Exploration of Automatic Label Generation from Social Media through Classification of Instagram Selfies

Luke Merrick*
University of Virginia
Charlottesville, VA 22903
contact@lukemerrick.com

Colin Cassady
University of Virginia
Charlottesville, VA 22903
cvc2cq@virginia.edu

Abhimanyu Banerjee
University of Virginia
Charlottesville, VA 22903
ab3cb@virginia.edu

Abstract

A ubiquitous issue in the realm of computer vision is the acquisition of labeled data. In this work, we explore the potential for extracting image annotations from raw social media data. We investigate a data set consisting of images with the "#selfie" hashtag from the popular image hosting service Instagram and the publicly available user information, likes, and comments associated with those images. We propose a new method for extracting image "likabaility" labels for binary classification using a predictive linear model that takes into account the relationship between likes garnered, user followers, and number of hashtags used in an image uploaded to Instagram. Finally, we train a modified version of the famous VGG-16 model to classify this data set, achieving an accuracy of 72%. Our results prove that community response is affected by image content, and that the visual features of likable content can be learned.

1 Introduction

The selfie is a style of photography in which a person takes an photograph of him or herself, and it is an extremely prevalent style of photograph on social media today. Due to its somewhat vain undertones and immense popularity, many users of social media websites such as Facebook and Google+ seek to frame the perfect selfie. Doing so rewards the user with followers and likes, which can translate to an ego boost and even sponsorship from fashionable brands.

The website Instagram is one such social media platform, where the focus is on user-generated image content. One of Instagram's key features is the "hashtag" functionality, where users can tag their images using an octothorpe (#) and a tag name (e.g. #SpringBreak). Users can search for images by hashtag, thus the feature helps to organize the sheer number of photos Instagram's 300 million daily users upload onto the site [1]. Users can opt to "follow" other users, which indicates that they'd like to be notified when the user they're following uploads an image. Additionally, for public images, any user of Instagram can "like" that image, indicating their approval.

We believe that success in understanding the features present in well-liked selfies provides valuable insight into the way that humans interact with image-based social media, and that an understanding of these features can provide benefit to social scientists in their attempt to understand why users of social media like certain images more than others. In this work, we undertake not only the task of training a convolutional neural network to recognizes likable selfies, but also attempt to provide insight into the visual process that stimulates affects the human response to image content on social media.

*author bio and other information at lukemerrick.com

2 Related Work

In recent years, there have been many attempts to leverage the tremendous amount of information available on social media to gain insights into decidedly human tasks like facial recognition and visual sentiment analysis.

Nguyen et al. presented their analysis of micro-videos – five to ten minute long videos which are prevalent on social media. These videos utilize camera viewpoints uncommon to traditional film making, which the authors aptly name "egocentric" and "self-facing." For reasons similar to our own approach, these videos were attractive to the researchers due to their already being tagged by users of social media. Campos et al. applied the established VGG-19 model to the task of predicting both these tags and the novel camera angles used, and were able to achieve accuracy far greater than random guessing for both.

Most closely related to our approach, Karpathy, now a research scientist for OpenAI who at the time of his selfie research was pursuing his PhD in machine learning at Stanford, also attempted to predict the quality of Instagram selfies. His approach, which heavily influenced ours, scraped 5 million month-old or older images from Instagram, generously removed outliers, sorted the photos based on number of followers and classified them in groups of 100 photos split into top and bottom halves by "likes." He then trained a convolutional neural network on this labeled data set. In our approach, we do not remove outliers and we do not bin our images, but we utilize a novel linear regression controlling for followers and number of hashtags present to solve the same problem. Karpathy's work has garnered a lot of attention and respect, and has influenced at least one paper [5].

3 Dataset

Figure 1: A random selection of 12 images from our data set. These are images from the website Instagram tagged with the hashtag #selfie, posted between November 8th and November 15th.

