Mood Detection: Linking Extracted Mood From Images to Spotify Songs

Yaman Shrestha, Bryan Tran, Zachery Morris
Department of Computer Science, University of Virginia, Charlottesville, VA 22904
[ys2nc, bt2kg, ztm4qv]@virginia.edu

Abstract

Humans are emotional creatures and throughout history, music has helped experience many of these emotions. In addition, when we look at various images within our history, they have contained very strong emotions within them. In order to correlate music and images, we will train a model and integrate it into a web application that determines the mood of a given image. Based on the analyzed mood, the web application will redirect you to Spotify and output a song that best fits the image's mood. In our project, we utilize machine learning to identify the best mood that represents the image. We explore different neural network architectures and how they each perform. We will start with a basic neural network with just linear layers and then moved to more complicated convolutional neural networks (CNN) like VGG-16. We will train our model on the dataset found on Rochester's "Large Scale Dataset for Image Emotion Recognition" [2].

1. Introduction

Describing images not based on features within the image, like objects or corners, is hard for computers to do. Incorporating deep learning techniques like convolutional neural networks allowed us to effectively score images based on their associated mood. We got our dataset from Rochester's image repository, where they included a list of links for each labeled image. We used urllib to concatenate 20,000 images for our training set and 5,000 images for our validation set. The images within these sets consist of a wide range of different scenes for each emotion. Our model predicts four categories: anger, sadness, amusement, and excitement. We believe these four emotions capture the wide spectrum of day to day human emotions. Popular works have predicted on human facial expressions, but the area we focus on is extracting mood from a general image, not just facial expressions. We use popular deep-learning techniques like convolutional neural networks (CNN) and well-known architectures (e.g ResNet50, VGG-16) to maximize our predictions. Then, we deploy our model onto a web application. This application asks for an input image, outputs the predictions, and produces a Spotify playlist which consists of similar songs to the outputted predictions.

2. Related Work

There have been many attempts to build models for emotion detection. One of the big areas for emotion detection has been in facial expressions. A group, at Stanford University, attempted to classify facial emotions into seven categories [1]. They got their data from a Kaggle competition, and were able to get comparable results to the top ranked participants. They used popular deep-learning architectures ResNet50 and VGG-16, two architectures that we will also utilize in our project [1].

3. Model

We propose several models for our project. These models include: a simple neural network with seven linear layers, LeNet, ResNet16, ResNet50, and VGG-16. We explore how our results change as we progressively use more complex models. We ended up choosing ResNet50 and VGG-16 because they are two popular deep-learning architectures that perform well in facial expression recognition [1]. Although our networks feel like a blackbox, we hope our models well be able to extract a combination of low level and high level features that will help with producing an effective prediction. Low level features, like color and pixel intensity, and high level features, like edges and combination of edges, are characteristics of images that we hope will make it possible to predict emotions from a scene.

4. Experiments and Results

We use pytorch to create our models and run it against our training and validation sets. Our worst model was our neutral network with no convolutional layers, just seven linear layers. This seven linear layer model had an accuracy of 0.2590, which is as good as random guessing. For our LeNet model, we transformed the image to a 32 pixel wide and 32 pixel tall, three channel, normalized, tensor. Then, this tensor was used to train our LeNet model. After training

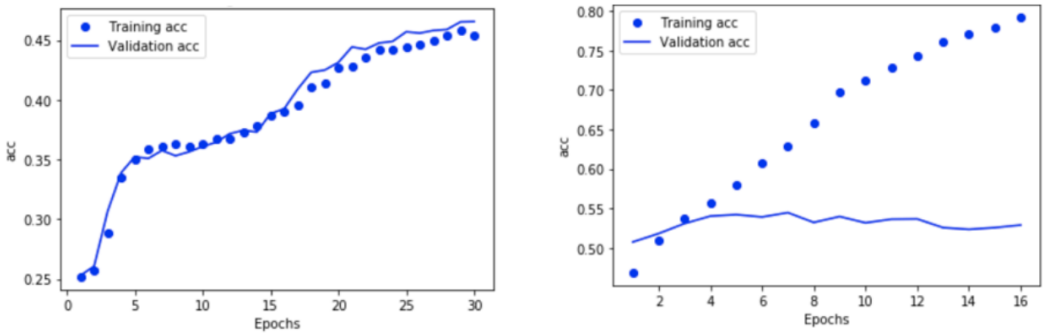

Figure 1. Here we show a figure of the accuracy of both ResNet50(left) and VGG-16 (right) from the first epoch.

all the weights for LeNet from scratch, we achieved 0.3597 accuracy which is slightly better than random guessing. To improve on these results, we employed ResNet50, VGG-16, and transfer learning. ResNet50 had a final accuracy of 0.4655 where as VGG-16 has the best result with a final accuracy of 0.5224. VGG-16 produced an accuracy much better than random guessing.

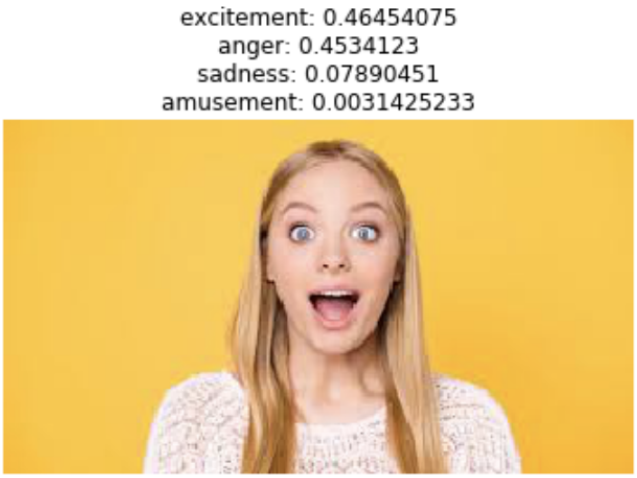

Figure 2. This figure shows an example of an input image and the predictions.

5. Deployment

In order to deploy the trained model and utilize it to obtain songs of a particular mood, we created a web application using Flask, which is a python based web framework. Flask made it easy to use python libraries such as pytorch, which we used to train and load out model into our web application. In order to correlate the results of our model to songs, Spotipy, a lightweight Python library, was used to pull data from the Spotify Web API. The web application first asks the user for their Spotify username. Once their username is inputted, they are navigated to Spotify's authentication, in which, they can allow the application to access their user's library, top and followed artists, and create or modify playlists. After authentication, the user is taken to our application's upload screen, in which, they can upload an image. After the image is uploaded, and the user clicks predict, the application outputs the four predicted moods from the model. Our application takes the highest probable mood and passes that to our python microservice in the Flask application. Our microservice utilizes the mood and calculates the mood score, ranging from 0 to 1. This correlates with Spotify's API variable called valance, which also has a 0 to 1 range, where 0.0 is a very negative mood state and 1.0 is a very positive mood state. Using this valance score and the user's music listening history from Spotify, the algorithm finds songs that had a similar valance score to create the playlist. Once it generates this playlist, the application attempts to open the Spotify app to display the playlist to the user. The Github page with all the installation instructions (on the readme.md) can be found here: `https://github.com/shresthayaman/CVMoodProject`

References

[1] J. W. Alexandru Savoiu. Recognizing facial expressions using deep learning. *CS231n: Convolutional Neural Networks for Visual Recognition*, 2017.

[2] H. J. Quanzeng You, Jiebo Luo and J. Yang. Building a large scale dataset for image emotion recognition: The fine print and the benchmark. In *Thirtieth AAAI Conference on Artificial Intelligence (AAAI)*, 2016.