PicMe – Predicting Ideal Instagram Post

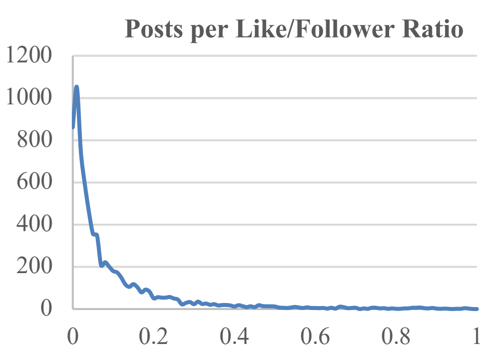
Raul Dagir, Philip Pfeffer, Victor Yin

Overview

- Project goal: to redefine the way people view and share their memories by optimizing which photos taken should be curated into an album (based on the artistic and sentimental qualities of an image).
- Model goal: take a whole album of photos from a user, determine what the best pictures are, and return a curated set of highlights. We collect Instagram data to inform our model and use the ratio of likes to number of followers as a quantitative measure of how good an image is.

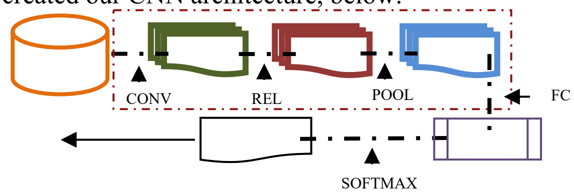
Initial Model & Data Acquisition:

- Starting point: binary classifier over data fetched with our Instagram scraper.
- Extracting basic features: post is an ad, number of likes, hashtags, as well as more implicit features that we thought would have an effect on the ratio of likes and over which the user would have control. This includes hour of day, tags and caption.
- Complex features from the image: colors, facial/object recognition using image processing libraries scikit-image and OpenCV.
- Our classifier predicted if an image should get more or less than 0.35 likes to followers. The accuracy was 82%, though at this time the dataset was small, so the reliability of results was called into question.



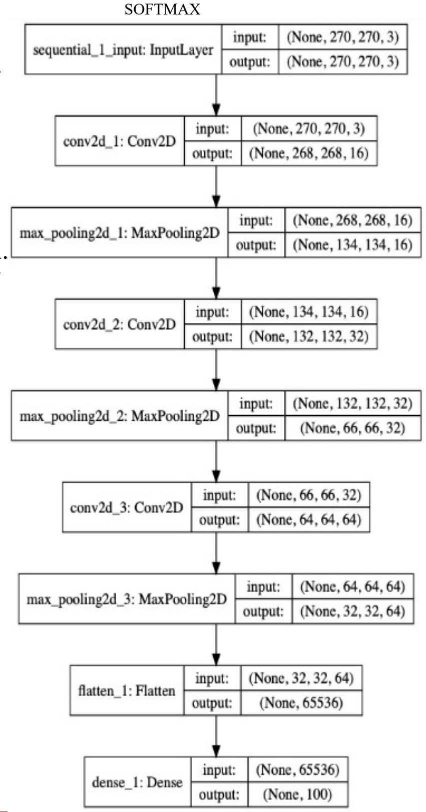
Advanced Model

We decided to make our model less reliant on Instagram-related features and more focused on image specific predictions. For this, we collected a significantly larger dataset limited to square images. We standardized and resized the images to 270 x 270px to retain image features while optimizing the photo for time-efficient learning. Then we created our CNN architecture, below:



The input layer consists of the **270x270 RGB** images. Next, we convolve each image with a **3x3 filter matrix or 'kernel', using ReLU activation**. In between each pair of convolutions layers, there is a **pooling layer** (which performs down sampling on the preceding non-linear layers). Finally, the output of the last pooling layer is **flattened** and fed to the **dense** layer.

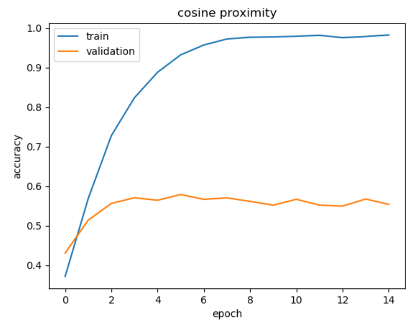
We ran our model for 15 Epochs with early stopping in order to optimize for both low training loss and high validation accuracy.

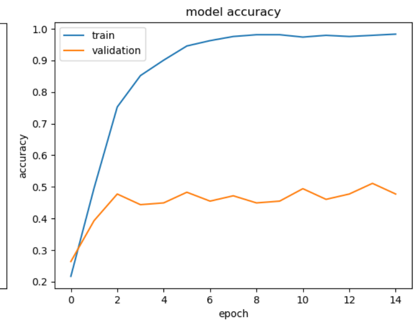


References

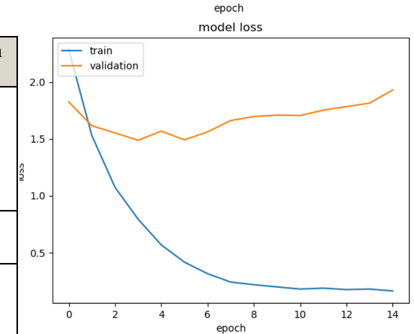
1. Chen, Luyang, et al. "Image-Based Product Recommendation System with Convolutional Neural Networks."
2. Sorokina, Ksenia. "Image Classification with Convolutional Neural Networks." Medium, Medium, 26 Feb. 2019, https://medium.com/@ksusorokina/image-classification-with-convolutional-neural-networks-496815db12a8.

Result and Discussion





	Training set	Test/validation set
Categorical Cross Entropy Loss (Early Stopping)	0.4169	1.4909
Max accuracy	0.9794	0.5112
Cosine Proximity (Early Stopping)	0.9322	0.5789



Findings:

- Cosine proximity turns out to be a better approximation of our loss, given it punishes small variations in the output vector lightly compared to bigger changes. Notice the strong correlation between the model accuracy and the cosine proximity plots.
- Validation accuracy curve did not reflect the increases in training accuracy. The maximum validation accuracy we could get, **51.12%,** is still weak and possibly indicates signs of overfitting.
- We believe we might not be overfitting given how the loss of the validation/test set keeps growing (and the accuracy keeps fluctuating around the same values) even before the training loss reached an asymptotic behavior after the **eighth epoch**. A sharp decrease in validation accuracy was expected once the model overfit.
- Believing that our problem was how much data we had, we scraped a **7500 samples** dataset, and the results were barely different. We are now scraping an even larger dataset.
- We will analyze **saliency maps** of each picture to understand what information the CNN is extract from our dataset, and if it's actually possible to predict the likes to follower ratio using only the visual elements.
- We are now finalizing a new model that combines the post metadata with the CNN outputs to create a more robust prediction.