MUSIC CLASSIFICATION AND TAGGING USING CONVOLUTIONAL NEURAL NETWORKS

Aarthy Ramesh (A59005333)
Department of Electrical and Computer Engineering
University of California San Diego

***Abstract*—In this project, I have implemented an audio-feature based automatic music tagging algorithm using convolutional neural networks. I evaluated different neural net architecture configurations and compared the AUC-ROC scores using the MagnaTagATune dataset, where a 4-layer architecture shows the best performance with mel-spectrogram input.**

I. Introduction

With the increasing availability of digital music, users are spoiled for choice and need a way to find the right tracks for the occasion. In the past, radio stations and jockeys acted as mediators but now in the absence of curators, this problem is exacerbated. While music recommendation systems work well they still struggle with the cold start problem. Music recommendation based on semantic search is another problem that hasn't been effectively solved. Automatic classification and tagging of music is one good way to solve this problem.

Automatic music tagging involves classifying music according to genre, mood, instruments and other such factors. As these tags are high-level features, this can be useful in music search based on semantic keywords. Particularly, I am interested in the case where users search for songs using abstract words like 'beach-vibes' or 'christmas' rather than based on the lyrics or the artist. By first treating automatic music tagging as a multi-class classification problem and then using word-embeddings on the user query to find the closest matching tag, we can return a list of songs that match the abstract user query.

II. Literature Review

Past work in music tagging involved hand-engineering features including MFCCs, MFCC derivatives and other spectral features and assigning tags using KNN or SVM classifiers. But with the success of Deep Neural Network models in speech processing, they are being more widely adopted in music processing too, thus circumventing the problem of feature engineering.

In particular, Convolutional Neural Networks architectures seem very well suited for the task of music tagging. [1]. This makes sense as music tags are high-level features and CNNs are best at hierarchically extracting high level features from given signals. However, this doesn't account for the sequential nature of music signals. So, not surprisingly, transformers and attention mechanisms also show good results. [2]

Additionally, zero-shot learning, where other information such as general word semantic information is also used to ensure the model can handle unseen labels such as tags users may arbitrarily use in music information retrieval was also studied. [3]. For this project, I have focused on the task of auto-tagging using Convolutional Neural Networks. Since, most previous works are benchmarked using the MagnaTagATune dataset, I have decided to use the same here for comparison.

III. Dataset Description

The MagnaTagATune dataset was obtained by humans playing the TagAtune online game. In this game, the two players are either presented with the same or a different audio clip. Subsequently, they are asked to come up with tags for their specific audio clip. Afterward, players view each other's tags and are asked to decide whether they were presented the same audio clip. Tags are only assigned when more than two players agreed. The annotations include tags like 'singer', 'no singer', 'violin', 'drums', 'classical', 'jazz'. The dataset contains 25863 music clips, each annotated with 188 tags. Each clip is a 29-seconds-long excerpt belonging to one of the 5223 songs, 445 albums and 230 artists. [4]

IV. Data Analysis

First, we consider the types and frequencies of different tags present in this dataset. Considering teh tags associated with each clip, we can see from Figure 1 that most clips have fewer tags associated with them and the average tags per clip is 3.45 As seen from Figure 2, the tags are not evenly distributed. Most of the songs seem to be associated with the a few popular tags only. So, to make the analysis easier, we only consider the top 50 tags for the classification task. Further, the 50 most popular tags seem to mostly consist of genres like 'rock', 'pop', instruments like 'guitar, 'flute', vocal elements like 'chorus', 'male vocals' and other generic tags like 'quiet', 'weird'. The tag counts are almost equally distributed between these categories.

V. Feature Extraction

For audio signal processing, mel scale based features are generally preferred. The mel scale (after the word melody) is

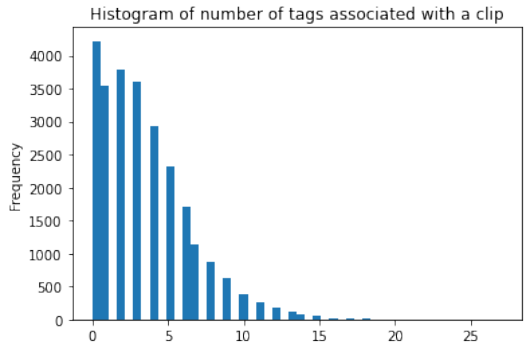


Fig. 1. Histogram of tags associated with a clip.

