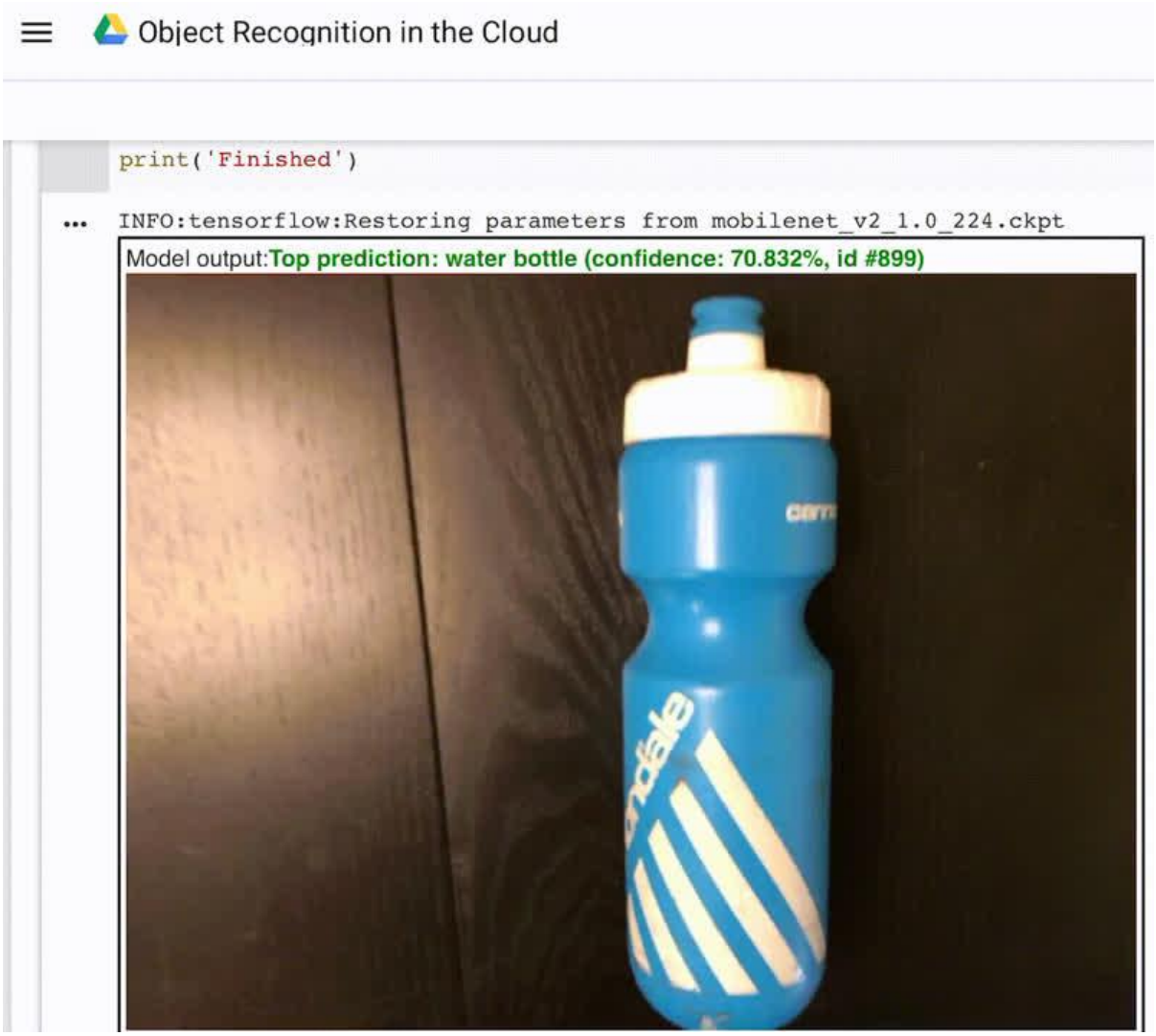
Basic TensorFlow Object Recognition on any Computer or iOS device with Google Colab

Created by Andrew Reusch



https://learn.adafruit.com/basic-tensorflow-object-recognition-in-the-cloud-google-colab