We have collected 486,149 images labeled with the hashtag *#selfie* scraped from the website Instagram. These images have metadata associated with them including number of likes, number of comments, date posted, caption, owner, etc. We utilized a technique of collecting only freshly-posted images and revisiting the metadata for collection later to avoid issues with changes in macroscopic trends in Instagram user behavior and issues resulting from some photos having had longer to accrue likes and comments than others, regardless of their visual content.

After fresh images were collected over the course of a week, over the period of the following week we revisited each image exactly 7 days after collection to update its metadata, meaning that each image was given almost exactly the same amount of time to accrue likes and comments. Interestingly, 56,966 photos, or about 11.7% of our data set had been deleted by the time of updating. We speculate that this deletion was due to users deciding that their selfie wasn't performing well with respect to likes, although it would be odd if one in nine selfies were deleted for this reason. Other possibilities include full account deletion, network errors when attempting to access the images, or perhaps removal of the photo by Instagram for breaking their terms of use policy.

During this update process, other than the metadata associated with the image, other information directly relevant to the image were also updated, including user information and comments. Of the

429,183 remaining images, we have data on 311,475 unique users, each having posted at least one image in our dataset. Information about users include their name, number of followers, number of people they are following and personal description. A massive 1,500,967 comments have been made on images in our database, for an average of 3.5 comments per image. Comment data includes the comment itself, time of posting, and name of the posting user. At the time of writing, our approach to classification has not yet incorporated data from comments or image descriptions.

After a visual analysis of the data, we realized that a fair number of images weren't selfies despite being tagged as such. To rectify this, we utilized the facial recognition functionality of the open source tool dlib [6]. Dlib recognized faces in 232,971 images. We opted to only continue using images that contain exactly one face, which totaled 200,545 images.

3.1 Label Creation

Intuitively, we want our labels to capture the likability of a selfie as a score, but extracting this information from the metadata we collected poses a serious challenge. The number of likes that a selfie garners is influenced not just by the content of the image, but the number of followers the user has, the number of hashtags associated with the image, and other unaccountable factors such as the selfie being posted on external websites. Our goal is to isolate the influence the image content has on the number of likes, controlling for all other factors.

Our approach utilizes a linear model of the following form

$$\log_{10}(1 + Likes) \sim \log_{10}(1 + Followers) + NumberOfHashtags \tag{1}$$

Figure 2 plots this relationship, as exhibited by the metadata of the collected images. We think it is logical to assume that the noise in this data can be explained by the content of the image itself. Therefore, we subtract the number of likes this linear model predicts an image would receive from the number of likes recorded in the metadata. This difference represents our score metric. Images that outperform our prediction receive a high score and images that underperform our prediction receive a low score.

After initial testing and unsuccessful training, we determined that it would be best to subset our data even further, by isolating images with scores in the top and bottom 10% of the overall distribution. These images were given the class names "Top" and "Bottom" respectively, and rather than attempting to predict the image's score, we instead predict on this binary classification. This puts our dataset size at 40,109.

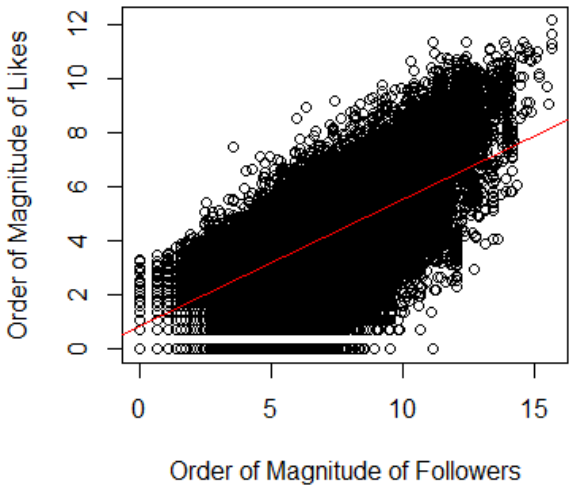

Figure 2: Scatter plot visualizing the relationship between Log(Followers) and Log(Likes). We propose that the distance between the predicted value and actual value is explained by the features in the image itself