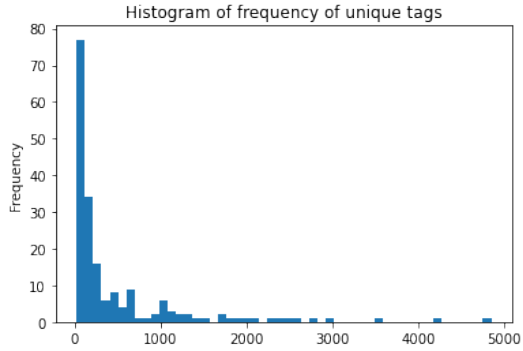


Fig. 2. Histogram of frequency of different tags.

a perceptual scale of pitches judged by listeners to be equal in distance from one another. Since, music classification depends on how humans perceive music, this seems like a good place to start. Previous works have used MFCCs, Mel spectrograms and even raw audio signals as inputs in deep learning models. For CNN architectures, Mel spectrograms have been shown to outperform MFCCs so I have used Mel Spectrograms for this analysis. I computed the mel scaled frequency domain spectrogram using the Librosa library. The differences between different genres or vocal features are visually visible. As seen in figure, flute and guitar based music and male and female vocals show a distinct difference in the spectrograms. Ideally,

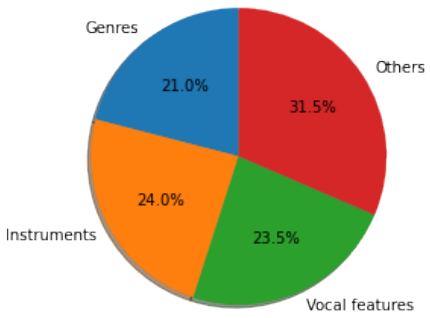


Fig. 3. Pie chart of counts of tags belonging to each category.

this will translate into good classification accuracy using Mel Spectrograms.

