ML Practicum: Image Classification

Learn how Google developed the state-of-the-art image classification model powering search in Google Photos. Get a crash course on convolutional neural networks, and then build your own image classifier to distinguish cat photos from dog photos.

Estimated Completion Time: 90–120 minutes

Prerequisites

- Machine Learning Crash Course (https://developers.google.com/machine-learning/crash-course/) or equivalent experience with ML fundamentals
- Proficiency in programming basics, and some experience coding in Python

Note: The coding exercises in this practicum use the Keras (https://keras.io/) API. Keras is a high-level deep-learning API for configuring neural networks. It is available both as a standalone library and as a module within TensorFlow. (https://www.tensorflow.org/api_docs/python/tf/keras)

Prior experience with Keras is not required for the Colab exercises, as code listings are heavily commented and explained step by step. Comprehensive API documentation is also available on the Keras site (https://keras.io/).

Introduction

In May 2013, Google released search for personal photos (https://search.googleblog.com/2013/05/finding-your-photos-more-easily-with.html), giving users the ability to retrieve photos in their libraries based on the objects present in the images.

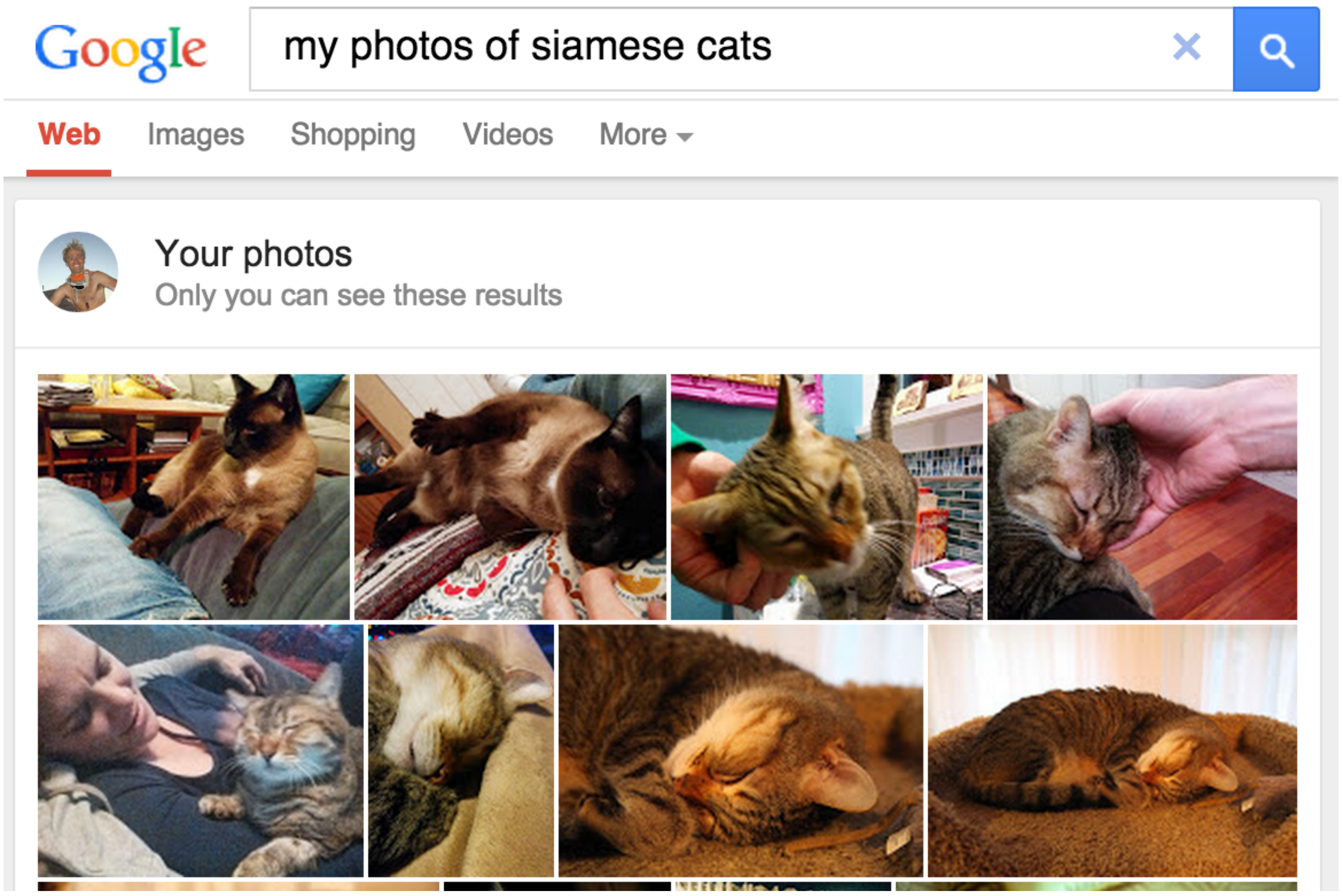

Figure 1. Google Photos search for Siamese cats delivers the goods!