4 Predicting Likability of Instagram Selfies

4.1 Network Architecture

Due to the limitations of our dataset's size and chaotic nature of our labels, we elected to build all of our models on the existing VGG-16 ConvNet architecture presented in [7]. This allowed us to utilize the model's pretrained weights, which were trained on the ImageNet dataset [8]. We evaluated two different architectures in which to apply the pretrained VGG-16 model to our binary classification task, utilizing several different SGD optimization strategies. The first model uses only the convolutional portion of the VGG-16 network fed into a single randomly-initialized logistic classification layer, while the second model uses all of the VGG-16 layers except for the final softmax classification layer, which we replaced with a randomly-initialized logistic classification layer.

For all random initialization of weights, we used the default configuration of the Keras deep learning library [9] which samples from a uniform distribution normalized to layer size under the scheme proposed by Glorot and Bengio in [10].

5 Experiments

5.1 Experiment Setup

Training and Test Sets As described in 3, we collected 490K images from Instragram with #selfie. We then performed facial recognition using dlib [6] to filter out images without exactly one person in them, resulting in a dataset of 200K images. Finally to make the difference between the two classes of images marked, we chose only those images whose score labels are in the top or bottom decile. A quarter of these images were held out of training to serve as a validation set.

Training Pipeline For the training of the proposed convolutional networks described in 4.1, we used all 30k images in the training set with random horizontal flips. We were sure to resize all images to 224x224 pixels and normalize their pixel weights to the original ImageNet mean, which was used by [7] to train the VGG-16 network .

All training was performed via mini-batch stochastic gradient descent with batch size of 32 and a global learning rate of 0.001. We experimented with several training approaches including the freezing of pretrained convolutional layers and the addition of l2 regularization, dropout, and momentum as configured in the original pretraining of the underpinning VGG-16 network 5.2.

Our model was trained using the Keras deep learning library [9] on a AWS P2 instance equipped with an Nvidia K40 GPU. We trained in experiments consisting of 5-20 epochs of the training set, with the maximum validation accuracy usually observed well before training was terminated (termination was triggered by various conservative saturation rules we tested).

5.2 Configuration of Loss and Gradient Descent

For our initial benchmarking of our selected architectures, we performed three tests:

(A) SGD optimization of the frozen pretrained VGG-16 convolutional layers + unfrozen logistic classification layer (i.e. logistic classification of deep features)

(B) End-to-end SGD optimization of pretrained VGG-16 convolutional layers + logistic classification layer

(C) End-to-end SGD optimization of full pretrained VGG-16 model with final layer replaced by logistic classification layer

As can be seen in Figure 3, both of the end-to-end configurations suffered from a precipitous drop in validation accuracy after rapidly peaking. This drop was subsequently curtailed by the accuracy saturation termination rule we employed to eliminate needless computation. The results in Figure 4, which are consistent with the trend in training accuracy of configuration B as well, suggest that overfitting was the cause of this decrease in accuracy.

To combat this issue, we created a new configuration:

