TensorFlow Lite Object Detection API in Colab

Author: Evan Juras, EJ Technology Consultants

Last updated: 1/28/23

GitHub: TensorFlow Lite Object Detection

Introduction

This notebook uses the TensorFlow 2 Object Detection API to train an SSD-MobileNet model or EfficientDet model with a custom dataset and convert it to TensorFlow Lite format. By working through this Colab, you'll be able to create and download a TFLite model that you can run on your PC, an Android phone, or an edge device like the Raspberry Pi.

Custom SSD-MobileNet-FPNLite model in action!

I also made a YouTube video that walks through this guide step by step. I use a coin detection model as an example for the video. I recommend following along with the video while working through this notebook.

[Click here to go to the video!]

Important note: This notebook will be continuously updated to make sure it works with newer versions of TensorFlow. If you see any differences between the YouTube video and this notebook, always follow the notebook!

Working in Colab

Colab provides a virtual machine in your browser complete with a Linux OS, filesystem, Python environment, and best of all, a free GPU. It comes with most TensorFlow backend requirements (like CUDA and cuDNN) pre-installed. Simply click the play button on sections of code in this notebook to execute them on the virtual machine.

> *Note: Make sure you're using a GPU-equipped machine by going to "Runtime" -> "Change runtime type" in the top menu bar, and then selecting "GPU" from the Hardware accelerator dropdown.*

This Colab notebook uses TensorFlow 2. If you'd like to use TensorFlow 1, please see my TF1 Colab notebook.

Navigation

This is a long notebook! Each step of the training process has its own section. Click the arrow next to the heading for each section to expand it. You can use the table of contents in the left sidebar to jump from section to section.

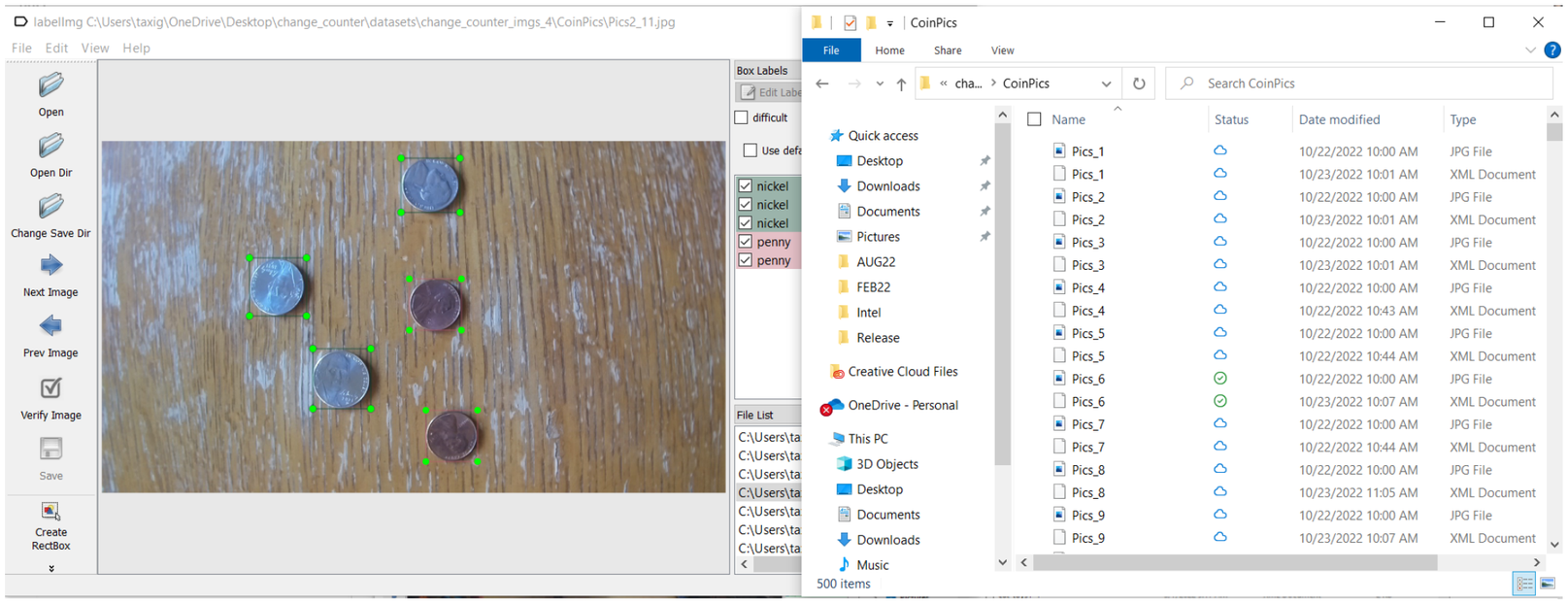
1. Gather and Label Training Images

Before we start training, we need to gather and label images that will be used for training the object detection model. A good starting point for a proof-of-concept model is 200 images. The training images should have random objects in the image along with the desired objects, and should have a variety of backgrounds and lighting conditions.

Watch the YouTube video below for instructions and tips on how to gather and label images for training an object detection model.

[Watch this video to learn how to capture and label images.]

When you've finished gathering and labeling images, you should have a folder full of images and corresponding .xml data annotation file for each image. An example of a labeled image and the image folder for my coin detector model are shown below.



Labeled Image **Image Folder**

2. Install TensorFlow Object Detection Dependencies

First, we'll install the TensorFlow Object Detection API in this Google Colab instance. This requires cloning the TensorFlow models repository and running a couple installation commands. Click the play button to run the following sections of code.