Last updated on 2023-08-29 04:17:15 PM EDT

Table of Contents

Welcome 3

- What is Object Detection?
- Mobilenet v2

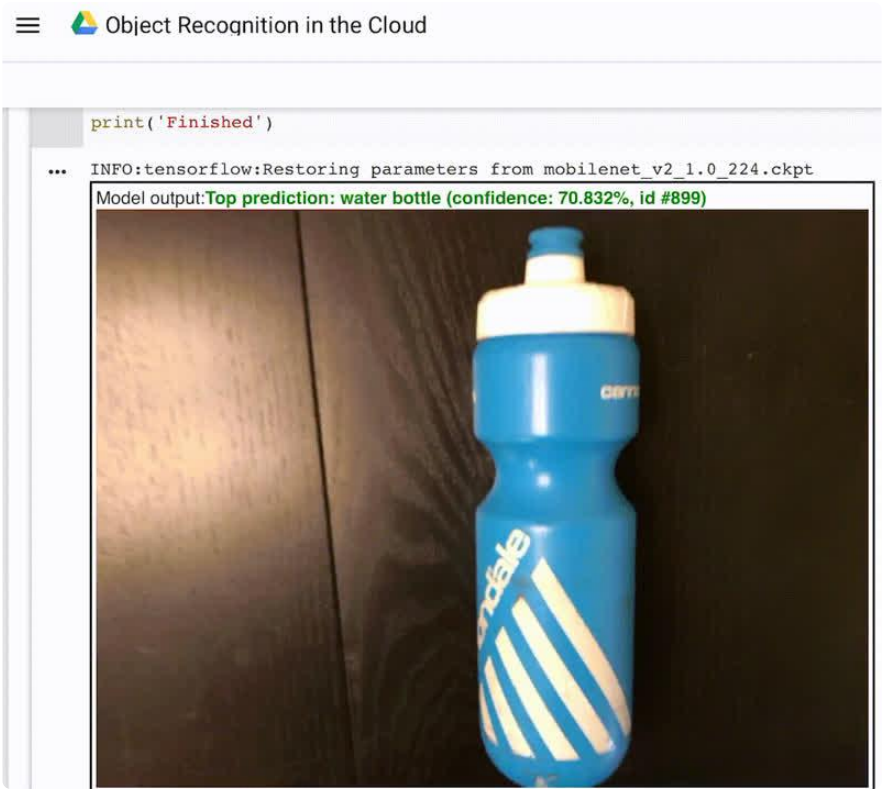
Opening the Notebook 5

- Making changes

Running the Detector 6

- What should happen?
- Aside: Behind the Scenes
- Configuring TensorFlow
- Starting the Demo
- Tips and Tricks

Welcome



You've heard about Machine Learning and AI - and you want to see what all the fuss is about. But you don't want to spend all your time installing bazel and Jupyter? Or maybe you're running an old computer or your only computer is a phone! What now, give up? Never! Thanks to Google Colab, you can run TensorFlow in a browser window, and all the computation is handled on Google's cloud service for free. It's a great way to dabble, without all the setup

We've hacked together a Colab notebook that will use your computer/laptop/phone camera or webcam to get images which are then categorized with the Mobilenet v2 model () to detect one of ~1000 different objects it recognizes. This way you can see what Mobilenet v2 can do, instantly!

For this tutorial you will need a free Google account, a computer, phone or tablet with a camera or webcam, and a recent browser

What is Object Detection?

Object Detection algorithms look at pictures and list out the objects they see. Take look at the example below:

The algorithm produces two outputs here:

- The identified object, given both by name (water bottle) and an id number
- Confidence Level, a measure of the algorithm's certainty

Early object detection algorithms used basic heuristics about the geometry of an object (for example, a tennis ball is usually round and green). While these had some successes, they were difficult to create and were prone to some hilarous false-positives.

Mobilenet v2

In recent years, a technology called neural nets has made it possible to let computers develop the heuristics on their own, by showing them a large number of examples. M obilenet v2 is one of the well-known models beacuse it's optimized to run on devices like your cell phone or a raspberry pi.