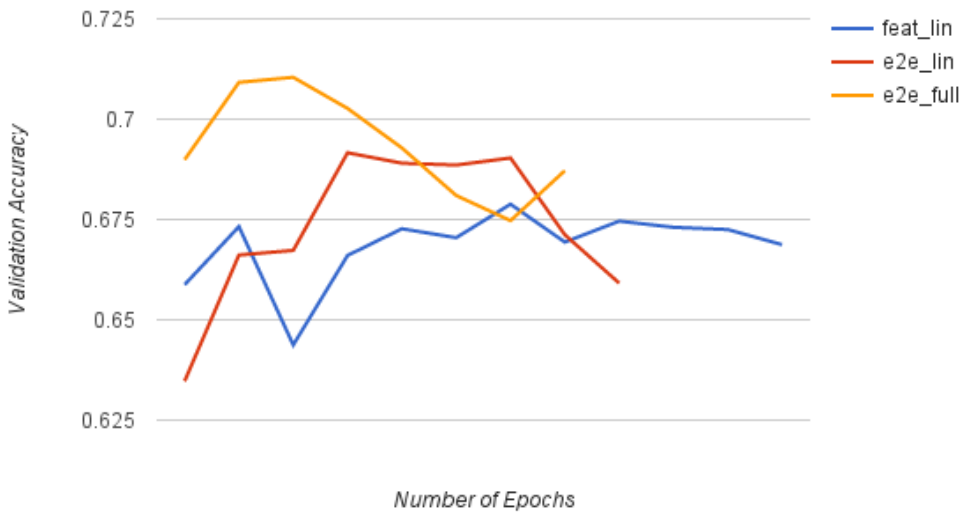

Figure 3: Validation accuracy over training time benchmark evaluations

(D) Architecture identical to configuration C with end-to-end SGD in the configuration used to train VGG-16 [7] for the ImageNet competition [8] (momentum = 0.9, dropout = 0.5 and l2-regularization = 0.0005 on the first two fully-connected layers)

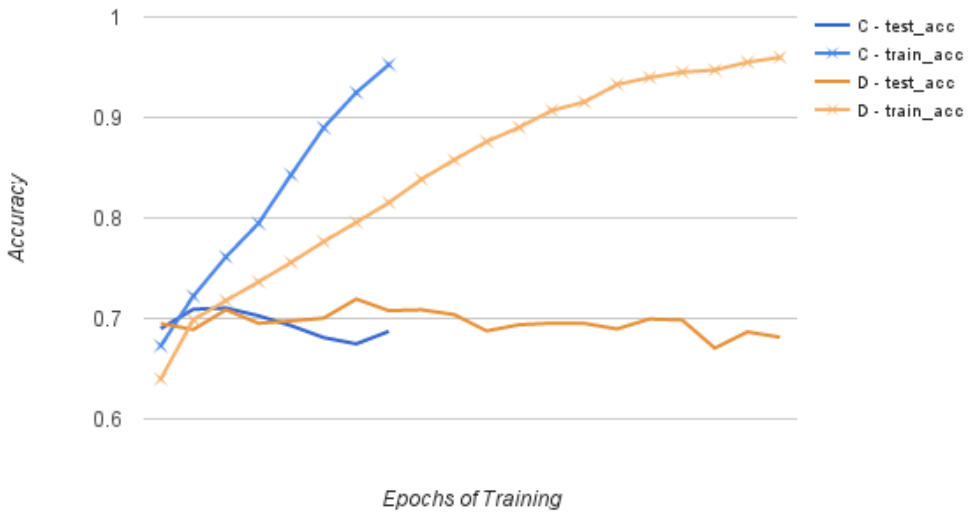

Figure 4: Test accuracy vs. validation accuracy for configurations with and without dropout and l2 regularization

As we can see from Figure 4, these regularization steps eliminated the drastic drop in validation accuracy witnessed in the training of configuration C while maintaining a high overall validation accuracy. We also see that it took more than twice as many epochs for the training accuracy to

reach 95%, which supports our belief that overfitting is the cause of the drastic dropoff in validation accuracy witnessed in training configurations B and C.

5.3 Results and Analysis

One of the goals of this work was to be able to automatically generate labels from social media metadata. Our assumption while creating our score was that once the influence of metadata metrics were accounted for, the likability of an image can be learned from visual cues in the image itself. While our training suffered from overfitting and only achieved a maximum of 72% validation accuracy, the overall performance is comparable to human performance, as can be seen from Table 1 and Table 2. Table 2 was populated via the following experimental procedure: participants were allowed to freely examine a large sample of labeled images and then were asked to classify a different set of unlabeled images.