The latest version of TensorFlow this Colab has been verified to work with is TF v2.8.0.

```
# Clone the tensorflow models repository from GitHub
!git clone --depth 1 https://github.com/tensorflow/models
```

```
Cloning into 'models'...
remote: Enumerating objects: 3909, done.
remote: Counting objects: 100% (3909/3909), done.
remote: Compressing objects: 100% (3014/3014), done.
remote: Total 3909 (delta 1129), reused 1996 (delta 842), pack-reused 0
Receiving objects: 100% (3909/3909), 49.65 MiB | 24.08 MiB/s, done.
Resolving deltas: 100% (1129/1129), done.
```

```
# Copy setup files into models/research folder
%%bash
cd models/research/
protoc object_detection/protos/*.proto --python_out=.
#cp object_detection/packages/tf2/setup.py .
```

```
# Modify setup.py file to install the tf-models-official repository targeted at TF v2.8.0
import re
with open('/content/models/research/object_detection/packages/tf2/setup.py') as f:
    s = f.read()

with open('/content/models/research/setup.py', 'w') as f:
    # Set fine_tune_checkpoint path
    s = re.sub('tf-models-official>=2.5.1',
               'tf-models-official==2.8.0', s)
    f.write(s)
```

```
# Install the Object Detection API
!pip install /content/models/research/

# Need to downgrade to TF v2.8.0 due to Colab compatibility bug with TF v2.10 (as of 10/03/22)
!pip install tensorflow==2.8.0
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Processing ./models/research
  Preparing metadata (setup.py) ... done
Collecting avro-python3 (from object-detection==0.1)
  Downloading avro-python3-1.10.2.tar.gz (38 kB)
  Preparing metadata (setup.py) ... done
Collecting apache-beam (from object-detection==0.1)
  Downloading apache_beam-2.48.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (14.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 14.3/14.3 MB 80.3 MB/s eta 0:00:00
Requirement already satisfied: pillow in /usr/local/lib/python3.10/dist-packages (from object-detection==0.1) (8.4.0)
Requirement already satisfied: lxml in /usr/local/lib/python3.10/dist-packages (from object-detection==0.1) (4.9.2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from object-detection==0.1) (3.7.1)
Requirement already satisfied: Cython in /usr/local/lib/python3.10/dist-packages (from object-detection==0.1) (0.29.34)
Requirement already satisfied: contextlib2 in /usr/local/lib/python3.10/dist-packages (from object-detection==0.1) (0.6.0.post1)
Requirement already satisfied: tf-slim in /usr/local/lib/python3.10/dist-packages (from object-detection==0.1) (1.1.0)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from object-detection==0.1) (1.16.0)
Requirement already satisfied: pycocotools in /usr/local/lib/python3.10/dist-packages (from object-detection==0.1) (2.0.6)
Collecting lvis (from object-detection==0.1)
  Downloading lvis-0.5.3-py3-none-any.whl (14 kB)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from object-detection==0.1) (1.10.1)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from object-detection==0.1) (1.5.3)
Collecting tf-models-official==2.8.0 (from object-detection==0.1)
  Downloading tf_models_official-2.8.0-py2.py3-none-any.whl (2.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.2/2.2 MB 95.0 MB/s eta 0:00:00
Collecting tensorflow_io (from object-detection==0.1)
  Downloading tensorflow_io-0.32.0-cp310-cp310-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (28.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 28.0/28.0 MB 60.6 MB/s eta 0:00:00
Requirement already satisfied: keras in /usr/local/lib/python3.10/dist-packages (from object-detection==0.1) (2.12.0)
Collecting pyparsing==2.4.7 (from object-detection==0.1)
  Downloading pyparsing-2.4.7-py2.py3-none-any.whl (67 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 67.8/67.8 kB 9.6 MB/s eta 0:00:00
Collecting sacrebleu<=2.2.0 (from object-detection==0.1)
  Downloading sacrebleu-2.2.0-py3-none-any.whl (116 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 116.6/116.6 kB 15.6 MB/s eta 0:00:00
Requirement already satisfied: gin-config in /usr/local/lib/python3.10/dist-packages (from tf-models-official==2.8.0->object-detection==0.1) (0.5.0)
Requirement already satisfied: google-api-python-client>=1.6.7 in /usr/local/lib/python3.10/dist-packages (from tf-models-official==2.8.0->object-detection==0.1) (2.84.0)
Requirement already satisfied: kaggle>=1.3.9 in /usr/local/lib/python3.10/dist-packages (from tf-models-official==2.8.0->object-detection==0.1) (1.5.13)
Requirement already satisfied: numpy>=1.15.4 in /usr/local/lib/python3.10/dist-packages (from tf-models-official==2.8.0->object-detection==0.1) (1.22.4)
Requirement already satisfied: oauth2client in /usr/local/lib/python3.10/dist-packages (from tf-models-official==2.8.0->object-detection==0.1) (4.1.3)
Requirement already satisfied: opencv-python-headless in /usr/local/lib/python3.10/dist-packages (from tf-models-official==2.8.0->object-detection==0.1) (4.7.0.72)
Requirement already satisfied: psutil>=5.4.3 in /usr/local/lib/python3.10/dist-packages (from tf-models-official==2.8.0->object-detection==0.1) (5.9.5)
Requirement already satisfied: py-cpuinfo>=3.3.0 in /usr/local/lib/python3.10/dist-packages (from tf-models-official==2.8.0->object-detection==0.1) (9.0.0)
Collecting pyyaml<6.0,>=5.1 (from tf-models-official==2.8.0->object-detection==0.1)
  Downloading PyYAML-5.4.1.tar.gz (175 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 175.1/175.1 kB 23.2 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting sentencepiece (from tf-models-official==2.8.0->object-detection==0.1)
  Downloading sentencepiece-0.1.99-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.3/1.3 MB 87.4 MB/s eta 0:00:00
Collecting seqeval (from tf-models-official==2.8.0->object-detection==0.1)
  Downloading seqeval-1.2.2.tar.gz (43 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 43.6/43.6 kB 5.5 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting tensorflow-addons (from tf-models-official==2.8.0->object-detection==0.1)
```

```
  Downloading tensorflow_addons-0.20.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (591 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 591.0/591.0 kB 60.3 MB/s eta 0:00:00
Requirement already satisfied: tensorflow-datasets in /usr/local/lib/python3.10/dist-packages (from tf-models-official==2.8.0->object-detection==0.1) (4.9.2)
Requirement already satisfied: tensorflow-hub>=0.6.0 in /usr/local/lib/python3.10/dist-packages (from tf-models-official==2.8.0->object-detection==0.1) (0.13.0)
Collecting tensorflow-model-optimization>=0.4.1 (from tf-models-official==2.8.0->object-detection==0.1)
  Downloading tensorflow_model_optimization-0.7.5-py2.py3-none-any.whl (241 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 241.2/241.2 kB 27.6 MB/s eta 0:00:00
Collecting tensorflow-text~=2.8.0 (from tf-models-official==2.8.0->object-detection==0.1)
  Downloading tensorflow_text-2.8.2-cp310-cp310-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (4.9 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.9/4.9 MB 120.5 MB/s eta 0:00:00
Collecting tensorflow~=2.8.0 (from tf-models-official==2.8.0->object-detection==0.1)
  Downloading tensorflow-2.8.4-cp310-cp310-manylinux2010_x86_64.whl (498.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 498.1/498.1 MB 3.2 MB/s eta 0:00:00
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas->object-detection==0.1) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->object-detection==0.1) (2022.7.1)
Collecting portalocker (from sacrebleu<=2.2.0->object-detection==0.1)
  Downloading portalocker-2.7.0-py2.py3-none-any.whl (15 kB)
Requirement already satisfied: regex in /usr/local/lib/python3.10/dist-packages (from sacrebleu<=2.2.0->object-detection==0.1) (2022.10.31)
Requirement already satisfied: tabulate>=0.8.9 in /usr/local/lib/python3.10/dist-packages (from sacrebleu<=2.2.0->object-detection==0.1) (0.8.10)
Collecting colorama (from sacrebleu<=2.2.0->object-detection==0.1)
  Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Requirement already satisfied: absl-py>=0.2.2 in /usr/local/lib/python3.10/dist-packages (from tf-slim->object-detection==0.1) (1.4.0)
Collecting crcmod<2.0,>=1.7 (from apache-beam->object-detection==0.1)
  Downloading crcmod-1.7.tar.gz (89 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 89.7/89.7 kB 11.2 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting orjson<4.0 (from apache-beam->object-detection==0.1)
  Downloading orjson-3.9.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (136 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 137.0/137.0 kB 18.4 MB/s eta 0:00:00
Collecting dill<0.3.2,>=0.3.1.1 (from apache-beam->object-detection==0.1)
  Downloading dill-0.3.1.1.tar.gz (151 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 152.0/152.0 kB 20.7 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: cloudpickle~=2.2.1 in /usr/local/lib/python3.10/dist-packages (from apache-beam->object-detection==0.1) (2.2.1)
Collecting fastavro<2,>=0.23.6 (from apache-beam->object-detection==0.1)
  Downloading fastavro-1.7.4-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.6/2.6 MB 111.6 MB/s eta 0:00:00
Collecting fasteners<1.0,>=0.3 (from apache-beam->object-detection==0.1)
  Downloading fasteners-0.18-py3-none-any.whl (18 kB)
Requirement already satisfied: grpcio!=1.48.0,<2,>=1.33.1 in /usr/local/lib/python3.10/dist-packages (from apache-beam->object-detection==0.1) (1.54.0)
Collecting hdfs<3.0.0,>=2.1.0 (from apache-beam->object-detection==0.1)
  Downloading hdfs-2.7.0-py3-none-any.whl (34 kB)
Requirement already satisfied: httplib2<0.23.0,>=0.8 in /usr/local/lib/python3.10/dist-packages (from apache-beam->object-detection==0.1) (0.21.0)
Collecting objsize<0.7.0,>=0.6.1 (from apache-beam->object-detection==0.1)
  Downloading objsize-0.6.1-py3-none-any.whl (9.3 kB)
Collecting pymongo<5.0.0,>=3.8.0 (from apache-beam->object-detection==0.1)
  Downloading pymongo-4.3.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (492 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 492.9/492.9 kB 51.5 MB/s eta 0:00:00
Requirement already satisfied: proto-plus<2,>=1.7.1 in /usr/local/lib/python3.10/dist-packages (from apache-beam->object-detection==0.1) (1.22.2)
Requirement already satisfied: protobuf<4.24.0,>=3.20.3 in /usr/local/lib/python3.10/dist-packages (from apache-beam->object-detection==0.1) (3.20.3)
Requirement already satisfied: pydot<2,>=1.2.0 in /usr/local/lib/python3.10/dist-packages (from apache-beam->object-detection==0.1) (1.4.2)
Requirement already satisfied: requests<3.0.0,>=2.24.0 in /usr/local/lib/python3.10/dist-packages (from apache-beam->object-detection==0.1) (2.27.1)
Requirement already satisfied: typing-extensions>=3.7.0 in /usr/local/lib/python3.10/dist-packages (from apache-beam->object-detection==0.1) (4.5.0)
Collecting zstandard<1,>=0.18.0 (from apache-beam->object-detection==0.1)
  Downloading zstandard-0.21.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.7 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.7/2.7 MB 105.9 MB/s eta 0:00:00
Requirement already satisfied: pyarrow<12.0.0,>=3.0.0 in /usr/local/lib/python3.10/dist-packages (from apache-beam->object-detection==0.1) (9.0.0)
```

```
Requirement already satisfied: cycler>=0.10.0 in /usr/local/lib/python3.10/dist-packages (from lvis->object-detection==0.1) (0.11.0)
Requirement already satisfied: kiwisolver>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from lvis->object-detection==0.1) (1.4.4)
Requirement already satisfied: opencv-python>=4.1.0.25 in /usr/local/lib/python3.10/dist-packages (from lvis->object-detection==0.1) (4.7.0.72)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->object-detection==0.1) (1.0.7)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->object-detection==0.1) (4.39.3)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->object-detection==0.1) (23.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem==0.32.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow_io->object-detection==0.1) (0.32.0)
Requirement already satisfied: google-auth<3.0.0dev,>=1.19.0 in /usr/local/lib/python3.10/dist-packages (from google-api-python-client>=1.6.7->tf-models-official==2.8.0->object-detection==0.
Requirement already satisfied: google-auth-httplib2>=0.1.0 in /usr/local/lib/python3.10/dist-packages (from google-api-python-client>=1.6.7->tf-models-official==2.8.0->object-detection==0.1)
Requirement already satisfied: google-api-core!=2.0.*,!=2.1.*,!=2.2.*,!=2.3.0,<3.0.0dev,>=1.31.5 in /usr/local/lib/python3.10/dist-packages (from google-api-python-client>=1.6.7->tf-models-o
Requirement already satisfied: uritemplate<5,>=3.0.1 in /usr/local/lib/python3.10/dist-packages (from google-api-python-client>=1.6.7->tf-models-official==2.8.0->object-detection==0.1) (4.1.
Collecting docopt (from hdfs<3.0.0,>=2.1.0->apache-beam->object-detection==0.1)
  Downloading docopt-0.6.2.tar.gz (25 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from kaggle>=1.3.9->tf-models-official==2.8.0->object-detection==0.1) (2022.12.7)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from kaggle>=1.3.9->tf-models-official==2.8.0->object-detection==0.1) (4.65.0)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.10/dist-packages (from kaggle>=1.3.9->tf-models-official==2.8.0->object-detection==0.1) (8.0.1)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.10/dist-packages (from kaggle>=1.3.9->tf-models-official==2.8.0->object-detection==0.1) (1.26.15)
Collecting dnspython<3.0.0,>=1.16.0 (from pymongo<5.0.0,>=3.8.0->apache-beam->object-detection==0.1)
  Downloading dnspython-2.3.0-py3-none-any.whl (283 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 283.7/283.7 kB 29.0 MB/s eta 0:00:00
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0,>=2.24.0->apache-beam->object-detection==0.1) (2.0.12)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0,>=2.24.0->apache-beam->object-detection==0.1) (3.4)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow~=2.8.0->tf-models-official==2.8.0->object-detection==0.1) (1.6.3)
Requirement already satisfied: flatbuffers>=1.12 in /usr/local/lib/python3.10/dist-packages (from tensorflow~=2.8.0->tf-models-official==2.8.0->object-detection==0.1) (23.3.3)
Requirement already satisfied: gast>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow~=2.8.0->tf-models-official==2.8.0->object-detection==0.1) (0.4.0)
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow~=2.8.0->tf-models-official==2.8.0->object-detection==0.1) (0.2.0)
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow~=2.8.0->tf-models-official==2.8.0->object-detection==0.1) (3.8.0)
Collecting keras-preprocessing>=1.1.1 (from tensorflow~=2.8.0->tf-models-official==2.8.0->object-detection==0.1)
  Downloading Keras_Preprocessing-1.1.2-py2.py3-none-any.whl (42 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 42.6/42.6 kB 4.9 MB/s eta 0:00:00
Requirement already satisfied: libclang>=9.0.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow~=2.8.0->tf-models-official==2.8.0->object-detection==0.1) (16.0.0)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.10/dist-packages (from tensorflow~=2.8.0->tf-models-official==2.8.0->object-detection==0.1) (3.3.0)
INFO: pip is looking at multiple versions of tensorflow to determine which version is compatible with other requirements. This could take a while.
Collecting tensorflow~=2.8.0 (from tf-models-official==2.8.0->object-detection==0.1)
  Downloading tensorflow-2.8.3-cp310-cp310-manylinux2010_x86_64.whl (498.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 498.5/498.5 MB 3.4 MB/s eta 0:00:00
  Downloading tensorflow-2.8.2-cp310-cp310-manylinux2010_x86_64.whl (498.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 498.0/498.0 MB 3.0 MB/s eta 0:00:00
  Downloading tensorflow-2.8.1-cp310-cp310-manylinux2010_x86_64.whl (498.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 498.0/498.0 MB 3.4 MB/s eta 0:00:00
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from tensorflow~=2.8.0->tf-models-official==2.8.0->object-detection==0.1) (67.7.2)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow~=2.8.0->tf-models-official==2.8.0->object-detection==0.1) (2.3.0)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow~=2.8.0->tf-models-official==2.8.0->object-detection==0.1) (1.14.1)
Collecting tensorboard<2.9,>=2.8 (from tensorflow~=2.8.0->tf-models-official==2.8.0->object-detection==0.1)
  Downloading tensorboard-2.8.0-py3-none-any.whl (5.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.8/5.8 MB 91.3 MB/s eta 0:00:00
Collecting tensorflow-estimator<2.9,>=2.8 (from tensorflow~=2.8.0->tf-models-official==2.8.0->object-detection==0.1)
  Downloading tensorflow_estimator-2.8.0-py2.py3-none-any.whl (462 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 462.3/462.3 kB 48.1 MB/s eta 0:00:00
Collecting keras (from object-detection==0.1)
  Downloading keras-2.8.0-py2.py3-none-any.whl (1.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.4/1.4 MB 64.6 MB/s eta 0:00:00
Requirement already satisfied: dm-tree~=0.1.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow-model-optimization>=0.4.1->tf-models-official==2.8.0->object-detection==0.1) (0.1.8)
Collecting numpy>=1.15.4 (from tf-models-official==2.8.0->object-detection==0.1)
  Downloading numpy-1.24.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (17.3 MB)
```