The feature, later incorporated into Google Photos (https://googleblog.blogspot.com/2015/05/picture-this-fresh-approach-to-photos.html) in 2015, was widely perceived as a game-changer, a proof of concept that computer vision software could classify images to human standards, adding value in several ways:

- Users no longer needed to tag photos with labels like "beach" to categorize image content, eliminating a manual task that could become quite tedious when managing sets of hundreds or thousands of images.
- Users could explore their collection of photos in new ways, using search terms to locate photos with objects they might never have tagged. For example, they could search for "palm tree" to surface all their vacation photos that had palm trees in the background.
- Software could potentially "see" taxonomical distinctions that end users themselves might not be able to perceive (e.g., distinguishing Siamese and Abyssinian cats), effectively augmenting users' domain knowledge.

How Image Classification Works

Image classification is a supervised learning problem: define a set of target classes (objects to identify in images), and train a model to recognize them using labeled example photos. Early computer vision models relied on raw pixel data as the input to the model. However, as shown in Figure 2, raw pixel data alone doesn't provide a sufficiently stable representation to encompass the myriad variations of an object as captured in an image. The position of the object, background behind the object, ambient lighting, camera angle, and camera focus all can produce fluctuation in raw pixel data; these differences are significant enough that they cannot be corrected for by taking weighted averages of pixel RGB values.

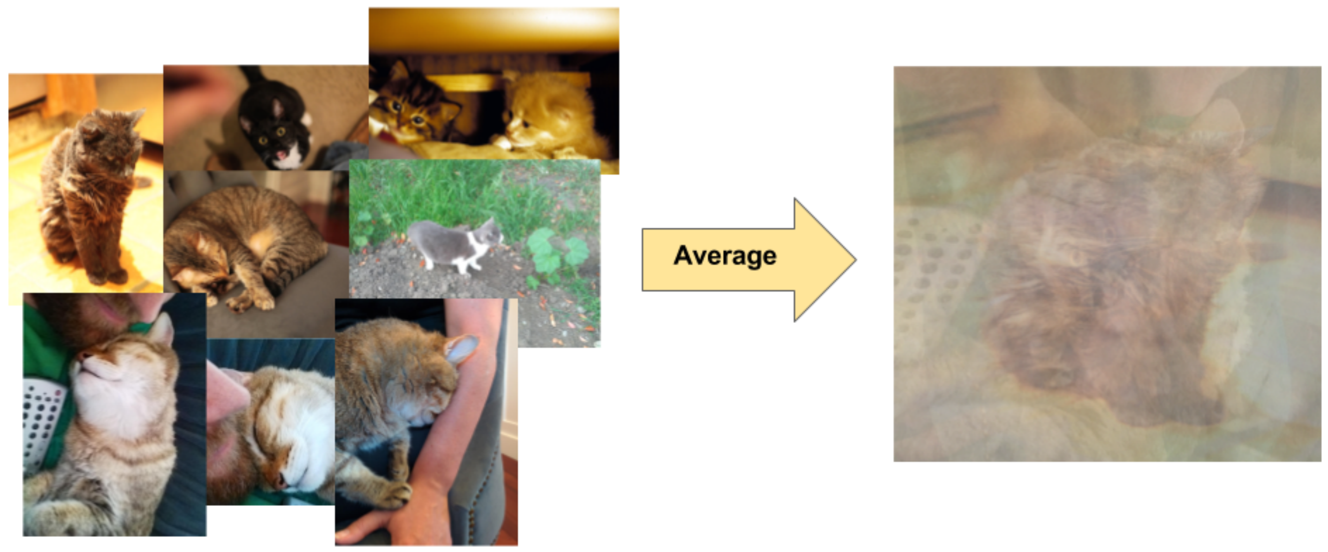

*Figure 2. **Left**: Cats can be captured in a photo in a variety of poses, with different backdrops and lighting conditions. **Right**: averaging pixel data to account for this variety does not produce any meaningful information.*

To model objects more flexibly, classic computer vision models added new features derived from pixel data, such as color histograms (https://wikipedia.org/wiki/Color_histogram), textures, and shapes. The downside of this approach was that feature engineering (http://goto.google.com/mlccss/representation/feature-engineering) became a real burden, as there were so many inputs to tweak. For a cat classifier, which colors were most relevant? How flexible should the shape definitions be? Because features needed to be tuned so precisely, building robust models was quite challenging, and accuracy suffered.

Key Terms

- feature engineering (https://developers.google.com/machine-learning/glossary#feature_engineering)

Next
Convolutional Neural Networks →
(https://developers.google.com/machine-learning/practica/image-classification/convolutional-neural-networks)