doing several experiments • Determine the final trained model used in our application • Work on the poster for our project	15
Week 8 - Team Project Demo Video & Tasks in if Time Allows Section • Create a team project detailed demonstration YouTube video that covers all of the main project features and more • Build our application If Time Allows • Build the application GUI by using Tkinter • Add a content-based recommender system to our application that will recommend to the users other music they might be interested in with their corresponding YouTube links	15
Total Time	**80**

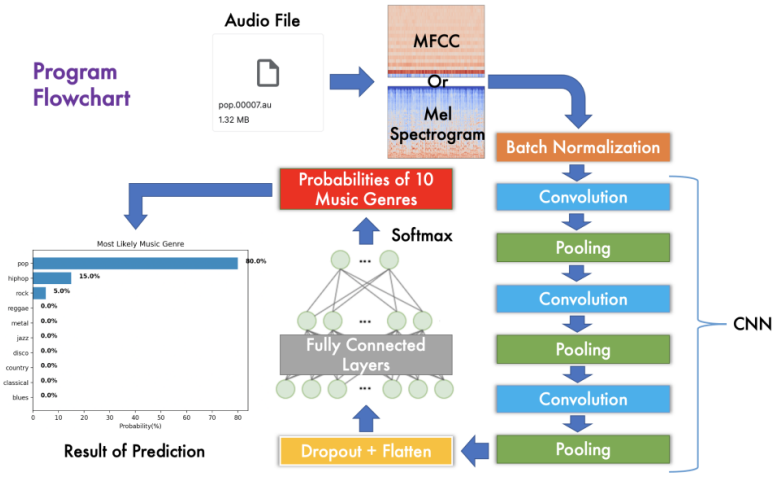
What We Developed

We developed several programs: **data_overview.ipynb** (the Jupyter Notebook with the exploration of GTZAN data), **data_preprocess.py** (used to preprocess the GTZAN raw audio data as the image data of different features), **train.py** (used to train models with different settings), **GUI.py** (the main program to run our application), which meet our plan, and **Tkinter App** (the integrated application that can be run by the users directly).

Deviation Discussion

Luckily, compared to the above original project plan, our development efforts have no major deviations from the project plan. The minor deviation is that due to time insufficiency, we only partially tune our model. That is, we only ran the designed CNN model with different batch sizes & learning rates & features. Also, we did not add a content-based recommender system to our application.

4. Major Technologies:



Major Software Libraries, Languages, APIs, Development Tools

Language: Python 3

Process Music and Prepare Data for Training: Librosa, OpenCV

Build and Train Model: Pytorch, Tensorflow, Keras

GUI: Python Tkinter, Matplotlib, Pyinstaller

Server: MacOS

IDE: Visual Studio Code

5. Accompaniment of Each Team Member:

We exercised two approaches to build and train models in both Tensorflow and Pytorch. Both approaches produced successful results with an accuracy level of 55%-76%. We exchanged image data processing methods and model tuning parameters and techniques.

Sophie: Built the Tensorflow model and used it for the prediction of multiple classifications. Built some part of the UI and connected it to the multiple classification results returned by the model. Packed the project into a standalone executable .exe file (Tkinter App).

Cheng-Ying: Developed data_overview.ipynb, data_preprocess.py, train.py, and GUI.py mentioned above. Mainly use PyTorch to build a framework that can train CNN models (used to predict a top-n list of inputted audio's most likely music genre) by using different settings, such as different batch sizes & learning rates. And created the README.md document on our GitHub repo.

6. Conclusion:

We have completed the following tasks: (a.) Use Librosa to preprocess GTZAN as different features like Mel spectrogram & MFCC. (b.) Use PyTorch to build a framework that can train CNN models with different settings. & Use Tensorflow to experiment with different data preprocessing and model tuning. (c.) Build a GUI that allows users to enter a song clip and receive a top-n list of this audio's most likely music genre sorted by confidence value in descending order. (d.) Create a Tkinter App for the users to run.

In the future, we will first address the bug that cannot load the audio file in mp3 format. Next, we plan to experiment with more different feature engineering & data augmentation to improve the result, employ the confusion matrix to do further analysis for different models, and try with different neural network models, such as LSTM & GRU, etc.

7. References:

Paper & Journal

(1.) G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on speech and audio processing*, vol. 10, no. 5, pp. 293-302, 2002.

(2.) Hareesh Bahuleyan, "Music genre classification using machine learning techniques", 2018.

(3.) Y.-H. Cheng, P.-C. Chang and C.-N. Kuo, "Convolutional Neural Networks Approach for Music Genre Classification", *2020 International Symposium on Computer Consumer and Control (IS3C)*, pp. 399-403, 2020.

Article

(1.) Parul Pandey, "Music Genre Classification with Python", https://towardsdatascience.com/music-genre-classification-with-python-c714d032f0d8

(2.) Sawan Rai, "Music Genres Classification using Deep learning techniques", https://www.analyticsvidhya.com/blog/2021/06/music-genres-classification-using-deep-learning-techniques/

GitHub Repo

(1.) sawan16, "Genre-Classification-using-Deep-learning", https://github.com/sawan16/Genre-Classification-using-Deep-learning

(2.) CNN-MNIST Example Using PyTorch, https://hackmd.io/@lido2370/SJMPbNnKN?type=view

(3.) HareeshBahuleyan, "music-genre-classification", https://github.com/HareeshBahuleyan/music-genre-classification