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 17.3/17.3 MB 68.8 MB/s eta 0:00:00
Requirement already satisfied: pyasn1>=0.1.7 in /usr/local/lib/python3.10/dist-packages (from oauth2client->tf-models-official==2.8.0->object-detection==0.1) (0.5.0)
Requirement already satisfied: pyasn1-modules>=0.0.5 in /usr/local/lib/python3.10/dist-packages (from oauth2client->tf-models-official==2.8.0->object-detection==0.1) (0.3.0)
Requirement already satisfied: rsa>=3.1.4 in /usr/local/lib/python3.10/dist-packages (from oauth2client->tf-models-official==2.8.0->object-detection==0.1) (4.9)
Requirement already satisfied: scikit-learn>=0.21.3 in /usr/local/lib/python3.10/dist-packages (from seqeval->tf-models-official==2.8.0->object-detection==0.1) (1.2.2)
Collecting typeguard<3.0.0,>=2.7 (from tensorflow-addons->tf-models-official==2.8.0->object-detection==0.1)
  Downloading typeguard-2.13.3-py3-none-any.whl (17 kB)
Requirement already satisfied: array-record in /usr/local/lib/python3.10/dist-packages (from tensorflow-datasets->tf-models-official==2.8.0->object-detection==0.1) (0.2.0)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from tensorflow-datasets->tf-models-official==2.8.0->object-detection==0.1) (8.1.3)
Requirement already satisfied: etils[enp,epath]>=0.9.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow-datasets->tf-models-official==2.8.0->object-detection==0.1) (1.2.0)
Requirement already satisfied: promise in /usr/local/lib/python3.10/dist-packages (from tensorflow-datasets->tf-models-official==2.8.0->object-detection==0.1) (2.3)
Requirement already satisfied: tensorflow-metadata in /usr/local/lib/python3.10/dist-packages (from tensorflow-datasets->tf-models-official==2.8.0->object-detection==0.1) (1.13.1)
Requirement already satisfied: toml in /usr/local/lib/python3.10/dist-packages (from tensorflow-datasets->tf-models-official==2.8.0->object-detection==0.1) (0.10.2)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.10/dist-packages (from astunparse>=1.6.0->tensorflow~=2.8.0->tf-models-official==2.8.0->object-detection==0.1) (0.4
Requirement already satisfied: importlib_resources in /usr/local/lib/python3.10/dist-packages (from etils[enp,epath]>=0.9.0->tensorflow-datasets->tf-models-official==2.8.0->object-detection=
Requirement already satisfied: zipp in /usr/local/lib/python3.10/dist-packages (from etils[enp,epath]>=0.9.0->tensorflow-datasets->tf-models-official==2.8.0->object-detection==0.1) (3.15.0)
Requirement already satisfied: googleapis-common-protos<2.0dev,>=1.56.2 in /usr/local/lib/python3.10/dist-packages (from google-api-core!=2.0.*,!=2.1.*,!=2.2.*,!=2.3.0,<3.0.0dev,>=1.31.5->go
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from google-auth<3.0.0dev,>=1.19.0->google-api-python-client>=1.6.7->tf-models-official==2.8
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=0.21.3->seqeval->tf-models-official==2.8.0->object-detection==0.1) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=0.21.3->seqeval->tf-models-official==2.8.0->object-detection==0.1) (3.1.0)
Collecting google-auth-oauthlib<0.5,>=0.4.1 (from tensorboard<2.9,>=2.8->tensorflow~=2.8.0->tf-models-official==2.8.0->object-detection==0.1)
  Downloading google_auth_oauthlib-0.4.6-py2.py3-none-any.whl (18 kB)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.9,>=2.8->tensorflow~=2.8.0->tf-models-official==2.8.0->object-detection==0.1) (3
Collecting tensorboard-data-server<0.7.0,>=0.6.0 (from tensorboard<2.9,>=2.8->tensorflow~=2.8.0->tf-models-official==2.8.0->object-detection==0.1)
  Downloading tensorboard_data_server-0.6.1-py3-none-manylinux2010_x86_64.whl (4.9 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.9/4.9 MB 74.8 MB/s eta 0:00:00
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.9,>=2.8->tensorflow~=2.8.0->tf-models-official==2.8.0->object-dete
Requirement already satisfied: werkzeug>=0.11.15 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.9,>=2.8->tensorflow~=2.8.0->tf-models-official==2.8.0->object-detection==0.1)
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.10/dist-packages (from python-slugify->kaggle>=1.3.9->tf-models-official==2.8.0->object-detection==0.1) (1.3)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.10/dist-packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorboard<2.9,>=2.8->tensorflow~=2.8.0->tf-models
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.10/dist-packages (from werkzeug>=0.11.15->tensorboard<2.9,>=2.8->tensorflow~=2.8.0->tf-models-official==2.8.0->obje
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.10/dist-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard<2.9,>=2.8->tensorflow~
Building wheels for collected packages: object-detection, avro-python3, crcmod, dill, pyyaml, seqeval, docopt
  Building wheel for object-detection (setup.py) ... done
  Created wheel for object-detection: filename=object_detection-0.1-py3-none-any.whl size=1696981 sha256=2010219fa3ddb71f9f267e9e5109502b5c83c14d0bbb4c087520374fcc8d1d59
  Stored in directory: /tmp/pip-ephem-wheel-cache-03d1fx9v/wheels/53/dd/70/2de274d6c443c69d367bd6a5606f95e5a6df61aacf1435ec0d
  Building wheel for avro-python3 (setup.py) ... done
  Created wheel for avro-python3: filename=avro_python3-1.10.2-py3-none-any.whl size=43994 sha256=660b7f6ad09d113b7f42b40c06225f5037f0aaa5d41cef3e17567ae73f332261
  Stored in directory: /root/.cache/pip/wheels/bc/85/62/6cdd81c56f923946b401cecff38055b94c9b766927f7d8ca82
  Building wheel for crcmod (setup.py) ... done
  Created wheel for crcmod: filename=crcmod-1.7-cp310-cp310-linux_x86_64.whl size=37102 sha256=e53f914754b40f1434dd8dc3942eac385ba4707ca219bd3809fbce660cdead4d
  Stored in directory: /root/.cache/pip/wheels/85/4c/07/72215c529bd59d67e3dac29711d7aba1b692f543c808ba9e86
  Building wheel for dill (setup.py) ... done
  Created wheel for dill: filename=dill-0.3.1.1-py3-none-any.whl size=78545 sha256=39e6656e3eb97f10a2e1a88671a71ced70c46bd934542dc5a0452694c4a7af30
  Stored in directory: /root/.cache/pip/wheels/ea/e2/86/64980d90e297e7bf2ce588c2b96e818f5399c515c4bb8a7e4f
  Building wheel for pyyaml (pyproject.toml) ... done
  Created wheel for pyyaml: filename=PyYAML-5.4.1-cp310-cp310-linux_x86_64.whl size=45658 sha256=36dd17cde5003a3009e11d18f6cce59a171d8927910236297486ecfaea92f829
  Stored in directory: /root/.cache/pip/wheels/c7/0d/22/696ee92245ad710f506eee79bb05c740d8abccd3ecdb778683
  Building wheel for seqeval (setup.py) ... done
  Created wheel for seqeval: filename=seqeval-1.2.2-py3-none-any.whl size=16165 sha256=ed184217a63428b2d03c80c829ef3c98743c6008f2a956fa90312b0b315b666f
  Stored in directory: /root/.cache/pip/wheels/1a/67/4a/ad4082dd7dfc30f2abfe4d80a2ed5926a506eb8a972b4767fa
  Building wheel for docopt (setup.py) ... done
  Created wheel for docopt: filename=docopt-0.6.2-py2.py3-none-any.whl size=13707 sha256=6dee173790c00b5f181d04650a9cf278b671885341f505e2c9caecb95cd904a6
  Stored in directory: /root/.cache/pip/wheels/fc/ab/d4/5da2067ac95b36618c629a5f93f809425700506f72c9732fac
Successfully built object-detection avro-python3 crcmod dill pyyaml seqeval docopt
Installing collected packages: tensorflow-estimator, sentencepiece, keras, docopt, crcmod, zstandard, typeguard, tensorflow_io, tensorboard-data-server, pyyaml, pyparsing, portalocker, orjso
  Attempting uninstall: tensorflow-estimator
```

