What's Underneath The Cover?

Predicting Music Genres of Albums Based on their Cover Art

Michael Morrison
Department of Computer Science
California State University, Sacramento
El Dorado, CA, USA
morrison.a.michael@gmail.com

Angela Netzke
Department of Computer Science
California State University, Sacramento
Sacramento, CA, USA
angela.netzke@gmail.com

Cyrill Castro
Department of Computer Science
California State University, Sacramento
Sacramento, CA, USA
cyrillhannahc@gmail.com

ABSTRACT

The goal of this project is to explore the usefulness of neural networks in making predictions to help solve real world problems. In this case, the accuracy of neural networks would be assessed within the scope of predicting music genres based on the artwork of their album covers. This problem was solved through the use of neural networks on a manually-created dataset using the Spotify API (known as Spotipy) to create the training dataset. Through this solution, we were able to accomplish a few significant items: a thorough experiment on the success of neural networks, the application of intelligent systems in real world needs, and an opportunity to create our own dataset from scratch with an API from a well-known company.

CCS CONCEPTS

• Computing methodologies • Machine learning • Machine learning approaches • Neural networks • Industry API

KEYWORDS

Music; machine learning; image classification; neural networks; album; cover artwork; genre; Spotipy

1 Introduction

Music genres are categories that are used to classify various pieces of music with each other based on a number of factors, such as musical lyrics, instruments, and bpm. Associated with every music genre is a certain style that is reflected in all elements of a song/album - which can include but are not limited to: the lyrics of a song, the instruments that are used, and the visuals of the album cover artwork. More specifically, album covers of a certain genre tend to display similarities in the design choice of the artwork, reflecting its content and ultimately, its music genre. This information will be considered as image classification, a method used to classify images into their respective categories/classes using trained neural networks combined with the fine tuning of the top layers of the model and VGG16, is explored and imposed on the manually made album covers dataset.

2 Problem Formulation

Just as the cover of a book may be able to hint towards the plot of its story, the cover of a music album may also yield hints about its genre. The goal of this project is to use the data from the Spotify API to identify how well machine learning models can predict the genre of an album based on the design of the album's cover. In addition, we plan on using several different neural network models to compare their image classification performances, and we plan on testing various structures through parameter tuning. To evaluate the accuracy of these models, we plan to compare their classification metrics (i.e. precision, recall, F-measure, classification reports, confusion matrices, ROC curves) to see which out of the three are the most successful at classifying the images.

3 System/Algorithm Design

3.1 System Architecture

Various neural network models were used in a comparison on the classification performance of album covers. More specifically, Convolutional Neural Networks and Transfer Learning were used. With these models, feature normalization and parameter tuning were performed to further test the accuracy of the predictions of the models.

3.2 Model 1

3.2.1 Algorithm Description. Module 1 was a CNN that had a kernel size of (5,5) and a pool size of (2,2). "Relu" activation was used for all Conv2D layers, with compilation optimizer "adam."

3.3 Model 2

3.3.1 Algorithm Description. Module 2 was another CNN model that had a kernel size of (10,10) and a pool size of (2,2). The

"relu" activation was used for all of the Conv2D layers, and "adam" optimizer was used to compile the model.

4 Experimental Evaluation

4.1 Methodology

To execute this project, first we imported the dependencies and the mounting directory, to verify the presence of all of the libraries needed - including SpotifyClientCredentials from Spotipy, as well as the sequence model, appropriate layers from TensorFlow, and the drive from Google Colab. From there, we navigated to the correct directory.

Next, we built our datasets with Spotipy. In both of the datasets, we used compressed 256x256 images of the albums - corresponding to the 'images' column - categorized within their genres. The assembled dataset also included columns such as artists, items, url, etc. In total, we built two datasets: the first consisted of our chosen genres of rock, rap, and classical, while the second dataset consisted of the top three genres that were listed by Spotify. The first dataset was trained on 8,382 images and tested on a 10% subsample, while the second dataset was trained on 8,665 images and also tested on a 10% subsample.

