Cole Hoffer, Bibek Pandit
6.s198 Data Review Deliverable

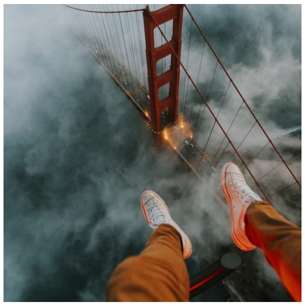
What the data looks like

Each image is 650x650 pixels and about 50kb on average. We have scraped 750 images from 62 individual instagram photographer accounts. In my research, the general number of photos on a popular and professional account was around a thousand, so we chose to use the 750 most recent posts per photographer in order to have a standard number of posts. There were also cases where a particular account was a “personal” one in the beginning, and then transitioned into one of professional photography, so choosing the most recent 750 allowed us to make sure we were only getting the professional quality images. In terms of labels, there is only one label/category for each image, which is the photographers instagram username. That username is stored in the filename (“colehoffer-234.png”) as well as it’s sub-directory.

The current plan is to have a safe goal of identifying 10 extremely distinct photographers, which would have a total storage size of 500-550mb. The goal beyond that would be to be able to identify from a set of 25 photographers, which looks to be at about 1.4GB for the 25 photographers we’ve initially chosen from the 62 we have images for.

cestmaria-61.png

paulnicklen-428.png


shortstache-339.png

airpixels-5.png

Where the data came from

The data was scraped directly from Instagram’s desktop website. I used Selenium, a web browser automation library to load all of the photographers posts onto the page. I then extracted the html and parsed for the 750 thumbnail image sources using Selenium's web parsing tools. Once I had the urls, I ran “urllib.request.urlretrieve(...)” to save the download the images locally.

Is the data good enough

I think 750 images (+ augmentation in preprocessing) will be enough data for our project. Similar papers for painter identification were able to identification with 500 paintings and 100 paintings per artist, so us having the 750 unaugmented images should hopefully be enough. We have also done manual quick checks to of the images to make sure they were all professional quality.

Licensing

The images are publicly available to viewing and downloading, we just could not use them in any commercial application since they are technically owned by Instagram.

Data Storage

The nice thing about the pulling the images from the thumbnails was that they were individually small in size (around 50kb each on average). Thus, even looking at 25 separate photographers only requires about 1.4GB total. So we can reasonably store the image data on our local machines. Then we will move them to wherever we run our training/validation of the model (probably just on the cloud machines provided to the class)

Base Neural Network

We intend to use pre-trained CNN models in our final project, specifically, GoogleNet and ResNet models pre-trained on the ImageNet dataset. We plan to start off with the pre-trained GoogleNet architecture, record the performance, and do the same on pre-trained ResNet-50, ResNet-101 and ResNet-152. After that, we choose the best architecture within them, and work to improve the performance on that architecture. We might also implement ensemble methods to further improve performance if time permits.

Why is it a good fit?

A similar project to ours had been done at Stanford for artist identification (not photographer identification). The paper describing the result mentions how pre-trained ResNets had the highest accuracy (that too by a huge margin) among all the models used. This obviously makes pre-trained ResNets very attractive to work with. In addition, ResNets have been able to produce great results on the ImageNet dataset. All this signal the immense power these CNN architectures have.

Why additional data?

Finally, as known to everyone in the Machine Learning community, the more data, the better. More data is specifically nicer to have for deep neural network architectures like ResNets because these architectures have a great number of parameters, and training them with insufficient data can cause overfitting, hence more data is desired. Additional data also means bigger validation and test set, hence better metrics of measurement.

Pre-Processing

Right now, most of the images we have are of the size 650 ✖ 650 ✖ 3, but most standard CNN architectures accept images of size 224 ✖ 224 ✖ 3. We would either crop the images or resize them or perform a combination of the two to achieve the required size. We actually have the code to resize and crop, so that should be relatively straightforward. Next we plan to augment the images. The two reasons for this are: 1) Increase the overall dataset, 2) Capture features and styles within the images better by providing the same image, but rotated or translated or blurred. Finally, we need to convert the images into a numpy array to feed into the neural nets. We can do this beforehand and store the numpy array in a h5 file (also have code for that). Or, we could just pass the path to the dataset to the ImageDataGenerator in Keras/TF which convert the images into numpy arrays before feeding them into the neural nets.