```
Attempting uninstall: tensorflow-estimator
  Found existing installation: tensorflow-estimator 2.12.0
  Uninstalling tensorflow-estimator-2.12.0:
    Successfully uninstalled tensorflow-estimator-2.12.0
Attempting uninstall: keras
  Found existing installation: keras 2.12.0
  Uninstalling keras-2.12.0:
    Successfully uninstalled keras-2.12.0
Attempting uninstall: tensorboard-data-server
  Found existing installation: tensorboard-data-server 0.7.0
  Uninstalling tensorboard-data-server-0.7.0:
    Successfully uninstalled tensorboard-data-server-0.7.0
Attempting uninstall: pyyaml
  Found existing installation: PyYAML 6.0
  Uninstalling PyYAML-6.0:
    Successfully uninstalled PyYAML-6.0
Attempting uninstall: pyparsing
  Found existing installation: pyparsing 3.0.9
  Uninstalling pyparsing-3.0.9:
    Successfully uninstalled pyparsing-3.0.9
Attempting uninstall: numpy
  Found existing installation: numpy 1.22.4
  Uninstalling numpy-1.22.4:
    Successfully uninstalled numpy-1.22.4
Attempting uninstall: google-auth-oauthlib
  Found existing installation: google-auth-oauthlib 1.0.0
  Uninstalling google-auth-oauthlib-1.0.0:
    Successfully uninstalled google-auth-oauthlib-1.0.0
Attempting uninstall: tensorboard
  Found existing installation: tensorboard 2.12.2
  Uninstalling tensorboard-2.12.2:
    Successfully uninstalled tensorboard-2.12.2
Attempting uninstall: tensorflow
  Found existing installation: tensorflow 2.12.0
  Uninstalling tensorflow-2.12.0:
    Successfully uninstalled tensorflow-2.12.0
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
numba 0.56.4 requires numpy<1.24,>=1.18, but you have numpy 1.24.3 which is incompatible.
Successfully installed apache-beam-2.48.0 avro-python3-1.10.2 colorama-0.4.6 crcmod-1.7 dill-0.3.1.1 dnspython-2.3.0 docopt-0.6.2 fastavro-1.7.4 fasteners-0.18 google-auth-oauthlib-0.4.6 hdf
WARNING: The following packages were previously imported in this runtime:
  [numpy,pyparsing]
You must restart the runtime in order to use newly installed versions.
```

RESTART RUNTIME

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting tensorflow==2.8.0
  Downloading tensorflow-2.8.0-cp310-cp310-manylinux2010_x86_64.whl (497.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 497.6/497.6 MB 3.3 MB/s eta 0:00:00
Requirement already satisfied: absl-py>=0.4.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.8.0) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.8.0) (1.6.3)
Requirement already satisfied: flatbuffers>=1.12 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.8.0) (23.3.3)
Requirement already satisfied: gast>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.8.0) (0.4.0)
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.8.0) (0.2.0)
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.8.0) (3.8.0)
Requirement already satisfied: keras-preprocessing>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.8.0) (1.1.2)
Requirement already satisfied: libclang>=9.0.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.8.0) (16.0.0)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.8.0) (1.24.3)
```

```
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.8.0) (3.3.0)
Requirement already satisfied: protobuf>=3.9.2 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.8.0) (3.20.3)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.8.0) (67.7.2)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.8.0) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.8.0) (2.3.0)
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.8.0) (4.5.0)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.8.0) (1.14.1)
Requirement already satisfied: tensorboard<2.9,>=2.8 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.8.0) (2.8.0)
Collecting tf-estimator-nightly==2.8.0.dev2021122109 (from tensorflow==2.8.0)
  Downloading tf_estimator_nightly-2.8.0.dev2021122109-py2.py3-none-any.whl (462 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 462.5/462.5 kB 43.7 MB/s eta 0:00:00
Requirement already satisfied: keras<2.9,>=2.8.0rc0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.8.0) (2.8.0)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.8.0) (0.32.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.8.0) (1.54.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.10/dist-packages (from astunparse>=1.6.0->tensorflow==2.8.0) (0.40.0)
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.9,>=2.8->tensorflow==2.8.0) (2.17.3)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.9,>=2.8->tensorflow==2.8.0) (0.4.6)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.9,>=2.8->tensorflow==2.8.0) (3.4.3)
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.9,>=2.8->tensorflow==2.8.0) (2.27.1)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.9,>=2.8->tensorflow==2.8.0) (0.6.1)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.9,>=2.8->tensorflow==2.8.0) (1.8.1)
Requirement already satisfied: werkzeug>=0.11.15 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.9,>=2.8->tensorflow==2.8.0) (2.3.0)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3->tensorboard<2.9,>=2.8->tensorflow==2.8.0) (5.3.0)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3->tensorboard<2.9,>=2.8->tensorflow==2.8.0) (0.3.0)
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3->tensorboard<2.9,>=2.8->tensorflow==2.8.0) (4.9)
```