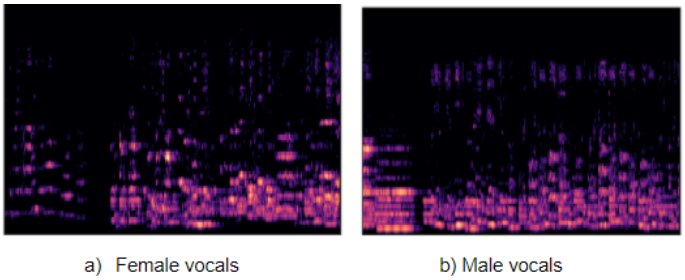


Fig. 4. Mel spectrograms of clips with female and male vocals

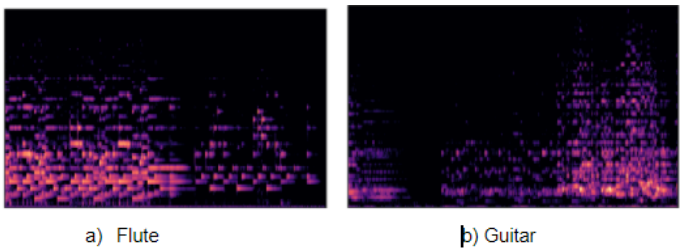


Fig. 5. Mel spectrograms of clips with flute and guitar

VI. Classification

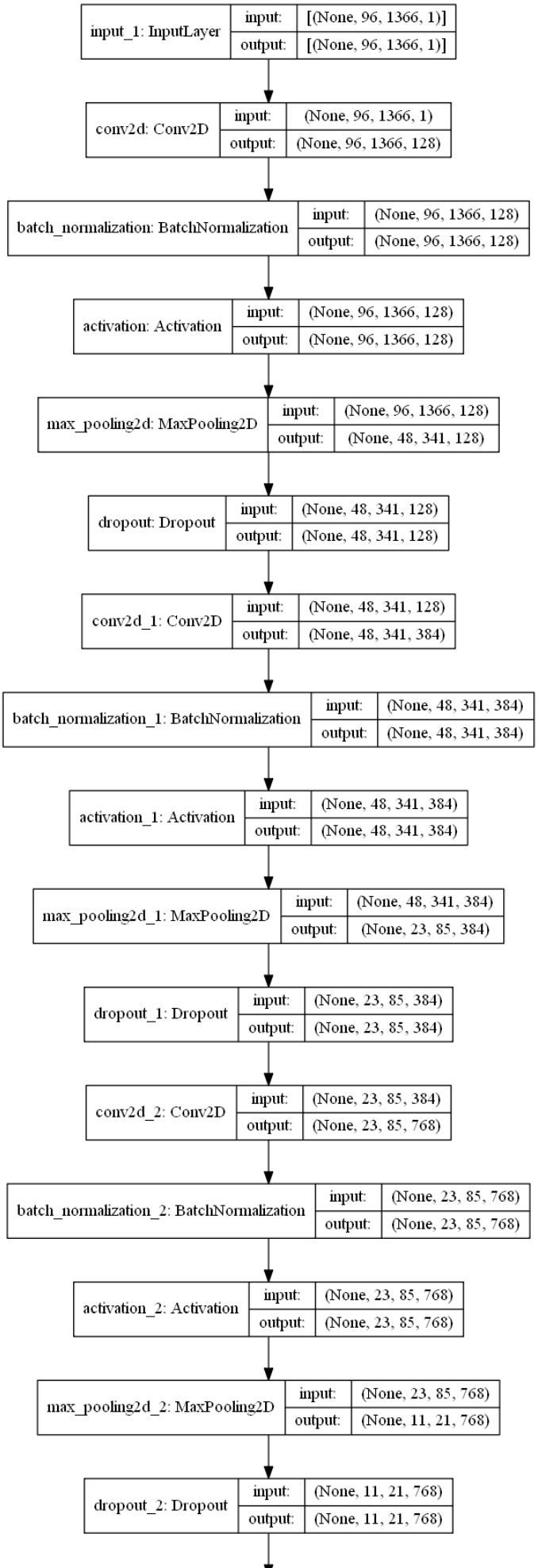
The music auto-tagging problem was treated as a multi-class classification task with 50 possible class labels. A CNN architecture with a final softmax layer for classification was used. The architecture used was the same one suggested in [1] as shown in Figure

A. CNNs for music processing

This model was implemented using Tensorflow and Librosa library was used for preprocessing of the audio signals. The Mel spectrograms were extracted and fed into a data input pipeline using Tensorflow's Dataset API. A sampling frequency of 12kHz and hop size of 256 were used to extract 96 mel spectral coefficients, resulting in a 96x1366 feature for each audio clip. Convolutional Neural Networks with a Relu activation function, batch normalization after every convolution and dropout after every max pooling operation was used. The convolution layers are defined such that the feature size gradually increases in depth. Max pooling layers were used after every convolution layer. Pooling reduces the size of features by subsampling the area with max values. Also, a croos entropy loss function with an adam optimizer was used.

B. Results

Area under ROC curve was used as the accuracy measure. I achieved an accuracy of 70 percentage on the test dataset.



VII. Future Work

I would like to expand this work to include word-embeddings for zero-shot learning. This would allow natural language query based music retrieval. I would also like to experiment with using transformer architectures for zero-shot learning.

References

[1] Keunwoo Choi, Gyorgy Fazekas, Mark Sandler, 'Automatic Tagging Using Deep Convolutional Neural Networks', 2017

[2] Minz Won, Sanghyuk Chun, Xavier Serra, 'Toward Interpretable Music Tagging with Self-Attention', 2019

[3] Jeong Choi, Jongpil Lee, Jiyoung Park, Juhan Nam, 'Zero-shot learning for Audio based msuic classification and Tagging'

[4] Edith Law, Kris West, Michael Mandel, Mert Bay J. ,Stephen Downie, 'Evaluation of algorithms using games: The case of music tagging'