Table 1: Benchmark Training Results

Configuration	Max Validation Acc.	Epochs Trained	Final Validation Acc.
A	0.678814	12	0.66873
B	0.691627	9	0.659145
C	0.710430	8	0.687167
D	**0.719316**	20	0.681343

Table 2: Human attempts at selfie categorization

Test Number	Participant	Number of Images Classified	Accuracy
1	Person A	50	0.61
2	Person B	50	0.76

Due to the saturation rules used, not all models were run for the same number of epochs. Nonetheless, configurations C and D achieve a peak validation accuracy of approximately 72% before slowly sliding downward and ending at 68% at termination. This suggests that training the VGG model end-to-end allows the model to learn a slightly richer representation of the underlying visual cues that influence likability.

(a) Images our model classified as "Top" with borders showing ground truth (blue is "Top," red is "Bottom")

(b) Images our model classified as "Bottom" with borders showing ground truth (blue is "Top," red is "Bottom")

Figure 5: From a random subset of 1000 images from the original dataset, our model was most confident that these were of the class (a) "Top" (b) "Bottom"

Finally, we chose to test configuration C, which had the best final accuracy, on a randomly sampled subset of 1000 images from our dataset. Figure 5 shows the best predictions from the model belonging to the two classes - "Top" and "Bottom" as well as ground truth, or more accurately, actual class of each image. While there are some misclassifications in each set of predictions, it is easy to see some

of the visual cues that the model is learning. For instance, soft lighting seems to improve likability compared to photos displaying washed out or garish colors.

6 Conclusion and Discussion

We have presented a method to automatically generate labels from the metadata of social media images. We trained a convolutional neural network to learn features from the images that are representative of the likability of the image. We conclude by discussing ways of overcoming the wide variation in our score variable.

One way of improving the generalizability of our model would be to use more data. We primarily focused on the top and bottom decile for our experiment, but it is also quite likely valid to include the top and bottom two or three deciles. Doing so would certainly provide more data, although we are not sure exactly how it would affect the accuracy of the labels.

Another limitation of our approach lies in the scope of the factors used to calculate the score for each image. We wish to simply determine how many people liked an image after seeing it, although view count is not a metric Instagram's website provides. Additionally, future work could take into account other information like the comments Instagram users make on an image. Finally, it would be interesting to see if this work can be extended to incorporate more than two class labels or perhaps a continuous score with less unexplained variation than the score we calculated in this paper.

Acknowledgments

We would like to thank Professor Vicente Ordóñez-Roman for his excellent instruction and mentorship throughout the semester. It has been a great pleasure to take part in the inaugural section of your *Computational Visual Recognition* course.

References

[1] Instagram. `https://www.instagram.com/press/`, 2016.

[2] Phuc Xuan Nguyen, Grégory Rogez, Charless C. Fowlkes, and Deva Ramanan. The open world of micro-videos. *CoRR*, abs/1603.09439, 2016. URL `http://arxiv.org/abs/1603.09439`.

[3] Victor Campos, Brendan Jou, and Xavier Giro-i Nieto. From pixels to sentiment: Fine-tuning cnns for visual sentiment prediction. *arXiv preprint arXiv:1604.03489*, 2016.

[4] Andrej Karpathy. What a deep neural network thinks about your #selfie. 2015. URL `http://karpathy.github.io/2015/10/25/selfie/`.

[5] Luke Merrick, Colin Cassady, and Abhimanyu Banerjee. Exploration of automatic label generation from social media through classification of instagram selfies. *unpublished*, 2016.

[6] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[7] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. URL `http://arxiv.org/abs/1409.1556`.

[8] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[9] François Chollet. Keras. `https://github.com/fchollet/keras`, 2015.

[10] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the International conference on artificial intelligence and statistics*, page 249–256, 2010.