Hot Dog or Not Hot Dog: An Analysis of Using Google Image in Object Recognition Training Data

Chris Allum, Alex Butler

February 27 2020

Summary

In this report, we will explore the viability of training an object recognition program using only the top 100 results of Google Images for the specified object. Due to the inherent biases "baked in" to Google Images and the limited search results for an object, we hope to analyze what effects these flaws may have on the accuracy of an object recognition program. To do this, we built a hot dog recognition program. Throughout the creation of this project, we found that the program was not very successful in differentiating objects that were not within the training set data, or in other words, the program over-fitted itself to the training set data of hot dogs. We had a 80% success rate in recognizing hot dogs but only a 60% success rate in recognizing non hot dog objects. These results indicate that it is very hard to create an object recognition program trained only on google image results.

Introduction

Often the hardest part of creating an object recognition program is collecting enough data to train that program. The training data set needs to be extensive, diverse, yet clean enough for a program to work with. These conditions are hard to meet. Without setting up a camera and taking photos of objects in closely controlled conditions, one of the easiest ways of collecting images is through something almost everyone has used: Google Images. However, even though collecting image data through Google Images is easy, it is not entirely ethical nor effective in creating a training data set.

The primary ethical concern lies in the fact that Google Image search results are inherently biased. Google returns images that are associated with the search term, not necessarily the search term itself. Because of this, it is possible for there to be pretty significant gaps between subject you're looking for and what you'll receive from Google.

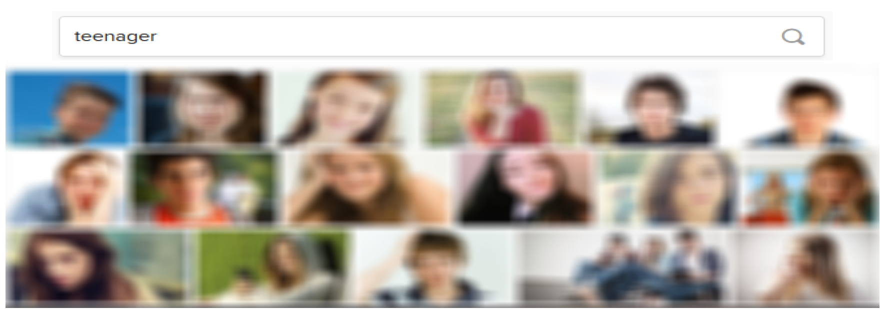

Figure 1

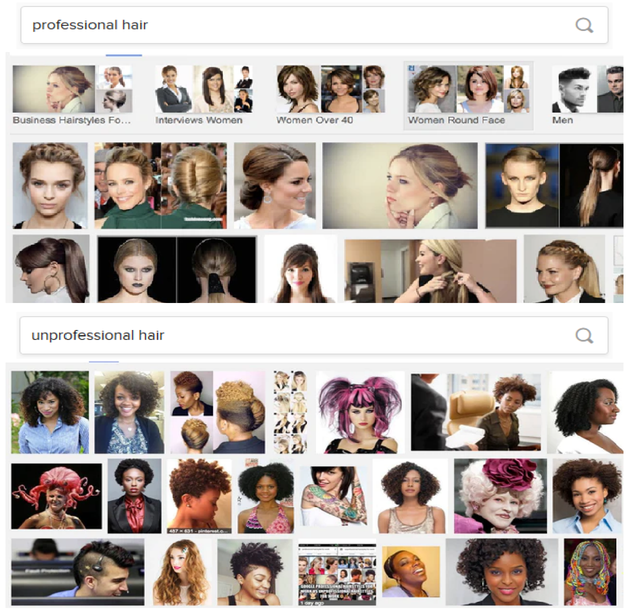

Figure 2

To demonstrate, two cases come to mind. The former showing the nearly explicit bias in a single term and the latter being a case of bias across different search terms. When using the search term “teenager” (Figure 1) Google returns images mainly consisting of Caucasian models, leaving no diversity represented in these images. The second example can be found by comparing the search results of two similar terms: specifically, ”professional hair” and ”unprofessional hair.” As shown in Figure 2, there is a clear discrepancy among what Google considers ”professional” and ”unprofessional”, a division that is split almost directly down the lines of race.