You may get warnings or errors related to package dependencies in the previous code block, but you can ignore them for now.

Let's test our installation by running `model_builder_tf2_test.py` to make sure everything is working as expected. Run the following code block and confirm that it finishes without errors. If you get errors, try Googling them or checking the FAQ at the end of this Colab.

```
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.10/dist-packages (from werkzeug>=0.11.15->tensorboard<2.9,>=2.8->tensorflow==2.8.0) (2.1.2)
```

```
# Run Model Bulider Test file, just to verify everything's working properly
!python /content/models/research/object_detection/builders/model_builder_tf2_test.py
```

```
I0616 03:03:51.075860 139722247518016 test_util.py:2373] time(__main__.ModelBuilderTF2Test.test_create_ssd_models_from_config): 25.74s
[       OK ] ModelBuilderTF2Test.test_create_ssd_models_from_config
[ RUN      ] ModelBuilderTF2Test.test_invalid_faster_rcnn_batchnorm_update
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_invalid_faster_rcnn_batchnorm_update): 0.01s
I0616 03:03:51.094336 139722247518016 test_util.py:2373] time(__main__.ModelBuilderTF2Test.test_invalid_faster_rcnn_batchnorm_update): 0.01s
[       OK ] ModelBuilderTF2Test.test_invalid_faster_rcnn_batchnorm_update
[ RUN      ] ModelBuilderTF2Test.test_invalid_first_stage_nms_iou_threshold
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_invalid_first_stage_nms_iou_threshold): 0.0s
I0616 03:03:51.096648 139722247518016 test_util.py:2373] time(__main__.ModelBuilderTF2Test.test_invalid_first_stage_nms_iou_threshold): 0.0s
[       OK ] ModelBuilderTF2Test.test_invalid_first_stage_nms_iou_threshold
[ RUN      ] ModelBuilderTF2Test.test_invalid_model_config_proto
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_invalid_model_config_proto): 0.0s
I0616 03:03:51.097318 139722247518016 test_util.py:2373] time(__main__.ModelBuilderTF2Test.test_invalid_model_config_proto): 0.0s
[       OK ] ModelBuilderTF2Test.test_invalid_model_config_proto
[ RUN      ] ModelBuilderTF2Test.test_invalid_second_stage_batch_size
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_invalid_second_stage_batch_size): 0.0s
I0616 03:03:51.099380 139722247518016 test_util.py:2373] time(__main__.ModelBuilderTF2Test.test_invalid_second_stage_batch_size): 0.0s
[       OK ] ModelBuilderTF2Test.test_invalid_second_stage_batch_size
[ RUN      ] ModelBuilderTF2Test.test_session
[  SKIPPED ] ModelBuilderTF2Test.test_session
[ RUN      ] ModelBuilderTF2Test.test_unknown_faster_rcnn_feature_extractor
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_unknown_faster_rcnn_feature_extractor): 0.0s
I0616 03:03:51.101295 139722247518016 test_util.py:2373] time(__main__.ModelBuilderTF2Test.test_unknown_faster_rcnn_feature_extractor): 0.0s
[       OK ] ModelBuilderTF2Test.test_unknown_faster_rcnn_feature_extractor
[ RUN      ] ModelBuilderTF2Test.test_unknown_meta_architecture
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_unknown_meta_architecture): 0.0s
I0616 03:03:51.101825 139722247518016 test_util.py:2373] time(__main__.ModelBuilderTF2Test.test_unknown_meta_architecture): 0.0s
[       OK ] ModelBuilderTF2Test.test_unknown_meta_architecture
[ RUN      ] ModelBuilderTF2Test.test_unknown_ssd_feature_extractor
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_unknown_ssd_feature_extractor): 0.0s
I0616 03:03:51.103201 139722247518016 test_util.py:2373] time(__main__.ModelBuilderTF2Test.test_unknown_ssd_feature_extractor): 0.0s
[       OK ] ModelBuilderTF2Test.test_unknown_ssd_feature_extractor
----------------------------------------------------------------------
Ran 24 tests in 30.539s

OK (skipped=1)
```