The authors of Mobilenet v2 claim it runs in 143ms on a Pixel 1. It can recognize 1000 different objects, including:

- animals, like fish, birds, and turtles
- household items, like brooms, coffee mugs, and pens
- airplanes, golf carts, mopeds

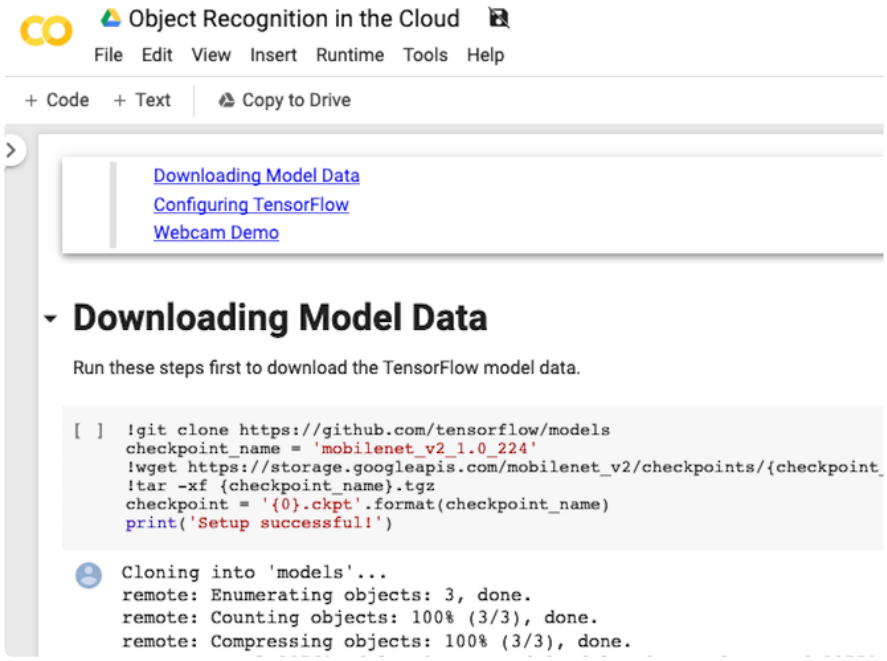
These objects are taken from a popular set of images used to develop object detection algorithms.

Opening the Notebook

To get started, click here to open our notebook in Colab.

When you click the link, it should take you to a page that looks like this:

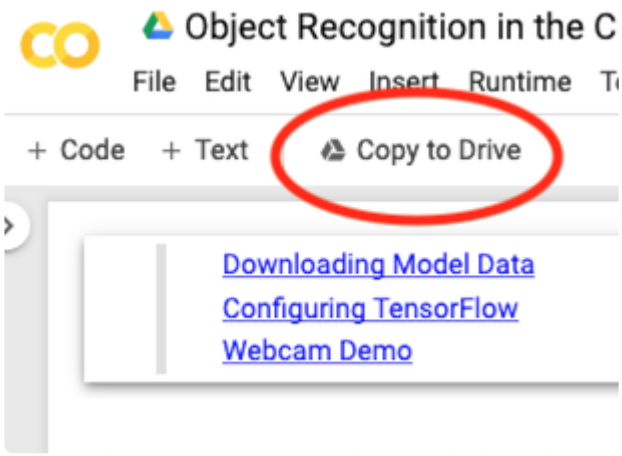
> We recommend the Chrome browser on a desktop/laptop. On iOS (phone/tablet) use the default Safari browser



Making changes

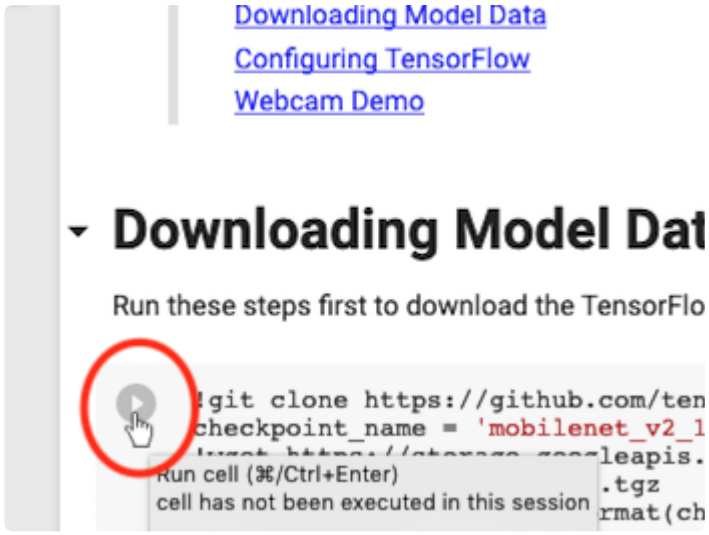
This link will open the notebook in playground mode, meaning it's read-only. But, you can make changes if you want to!