While Google Images’ algorithm does not inherently hide or mislabel minority groups, it does allow preexisting biases to propagate and heighten across the system. As synthesized by Kay, Matthew, & Munson, it can be expected that for most searches to have irregularities in the demographics of the search results and the real demographics of the search term of about 5%. However, irregularities greater than this can be assigned to the labeling group’s tendency to match labels with their own understanding of the stereotypes surrounding the subject matter. This phenomenon is known as *Qualitative differential representation*, the latter portrays this well.

Back to the first example, the first solo photo of a non-white teenager is the 33rd image. On the flip side, the search term “black teenager” results in many photos of black teenagers, but almost all of the photos are linked to articles discussing recent deaths of black teenagers, or shootings. To the average observer the

Using Google Images to collect training set data could be particularly harmful to anyone who isn’t seen as normalized humans in the eyes of Google Images. In this example, if a face recognition system was built around “teenager” images, it may not recognize any face that is not Caucasian.

With these biases in mind, we decided to test if building an image recognition program was even possible given the limitations of Google Images. Very few of the images have clear backgrounds, and the conditions of each photo differ drastically. To test the viability of creating an object recognition system and to see how the inherent google image bias affects the program, we attempted to make a hot dog or not-hot dog program. The premise of our test is that if we can quickly train a program to recognize a hot dog using only Google Image results, it is highly likely that an algorithm can be made to recognize any object, using only trained data sourced from what Google images has to offer.

We were initially inspired to create a hot dog recognition program from the iconic *Silicon Valley* TV show scene in which the character Jian Yang creates a hot dog or not-hot dog program. Furthermore, hot dogs are also a good benchmark to test an object recognition system on. While visually simple and virtually ubiquitous, Hot dogs are surprisingly difficult to detect in comparison to other foods such as fruit or other fast foods.

Our algorithm is trained on a set of approximately 100 images that are scraped from Google Images' results page. These images are first cleaned by isolating the hot dog subject on a white background and oriented them across the center horizontal line of a 72x128 frame. We then train our algorithm on these photos. We then test the accuracy of our algorithm by feeding it 50 new images of hot dogs and 50 images of random objects. The accuracy of the algorithm is determined by what percentage of it's guesses are correct.

Methods

To build our hot dog recognition program, we first constructed a set of 89 images of hot dogs. We removed the background noise in each photo and reorientated the hot dog so that it lays horizontal across the center of a 72x128 frame. We then vectorize each photo into a 9216x1x3 matrix (one column vector for each set of RGB values). We then combined the vectorized images into a 9216x89x3 matrix.

$$HotDog_{red} = \begin{bmatrix} r_{(1,1)} & \cdots & r_{(1,128)} \\ \vdots & & \vdots \\ \vdots & \ddots & \vdots \\ \vdots & & \vdots \\ r_{(72,1)} & \cdots & r_{(72,128)} \end{bmatrix}$$

Original red values of hot dog image.

$$HotDog_{red} = \begin{bmatrix} r_{(1,1)} \\ \vdots \\ \vdots \\ \vdots \\ r_{(9216,1)} \end{bmatrix}$$

Vectorized red values of hot dog image.

$$HotDogs_{red} = \begin{bmatrix} r_{(1,1)} & \cdots & r_{(1,89)} \\ \vdots & & \vdots \\ \vdots & \ddots & \vdots \\ \vdots & & \vdots \\ r_{(9216,1)} & \cdots & r_{(9216,89)} \end{bmatrix}$$

Collection of 89 vectorized red images values.