3. Upload Image Dataset and Prepare Training Data

In this section, we'll upload our data and prepare it for training with TensorFlow. We'll upload our images, split them into train, validation, and test folders, and then run scripts for creating TFRecords from our data.

First, on your local PC, zip all your training images and XML files into a single folder called "images.zip". The files should be directly inside the zip folder, or in a nested folder as shown below:

```
images.zip
-- images
  -- img1.jpg
```

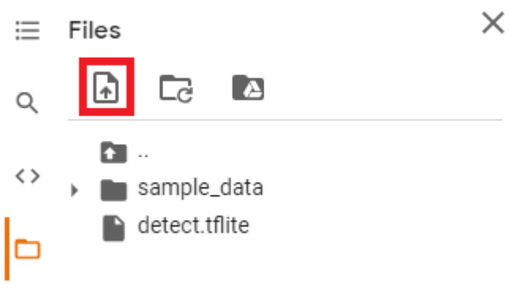
```
-- img1.xml
-- img2.jpg
-- img2.xml
...
```

3.1 Upload images

There are three options for moving the image files to this Colab instance.

Option 1. Upload through Google Colab

Upload the "images.zip" file to the Google Colab instance by clicking the "Files" icon on the left hand side of the browser, and then the "Upload to session storage" icon. Select the zip folder to upload it.



Option 2. Copy from Google Drive

You can also upload your images to your personal Google Drive, mount the drive on this Colab session, and copy them over to the Colab filesystem. This option works well if you want to upload the images beforehand so you don't have to wait for them to upload each time you restart this Colab. If you have more than 50MB worth of images, I recommend using this option.

First, upload the "images.zip" file to your Google Drive, and make note of the folder you uploaded them to. Replace `MyDrive/path/to/images.zip` with the path to your zip file. (For example, I uploaded the zip file to folder called "change-counter1", so I would use `MyDrive/change-counter1/images.zip` for the path). Then, run the following block of code to mount your Google Drive to this Colab session and copy the folder to this filesystem.

```
from google.colab import drive
drive.mount('/content/gdrive')

!cp /content/gdrive/MyDrive/datasets/menu/images.zip /content
```

```
Mounted at /content/gdrive
```

Option 3. Use coin detection dataset

If you don't have a dataset and just want to try training a model, you can download my coin image dataset to use as an example. I've uploaded a dataset containing 750 labeled images of pennies, nickels, dimes, and quarters. Run the following code block to download the dataset.

```
!wget -O /content/images.zip https://www.dropbox.com/s/gk57ec3v8dfuwcp/CoinPics_11NOV22.zip?dl=0  # United States coin images
```

3.2 Split images into train, validation, and test folders

At this point, whether you used Option 1, 2, or 3, you should be able to click the folder icon on the left and see your "images.zip" file in the list of files. Now that the dataset is uploaded, let's unzip it and create some folders to hold the images. These directories are created in the /content folder in this instance's filesystem. You can browse the filesystem by clicking the "Files" icon on the left.

```
!mkdir /content/images
!unzip -q images.zip -d /content/images/all
!mkdir /content/images/train; mkdir /content/images/validation; mkdir /content/images/test
```

Next, we'll split the images into train, validation, and test sets. Here's what each set is used for:

- **Train**: These are the actual images used to train the model. In each step of training, a batch of images from the "train" set is passed into the neural network. The network predicts classes and locations of objects in the images. The training algorithm calculates the loss (i.e. how "wrong" the predictions were) and adjusts the network weights through backpropagation.
- **Validation**: Images from the "validation" set can be used by the training algorithm to check the progress of training and adjust hyperparameters (like learning rate). Unlike "train" images, these images are only used periodically during training (i.e. once every certain number of training steps).
- **Test**: These images are never seen by the neural network during training. They are intended to be used by a human to perform final testing of the model to check how accurate the model is.

I wrote a Python script to randomly move 80% of the images to the "train" folder, 10% to the "validation" folder, and 10% to the "test" folder. Click play on the following block to download the script and execute it.

```
!wget https://raw.githubusercontent.com/EdjeElectronics/TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi/master/util_scripts/train_val_test_spl
!python train_val_test_split.py
```

```
--2023-06-16 03:05:32--  https://raw.githubusercontent.com/EdjeElectronics/TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi/master/util_scripts/train_val_test_split.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.110.133, 185.199.108.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2803 (2.7K) [text/plain]
Saving to: ‘train_val_test_split.py’

train_val_test_spli 100%[===================>]   2.74K  --.-KB/s    in 0s

2023-06-16 03:05:33 (38.6 MB/s) - ‘train_val_test_split.py’ saved [2803/2803]
```

```
Total images: 60
Images moving to train: 48
Images moving to validation: 6
Images moving to test: 6
```

3.3 Create Labelmap and TFRecords

Finally, we need to create a labelmap for the detector and convert the images into a data file format called TFRecords, which are used by TensorFlow for training. We'll use Python scripts to automatically convert the data into TFRecord format. Before running them, we need to define a labelmap for our classes.

The code section below will create a "labelmap.txt" file that contains a list of classes. Replace the `class1`, `class2`, `class3` text with your own classes (for example, `penny`, `nickel`, `dime`, `quarter`), adding a new line for each class. Then, click play to execute the code.

```
### This creates a a "labelmap.txt" file with a list of classes the object detection model will detect.
%%bash
cat <<EOF >> /content/labelmap.txt
butter milk - 10
grape - 40
lime - 20
musk melon - 30
water melon - 20
orange - 70
pineapple - 60
musambi - 60
pista milk - 20
badam milkshake - 50
pineapple drink - 20
nutty bar - 30
butterscotch cone - 35
black current cone - 40
black forest cake - 50
idly(4) - 32
wheat upma - 25
chapathi egg masala - 93
fish Meals - 75
mathi fry - 50
katla fry - 60
single omelet - 15
double omelet - 30
egg briyani half -50
egg briyani - 80
parotta - 90
veg puff - 15
badhusa -15
dill puff - 15
bread toast - 15
cookies - 5
```

```
cutlet - 20
Tea - 10
coffee - 15
boost - 35
horlicks - 35
bournvita - 35
sukku coffee - 10
black tea - 10
EOF
```