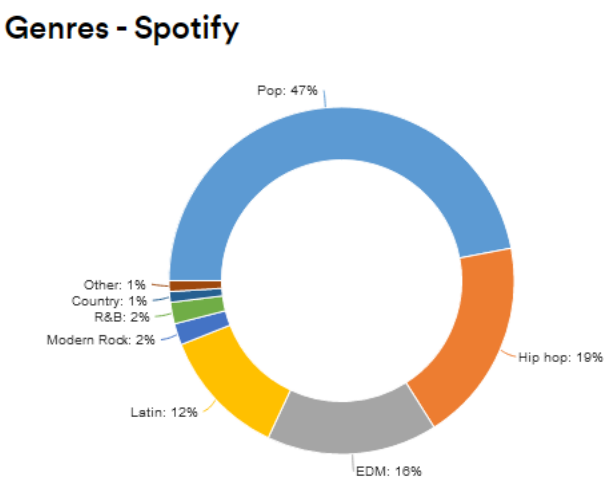

Figure 1: A chart [1] visualization consisting of all of the top genres listed on Spotify. The top three genres were pop, hip hop, and EDM, and this information determined the second dataset that we worked with for this project.

Once the datasets were built, we created our models and tested them. For both of the models, we followed one specific procedure:

1. Create the CNN - create a Sequential model, add the necessary layers. Compile the model.
2. Train the model efficiently, utilizing Early Stopping Monitors
3. Test the model using an RMSE test.
4. Display the image, the predicted genre, and the actual genre to see the accuracy of the current model.

For this project, we created two models, changing the kernel size and the pool size amongst the two,

4.2 Results

4.2.1 Module 1 Results. In the first module, the precision score, classification score, and F1 score were as follows:

```
Precision score: 0.7299417902322022
Recall score: 0.7296137339055794
F1 score: 0.7297437946493672
              precision    recall  f1-score   support

           0       0.71      0.72      0.71       393
           1       0.73      0.73      0.73       381
           2       0.78      0.76      0.77       158

    accuracy                           0.73       932
   macro avg       0.74      0.74      0.74       932
weighted avg       0.73      0.73      0.73       932
```

Figure 2: Results of the RMSE test for the first module on the first dataset. Exhibited a decent overall average accuracy of approximately 0.74.

Additionally, this model was tested on six images to predict their genres and gave the following results:

Index # of Test Dataset	Predicted	Actual
0	rap	rap
20	rock	rock
40	rock	rock
60	classical	classical
80	rap	rap
100	rock	rock

Table 1: A comparison predicted and actual results of the test data for module 1. Out of the six images that were tested, six were correctly classified into their actual genres.

4.2.2 Module 2 Results. In the second module, the precision score, classification score, and F1 score were as follows:

```
Precision score: 0.6038931757679785
Recall score: 0.5979899497487438
F1 score: 0.598731549355323
              precision    recall  f1-score   support

           0       0.58      0.58      0.58       327
           1       0.57      0.63      0.60       306
           2       0.72      0.57      0.63       163

    accuracy                           0.60       796
   macro avg       0.62      0.59      0.60       796
weighted avg       0.60      0.60      0.60       796
```

Figure 3: Results of the RMSE test for the second module on the first dataset. Exhibited an overall average accuracy of 0.60.

Additionally, the test dataset - which consisted of 5 images - had the following results when the module was tested on them:

Index # of Test Dataset	Predicted	Actual
0	rock	rap
20	rap	rock
40	rock	rock
60	classical	classical
80	rap	rap

Table 2: A comparison predicted and actual results of the test data for module 2. Out of the five images that were tested, three were correctly classified into their actual genres.

5 Related Work

One example of related work is a project done by Yanir Seroussi, who performed an album cover classification project that was similar to ours [5]. His goal was to "learn about deep learning by working on an actual problem", and he chose to solve this issue by classifying Bandcamp album covers by genre. His dataset consisted of a balanced dataset of 10,000 images from 10 different genres.

To execute his project, he thoroughly preprocessed his images with the use of downsampling, cropping and mirroring, mean subtraction, and shuffling. He then created an experimental environment and used several datasets to test the environment accuracies, without tuning the learning parameters. From there, he trained his experiments, then ran tests.

There are significant differences in implementation between our two projects, where his project used a Python library called "lasagne", which is also used for machine learning, while we used TensorFlow. Additionally, he used an API from Bandcamp to develop one dataset to solve his problem, while we used the Spotify API to develop two datasets to solve the same problem. Lastly, and most importantly, this person did not tune his parameters, and through this, his results displayed overall lower accuracies. For our project, we did a bit of tuning, but that allowed for better experimentation compared to using datasets. Through this, we were able to train the models that we already created and improve them as needed, instead of testing it on many other datasets.