We then took this training set matrix and created a 9216x9216x3 covariance matrix. Using this covariance matrix, we were then able to calculate the eigenvectors for the training data set. Our program uses the first 20 principal eigenvectors which represent 20 'building blocks' by which the training set data could be built.

$$Covariance_{red} = \begin{bmatrix} r_{(1,1)} & \cdots & r_{(1,9216)} \\ \vdots & & \vdots \\ \vdots & \ddots & \vdots \\ \vdots & & \vdots \\ r_{(9216,1)} & \cdots & r_{(9216,9216)} \end{bmatrix}$$

Covariance of red image values.

$$EigenVectors_{red} = \begin{bmatrix} r_{(1,1)} & \cdots & r_{(1,20)} \\ \vdots & & \vdots \\ \vdots & \ddots & \vdots \\ \vdots & & \vdots \\ r_{(9216,1)} & \cdots & r_{(9216,20)} \end{bmatrix}$$

First 20 principal EigenVectors of red hot dogs image values.

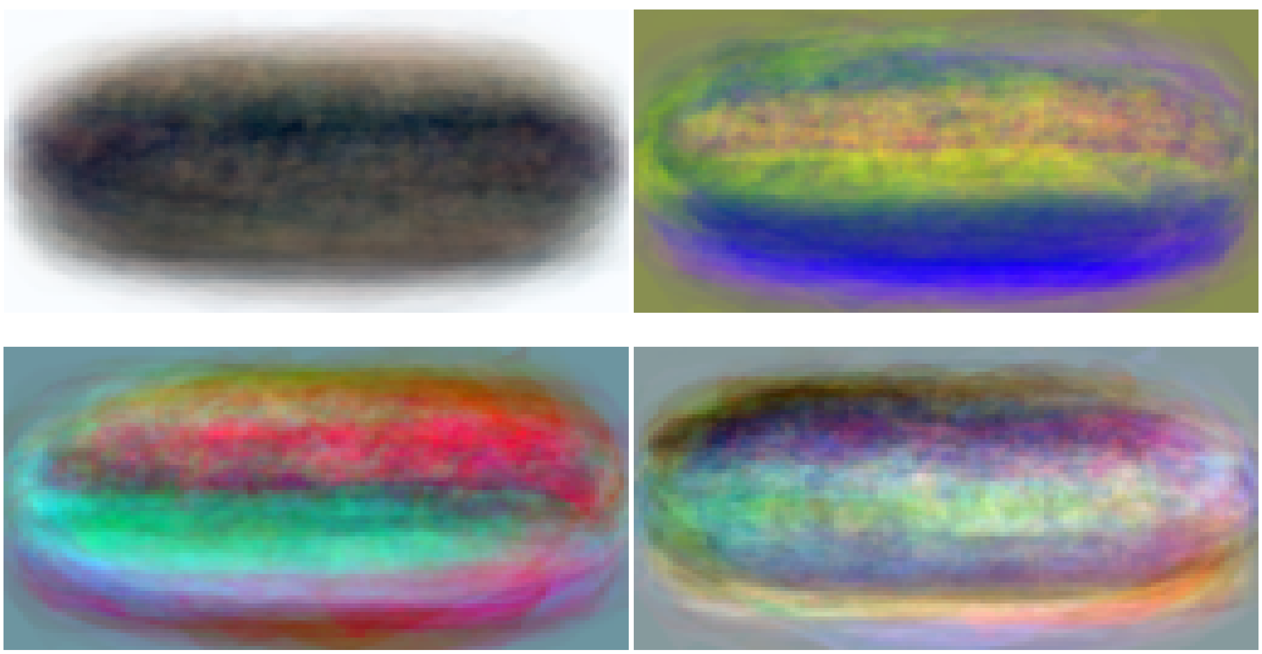

Figure 3: Eigenvectors of the training data set.

Our approach for classifying whether an object was a hot dog was or not a hot dog was to take an image of the object and to project it onto the eigenvector space. We then found the correlation between this compressed image and the original image. Images of hot dogs would not change by a significant margin when they were compressed by eigenvectors, while images of non hot dogs would be deformed by a significant amount.