Download and run the data conversion scripts from the GitHub repository by clicking play on the following three sections of code. They will create TFRecord files for the train and validation datasets, as well as a `labelmap.pbtxt` file which contains the labelmap in a different format.

```
# Download data conversion scripts
! wget https://raw.githubusercontent.com/EdjeElectronics/TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi/master/util_scripts/create_csv.py
! wget https://raw.githubusercontent.com/EdjeElectronics/TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi/master/util_scripts/create_tfrecord.p
```

```
--2023-06-16 03:06:11--  https://raw.githubusercontent.com/EdjeElectronics/TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi/master/util_scripts/create_csv.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1348 (1.3K) [text/plain]
Saving to: ‘create_csv.py’

create_csv.py       100%[===================>]   1.32K  --.-KB/s    in 0s

2023-06-16 03:06:11 (75.2 MB/s) - ‘create_csv.py’ saved [1348/1348]

--2023-06-16 03:06:11--  https://raw.githubusercontent.com/EdjeElectronics/TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi/master/util_scripts/create_tfrecord.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4414 (4.3K) [text/plain]
Saving to: ‘create_tfrecord.py’

create_tfrecord.py  100%[===================>]   4.31K  --.-KB/s    in 0s

2023-06-16 03:06:11 (64.5 MB/s) - ‘create_tfrecord.py’ saved [4414/4414]
```

```
# Create CSV data files and TFRecord files
!python3 create_csv.py
!python3 create_tfrecord.py --csv_input=images/train_labels.csv --labelmap=labelmap.txt --image_dir=images/train --output_path=train.tfrecord
!python3 create_tfrecord.py --csv_input=images/validation_labels.csv --labelmap=labelmap.txt --image_dir=images/validation --output_path=val.tfrecord
```

```
Successfully converted xml to csv.
Successfully converted xml to csv.
Successfully created the TFRecords: /content/train.tfrecord
Successfully created the TFRecords: /content/val.tfrecord
```

We'll store the locations of the TFRecord and labelmap files as variables so we can reference them later in this Colab session.

```
train_record_fname = '/content/train.tfrecord'
val_record_fname = '/content/val.tfrecord'
label_map_pbtxt_fname = '/content/labelmap.pbtxt'
```

4. Set Up Training Configuration

In this section, we'll set up the model and training configuration. We'll specifiy which pretrained TensorFlow model we want to use from the TensorFlow 2 Object Detection Model Zoo. Each model also comes with a configuration file that points to file locations, sets training parameters (such as learning rate and total number of training steps), and more. We'll modify the configuration file for our custom training job.

The first section of code lists out some models availabe in the TF2 Model Zoo and defines some filenames that will be used later to download the model and config file. This makes it easy to manage which model you're using and to add other models to the list later.

Set the "chosen_model" variable to match the name of the model you'd like to train with. It's currently set to use the popular "ssd-mobilenet-v2" model. Click play on the next block once the chosen model has been set.

Not sure which model to pick? Check out my blog post comparing each model's speed and accuracy.

```
# Change the chosen_model variable to deploy different models available in the TF2 object detection zoo
chosen_model = 'ssd-mobilenet-v2-fpnlite-320'

MODELS_CONFIG = {
    'ssd-mobilenet-v2': {
        'model_name': 'ssd_mobilenet_v2_320x320_coco17_tpu-8',
        'base_pipeline_file': 'ssd_mobilenet_v2_320x320_coco17_tpu-8.config',
        'pretrained_checkpoint': 'ssd_mobilenet_v2_320x320_coco17_tpu-8.tar.gz',
    },
    'efficientdet-d0': {
        'model_name': 'efficientdet_d0_coco17_tpu-32',
        'base_pipeline_file': 'ssd_efficientdet_d0_512x512_coco17_tpu-8.config',
        'pretrained_checkpoint': 'efficientdet_d0_coco17_tpu-32.tar.gz',
    },
    'ssd-mobilenet-v2-fpnlite-320': {
        'model_name': 'ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8',
        'base_pipeline_file': 'ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8.config',
        'pretrained_checkpoint': 'ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8.tar.gz',
    },
    # The centernet model isn't working as of 9/10/22
    #'centernet-mobilenet-v2': {
    #    'model_name': 'centernet_mobilenetv2fpn_512x512_coco17_od',
    #    'base_pipeline_file': 'pipeline.config',
    #    'pretrained_checkpoint': 'centernet_mobilenetv2fpn_512x512_coco17_od.tar.gz',
    #}
```

```
}

model_name = MODELS_CONFIG[chosen_model]['model_name']
pretrained_checkpoint = MODELS_CONFIG[chosen_model]['pretrained_checkpoint']
base_pipeline_file = MODELS_CONFIG[chosen_model]['base_pipeline_file']
```

Download the pretrained model file and configuration file by clicking Play on the following section.

```
# Create "mymodel" folder for holding pre-trained weights and configuration files
%mkdir /content/models/mymodel/
%cd /content/models/mymodel/

# Download pre-trained model weights
import tarfile
download_tar = 'http://download.tensorflow.org/models/object_detection/tf2/20200711/' + pretrained_checkpoint
!wget {download_tar}
tar = tarfile.open(pretrained_checkpoint)
tar.extractall()
tar.close()

# Download training configuration file for model
download_config = 'https://raw.githubusercontent.com/tensorflow/models/master/research/object_detection/configs/tf2/' + base_pipeline_file
!wget {download_config}
```

```
/content/models/mymodel
--2023-06-16 03:06:44--  http://download.tensorflow.org/models/object_detection/tf2/20200711/ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8.tar.gz
Resolving download.tensorflow.org (download.tensorflow.org)... 108.177.127.128, 2a00:1450:4013:c07::80
Connecting to download.tensorflow.org (download.tensorflow.org)|108.177.127.128|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 20515344 (20M) [application/x-tar]
Saving to: ‘ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8.tar.gz’

ssd_mobilenet_v2_fp 100%[===================>]  19.56M  77.8MB/s    in 0.3s

2023-06-16 03:06:45 (77.8 MB/s) - ‘ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8.tar.gz’ saved [20515344/20515344]

--2023-06-16 03:06:45--  https://raw.githubusercontent.com/tensorflow/models/master/research/object_detection/configs/tf2/ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8.config
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4684 (4.6K) [text/plain]
Saving to: ‘ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8.config’

ssd_mobilenet_v2_fp 100%[===================>]   4.57K  --.-KB/s    in 0s

2023-06-16 03:06:45 (35.7 MB/s) - ‘ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8.config’ saved [4684/4684]
```

Now that we've downloaded our model and config file, we need to modify the configuration file with some high-level training parameters. The following variables are used to control training steps:

- **num_steps**: The total amount of steps to use for training the model. A good number to start with is 40,000 steps. You can use more steps if you notice the loss metrics are still decreasing by the time training finishes. The more steps, the longer training will take. Training can also be stopped early if loss flattens out before reaching the specified number of steps.
- **batch_size**: The number of images to use per training step. A larger batch size allows a model to be trained in fewer steps, but the size is limited by the GPU memory available for training. With the GPUs used in Colab instances, 16 is a good number for SSD models and 4 is good for EfficientDet models.