6 Conclusion

We found our first module to be the better model by about 18% compared to our second model. The first dataset, which were our selected genres of rock, rap, and classical, had the following results:

- Precision score: 0.7321
- Recall score: 0.7243
- F1 score: 0.7243

Meanwhile, our second dataset, which consisted of the most popular genres of pop, hip hop, and EDM, had the following results:

- Precision score: 0.5658
- Recall score: 0.5542
- F1 score: 0.5430

Overall, our models had better results with our chosen dataset compared to the given dataset from Spotify. One possible reason could be due to size; the limitations of our model training environment may not have been able to suffice for the expanded number of genres within the larger dataset. Another reason could be due to the fact that our genres consisted of those that were more separate and established from each other - meaning that they had existed in longer and developed their own standards of what is considered to be of that genre, which could include what their artwork should look like. Meanwhile, the dataset of the most popular genres consisted of genres that were more abstract. For example, the genre of pop has many subtypes to it, such as dance pop, pop rap, pop rock, etc. Additionally, since these genres are fairly new/current, the expectations/standards towards that specific genre may still be developing - in other words, Lastly, being that we are currently in an age of creativity, the artwork of album covers simply could be chosen just due to aesthetic and/or symbolism and have the possibility that there is little to no correlation between the genre and the album cover artwork.

7 Work Division

The entire project was divided amongst the three project members accordingly: Michael Morrison developed CNN model structure, optimized dataset construction, and performed results analysis. Angela Netzke created the dataset assembling structure, and performed model optimization. Cyrill Castro formulated the project proposal, project report, and the corresponding presentation materials.

8 Learning Experience

This project allowed for our group to further experiment with CNNs. Using the knowledge and experience that we have acquired from previous projects, we used this project as a way to solve a real life situation that we were all interested in. To execute, we had previously experienced fairly accurate prediction results with neural networks when using CNNs, the "relu" activation, "adam" optimizer, along with kernel sizes that were greater than or equal to 10 and pool sizes of (2,2).

We used this project as an opportunity to enhance our skills within this specialty of data science. More specifically, we utilized an opportunity to work with an API and build our own dataset. With the advising of our professor, we were able to build a dataset that was used to accomplish what we wanted out of this project, while being reasonable with the columns of data information we were working with. Additionally, we learned how to load and save the datasets (which were stored in all encompassing .npy files), models (which were stored in JSON format), and model weights (which were stored in .h5 files) through a mounted google drive directory to store all of the data. Overall, even though this project was intended for us to practice on skills we have already obtained through this course, we also used this project as a way to obtain new skills and utilize new resources.

ACKNOWLEDGMENTS

Sebastian Norena for his extremely well presented research[4] on the google colaboratory platform rivaling Google's own development documentation. Pulkit Sharma's publicized model structure and documentation[3] for CNN image classification of Film and Television promotional material. Yanir Seroussi for providing an experiment for us to compare our project to in the "Related Work" section. Dr. Haiquan Chen of California State University, Sacramento for his accelerated teaching structure in an area with no shortage of material or industry demand.

REFERENCES

[1] Madeleine Picard (2018). Visualized: Can we Quantify the Most Popular Music?, https://www.displayr.com/most-popular-music/

[2] Jared T.L.C., Prefix, 1387 Different Music Genres Discovered on Spotify, http://www.prefixmag.com/news/1387-different-music-genres-discovered-spotify/203669/

[3] Pulkit Sharma, Analytics Vidhya, Multi-Label Image Classification Model in Python, https://www.analyticsvidhya.com/blog/2019/04/build-first-multi-label-image-classification-model-python/

[4] Sebastian Norena, Good Audience, Train a Keras Neural Network With Image Net Synsets in Google Colaboratory https://blog.goodaudience.com/train-a-keras-neural-network-with-imagenet-synsets-in-google-colaboratory-e68dc4fd759f

[5] Yanir Seroussi (2015). Learning About Deep Learning Through Album Cover Classification, https://yanirseroussi.com/2015/07/06/learning-about-deep-learning-through-album-cover-classification/

MODEL DEMONSTRATION

The Project's Jupyter/Colab Cloud Notebook