We then compare the correlation values between these two images. If this correlation is greater than .77, our program considers it a hot dog. If this correlation is less than .77, our program considers it not a hot dog.

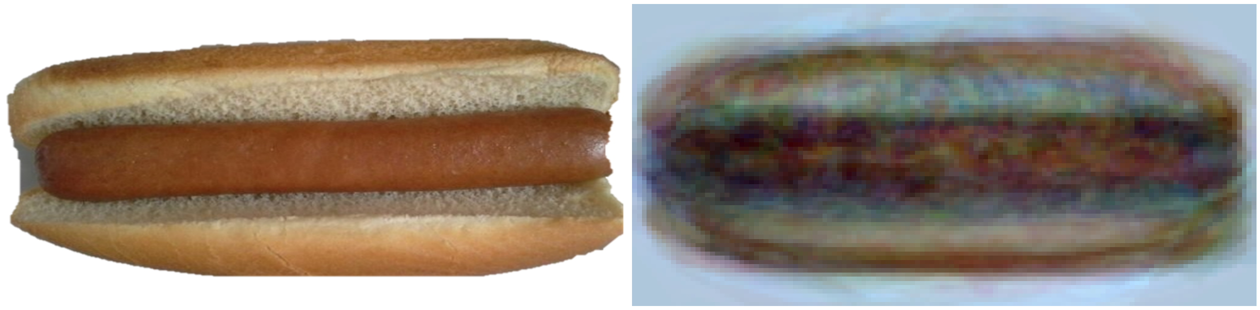

Figure 4: Correlation of hot dog with eigenvector reconstruction = .9006

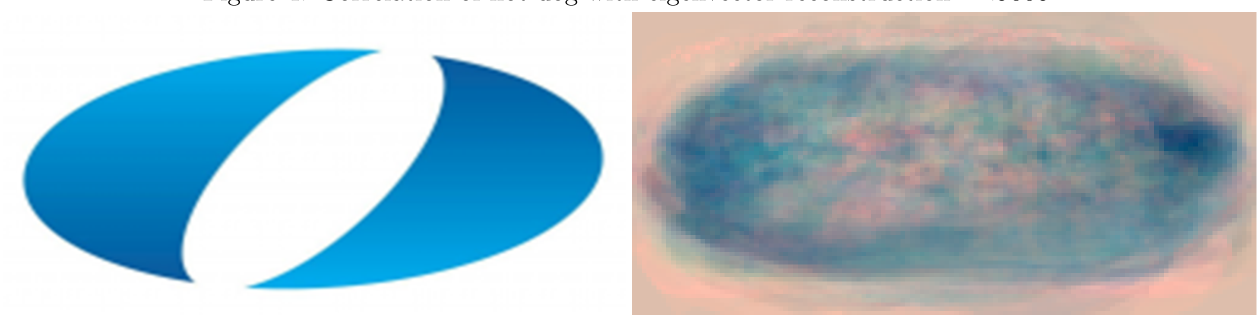

Figure 5: Correlation of Olin Logo with eigenvector reconstruction = .6567

We then tested our program with a testing data set of 100 images, 50 of them hot dogs, and 50 of them not hot dogs. After adjusting the correlation threshold to .77, we achieved a maximum accuracy in recognition of 70%.

Findings

Our algorithm has a 70% success rate in determining whether a photo is a hot dog or not. Broken down, the program recognized 80% of the hot dog test images as hot dogs, and only 60% of the non-hot dog test images as not-hot dogs. Overall, this is a 40% false positive rate and a 20% false negative rate.

The greatest short coming of the algorithm was its inability to correctly label "not hot dog" objects that were round or oblong in nature. For example, it mistook a banana as a hot dog despite the fact that, to a human, they look nothing alike. The program also struggled to recognize hot dogs as such if they were incomplete or if the photo was taken at an unusual angle. For example, if a hot dog did not have the full bread bun in the photo, the program was more likely to label it as "not a hot dog".