To make changes, you'll need to save a copy by clicking "Copy to Drive" like in the image below.



Running the Detector

To get started, move your mouse cursor over the `[ ]` box to the left of the first code snippet, underneath the Downloading Model Data header. It will change to a "Play" icon. Click on this icon.

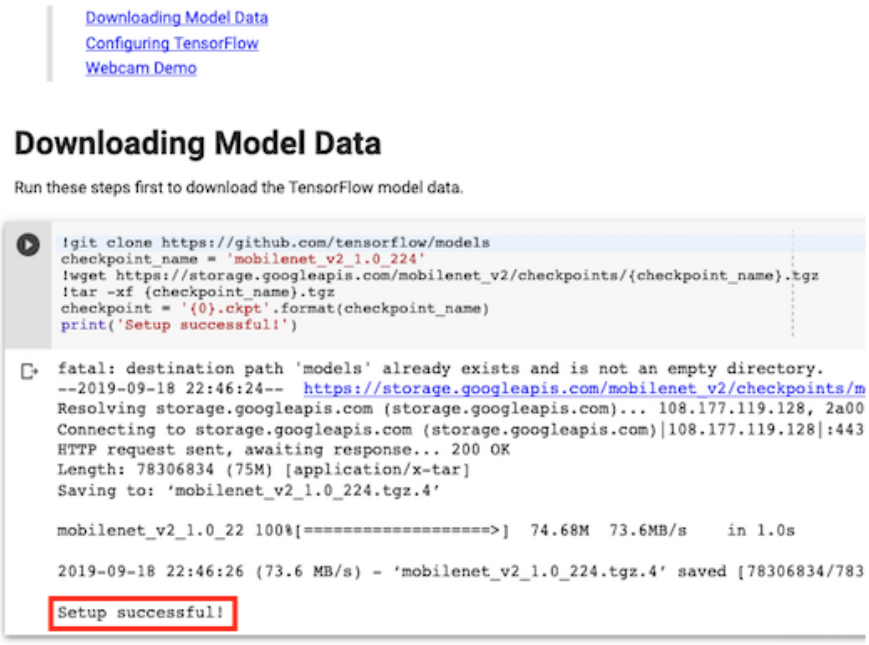


At this point, you may be prompted to sign in to your Google account. You'll need to sign in before you can continue with this guide.

What should happen?

After typically 20 seconds or so, you'll see the notebook come to life. The previous output will vanish and you'll see it replaced with the result of running on your new run time (see the section titledAside below for more about what a runtime is).

When you see `Setup Successful!`, you know you've finished this step. You can open another copy of the notebook and compare it to our previous run, just to make sure it looks correct.



Aside: Behind the Scenes

Each time you open a Colab notebook, Google lets you temporarily use a computer in their datacenter to run your code. This computer is running a program called the runtime, which lets you play around with TensorFlow without having to worry about how fast your computer and without needing to buy an expensive graphics card.

When you close your Colab notebook, Google replaces your runtime with a brand new one, and releases your machine to someone else. This means that each time you come back, you'll need to set up the machine from scratch.

The first cell in the notebook does just this. It downloads the TensorFlow Model that will be used to recognize objects.

Configuring TensorFlow

Scroll down to the next code block, titled Configuring TensorFlow. Again, move your mouse over the `[ ]` box on the left, and click to run the code block.

