Machine Learning Engineer Nanodegree

Capstone Proposal

Mofei Liu
March 16th, 2019

Proposal

Domain Background

In recent years, sneaker culture and the industry itself has been grown a lot. Last year, the U.S athletic footwear industry generated $17.5 billion, according to the NDP Group, and its resale industry is estimated to be worth $1 billion[1]. Start up for sneaker trading like GOAT[2] & StockX[3] surfaced and have gain a lot of attention from the community and also the outsiders. With more and more people start get into it collecting sneakers and using them as an outlet to express themselves, brands like Nike & Adidas start going at a very fast paced when it comes to sneaker release. Dozens of new colorway & new silhouettes being released weekly and hundreds new products every year, in 2018 along, Jordan brand has released 60 difference colorways of the classic Air Jordan 1[4]. Also, a lot of fashion houses like Balenciaga see the opportunities and joined the game, releasing their own sneaker focused on a different consumer base. As a sneakerhead I've enjoyed collecting, buying & selling and tell the stories, however it's very excited to see that I can use what I learned to create something helping more people learn the knowledge about it.

Problem Statement

As mentioned above, nowadays with the athletic brands, fashion designers and also new brand, most consumers are very overwhelmed by the new product they're seeing every day every week, it's really hard to "keep up". Most topic or question I saw on Instagram or a forum will be "what shoe are these?" "can anyone id these shoes for me?". Everyone can know what Air Jordan 11 looks like but maybe not 16 or 17, not to mention different colorways itself. So here I'm trying to solve a program for these people to tell & identify the name & colorway of the sneaker by the picture.

Datasets and Inputs

For this problem, the input will be a picture that user upload or taken from the mobile device, hence the idea of this program is to identify the object- Sneaker. The dataset will be lot of pictures of sneakers with label corresponding model & colors. These datasets will be used to train a CNN for image identification, to achieve this I will be scraping images to compose this dataset from GOAT & StockX website since it's already been standardized.

Solution Statement

To solve the problem, I'll be using the datasets collected to train a CNN model for image classification. The reason I'm using this deep learning model because it's very effective finding patters in within images using filter. The model will take a picture as input and giving out the prediction of the shoe model & colors. I would also consider using transfer learning of bottleneck feature from famous RESNET50 & VGG16. Depending on the sample size per label, my priority will be the model first then colors.

Benchmark Model

1. Random Choice: A dummy classifier that random choice one of the labels

2. Google Image: Searching the image on Google and return first result as labels.

Evaluation Metrics

To evaluation the model vs benchmark model, I'll be using the logarithmic loss method[5], which works by penalizing the false classification. It should work well here since this is a multi-class classification problem.

Suppose, there are N samples belonging to M classes, then the Log Loss is calculated as below:

$$LogarithmicLoss = \frac{-1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} * \log(p_{ij})$$

Where,

y_ij, indicates where sample i belongs to class j or not

p_ij, indicates the probability of sample i belonging to class j

Log Loss has no upper bound and it exists on the range [0, ∞). Log Loss nearer to 0 indicates higher accuracy, whereas if the Log Loss is away from 0 then it indicates lower accuracy.

Project Design

1. The first steps of my project will be data collecting. As mentioned above, I'll be using some python library to parsing/scraping images from websites and save into a different folder by model/colorway. Here I plan to use some popular library like scrapy/beautifulsoup/request. Eventually I hope I can get over 30000+ image from both sites, if not sufficiently enough add more from google search & other source.

2. Next, I'll do some analysis to see if there's any imbalanced number of images per category and try to add/remove to balance the image count for each. Compiling a file for every label of the images for training input.

3. Start training a small CNN from scratch using all existing datasets, comparing performance with transfer learning.

4. Using feature extract from pretrained network to run a fully connected network with 2 output on the last layer to get the predictions. Comparing performance with step #3.

5. Fine tuning hyperparameters with different optimizers, improve model performance.

6. Evaluation the result.

Reference

1. http://www.prweb.com/releases/2017/01/prweb14023771.htm

2. https://www.goat.com/

3. https://www.stockx.com/

4. https://theundefeated.com/features/60-signature-air-jordan-1s-were-released-in-2018-these-are-the-15-most-fly/

5. https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234