Figure 6: Images correctly identified as "Not Hot Dog"

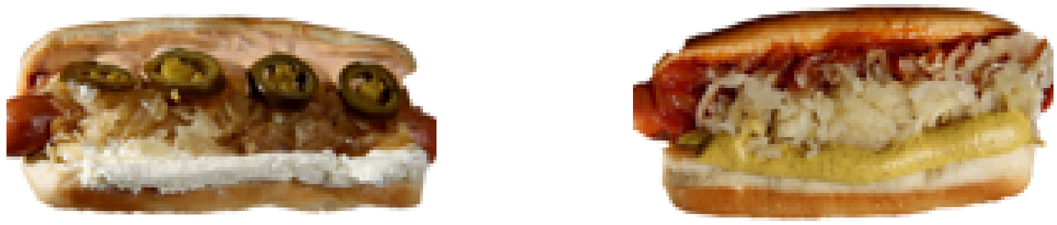

Figure 7: Images correctly identified as Hot Dog

Figure 8: Images incorrectly identified as "Hot Dog"

Figure 9: Images incorrectly identified as Not Hot Dog

One thing that we found particularly interesting was that increasing the number of eigenvectors did not increase the accuracy. While increasing the number of eigenvectors made it easier to compress the hot dogs without changing them much, it also made it easier to compress the non hot dogs. As a result, the false negative rate decreased but the false positive rate also increased. With more eigenvectors the overall accuracy of the program decreases. If we use 50 eigenvectors, accuracy in recognizing hot dogs jumps to 96%, but accuracy in recognizing non-hot dogs decreases to 30%, for a total accuracy of 63%.

Based on these findings, we can conclude that even though our algorithm had an accuracy of 70%, it still demonstrates the failures of using Google Images to build a training data set. In order to build up our data set, we used every single hot dog picture that we found on Google Images that was suitable and cleaned them up to the best of our ability. Even after scraping the entire google search page for the training set and after optimizing the number of eigenvectors, we still had issues with extremely high false positive rates. It was at this point that the balancing act of this program became clear: by increasing the number of eigenvectors we use in our program to more accurately recognize hot dogs, the program becomes worse at recognizing non-hot dogs, and vice versa.

Recommendations

Google images simply did not have enough images for us to create a reliable hot dog or not recognition system. Given this, we believe that it is impossible to create a reliable data set for a more complicated object recognition system such as face recognition. If one cannot trust google images to provide good enough data to make a hot dog recognition system (and frankly, a hot dog isn't that complex of an object to recognize; it is relatively uniform), we do not believe that one should trust google images in making a more complex object recognition system where there may be more prevalent examples of bias and or where that stakes may be higher. For example, using google images to collect face data result a disproportionately high number Caucasian faces, or images of criminals may result in disproportionately high numbers of ethnic minorities.

In conclusion, based off of the failures of our experimental object recognition program, we recommend that one should only use Google Image search results to build training data sets for algorithms where accuracy is not crucial and the results are not far-reaching. One should not use Google Images to create any object recognition program where the results have serious consequences or impact due to its limitations and inherent biases.

References

Kay, Matthew, et al. “Unequal Representation and Gender Stereotypes in Image Search Results for Occupations.” Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI ’15, Apr. 2015, doi:10.1145/2702123.2702520.

Behm-Morawitz, E., Mastro, D. The Effects of the Sexualization of Female Video Game Characters on Gender Stereotyping and Female Self-Concept. Sex Roles 61, 808–823 (2009). https://doi.org/10.1007/s11199-009-9683-8

Baker, Paul, and Amanda Potts. “‘Why Do White People Have Thin Lips?’ Google and the Perpetuation of Stereotypes via Auto-Complete Search Forms.” Critical Discourse Studies, vol. 10, no. 2, 2013, pp. 187–204., doi:10.1080/17405904.2012.744320.

O. A. Arigbabu, S. M. S. Ahmad, W. A. W. Adnan, S. Yussof, V. Iranmanesh and F. L. Malallah, ”Gender recognition on real world faces based on shape representation and neural network,” 2014 International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, 2014, pp. 1-5.