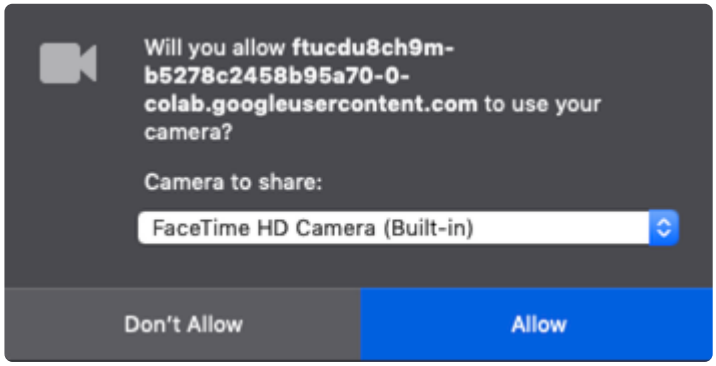
You might see a couple of warnings, like the ones below. You can ignore these.

```
WARNING:tensorflow:
The TensorFlow contrib module will not be included in TensorFlow 2.0.
For more information, please see:
  * https://github.com/tensorflow/community/blob/master/rfcs/20180907-contrib-
sunset.md
  * https://github.com/tensorflow/addons
  * https://github.com/tensorflow/io (for I/O related ops)
If you depend on functionality not listed there, please file an issue.

WARNING:tensorflow:From /content/models/research/slim/nets/mobilenet/mobilenet.py:
397: The name tf.nn.avg_pool is deprecated. Please use tf.nn.avg_pool2d instead.

WARNING:tensorflow:From /content/models/research/slim/nets/mobilenet/mobilenet.py:
364: The name tf.variable_scope is deprecated. Please use
tf.compat.v1.variable_scope instead.
```

Starting the Demo

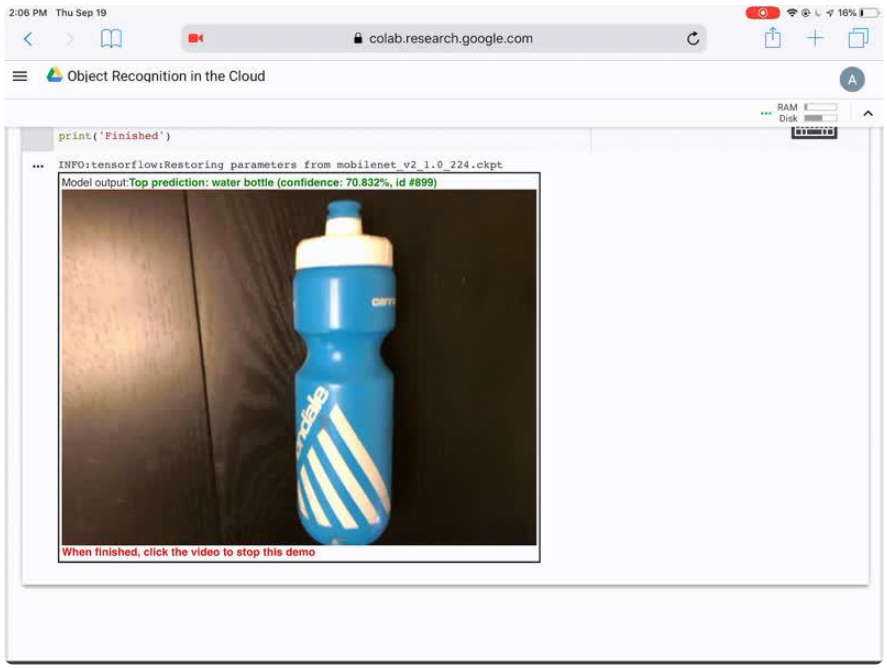
Now, scroll to the final code block, titled Demo and run it in the same way. This block of code won't stop until you tell it to.

After you click the play button, your browser should ask for permission to use your webcam. In Firefox, it looks like this:



Be sure to click Allow.

Finally, scroll down and you should see video from your camera appear on screen. Above the video, you'll see the output from the Object Detector.



Tips and Tricks

Play around with the detector to see how it performs best. Here are some hints from my experiments so far:

- Make sure the object takes up most of the frame. This demo reports only the most confident object it sees, even if it sees two or three others.
- Try with a uniform background for best results.
- The prediction text turns green when the model is more than 70% confident in what it sees. Sometimes, the model is much more confident--as high as 95%. Other times, it's less than 20% confident--it's probably not seeing what it says it is. You can change the 70% threshold to fit your proejct. Try adjusting the threshold and see what works for you (hint: look for "0.7" near the end of the code).