Other training information, like the location of the pretrained model file, the config file, and total number of classes are also assigned in this step. To learn more about training configuration with the TensorFlow Object Detection API, read this article from Neptune.

```
# Set training parameters for the model
num_steps = 5000

if chosen_model == 'efficientdet-d0':
  batch_size = 4
else:
  batch_size = 8
```

```
# Set file locations and get number of classes for config file
pipeline_fname = '/content/models/mymodel/' + base_pipeline_file
fine_tune_checkpoint = '/content/models/mymodel/' + model_name + '/checkpoint/ckpt-0'

def get_num_classes(pbtxt_fname):
    from object_detection.utils import label_map_util
    label_map = label_map_util.load_labelmap(pbtxt_fname)
    categories = label_map_util.convert_label_map_to_categories(
        label_map, max_num_classes=90, use_display_name=True)
    category_index = label_map_util.create_category_index(categories)
    return len(category_index.keys())
num_classes = get_num_classes(label_map_pbtxt_fname)
print('Total classes:', num_classes)
```

```
Total classes: 39
```

Next, we'll rewrite the config file to use the training parameters we just specified. The following section of code will automatically replace the necessary parameters in the downloaded .config file and save it as our custom "pipeline_file.config" file.

```
# Create custom configuration file by writing the dataset, model checkpoint, and training parameters into the base pipeline file
import re

%cd /content/models/mymodel
print('writing custom configuration file')

with open(pipeline_fname) as f:
    s = f.read()
```

```
with open('pipeline_file.config', 'w') as f:

    # Set fine_tune_checkpoint path
    s = re.sub('fine_tune_checkpoint: ".*?"',
               'fine_tune_checkpoint: "{}"'.format(fine_tune_checkpoint), s)

    # Set tfrecord files for train and test datasets
    s = re.sub(
        '(input_path: ".*?)(PATH_TO_BE_CONFIGURED/train)(.*?")', 'input_path: "{}"'.format(train_record_fname), s)
    s = re.sub(
        '(input_path: ".*?)(PATH_TO_BE_CONFIGURED/val)(.*?")', 'input_path: "{}"'.format(val_record_fname), s)

    # Set label_map_path
    s = re.sub(
        'label_map_path: ".*?"', 'label_map_path: "{}"'.format(label_map_pbtxt_fname), s)

    # Set batch_size
    s = re.sub('batch_size: [0-9]+',
               'batch_size: {}'.format(batch_size), s)

    # Set training steps, num_steps
    s = re.sub('num_steps: [0-9]+',
               'num_steps: {}'.format(num_steps), s)

    # Set number of classes num_classes
    s = re.sub('num_classes: [0-9]+',
               'num_classes: {}'.format(num_classes), s)

    # Change fine-tune checkpoint type from "classification" to "detection"
    s = re.sub(
        'fine_tune_checkpoint_type: "classification"', 'fine_tune_checkpoint_type: "{}"'.format('detection'), s)

    # If using ssd-mobilenet-v2, reduce learning rate (because it's too high in the default config file)
    if chosen_model == 'ssd-mobilenet-v2':
      s = re.sub('learning_rate_base: .8',
                 'learning_rate_base: .08', s)

      s = re.sub('warmup_learning_rate: 0.13333',
                 'warmup_learning_rate: .026666', s)

    # If using efficientdet-d0, use fixed_shape_resizer instead of keep_aspect_ratio_resizer (because it isn't supported by TFLite)
    if chosen_model == 'efficientdet-d0':
      s = re.sub('keep_aspect_ratio_resizer', 'fixed_shape_resizer', s)
      s = re.sub('pad_to_max_dimension: true', '', s)
      s = re.sub('min_dimension', 'height', s)
      s = re.sub('max_dimension', 'width', s)

    f.write(s)
```

```
/content/models/mymodel
writing custom configuration file
```

(Optional) If you're curious, you can display the configuration file's contents here in the browser by running the line of code below.

```
# (Optional) Display the custom configuration file's contents
!cat /content/models/mymodel/pipeline_file.config
```

```
    metrics_set: "coco_detection_metrics"
    use_moving_averages: false
  }

  eval_input_reader: {
    label_map_path: "/content/labelmap.pbtxt"
    shuffle: false
    num_epochs: 1
    tf_record_input_reader {
      input_path: "/content/val.tfrecord"
    }
  }
```

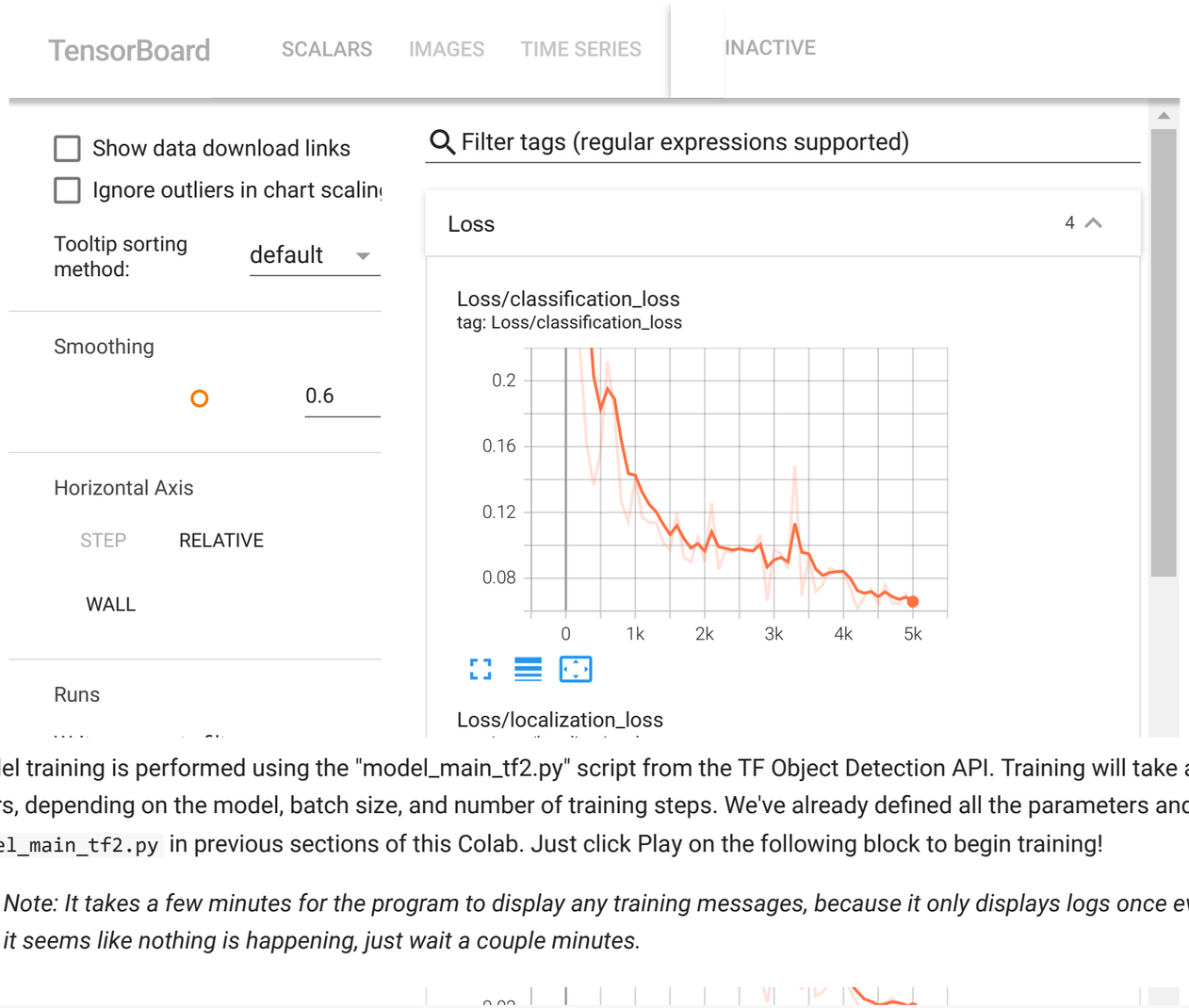
Finally, let's set the locations of the configuration file and model output directory as variables so we can reference them when we call the training command.

```
# Set the path to the custom config file and the directory to store training checkpoints in
pipeline_file = '/content/models/mymodel/pipeline_file.config'
model_dir = '/content/training/'
```

5. Train Custom TFLite Detection Model

We're ready to train our object detection model! Before we start training, let's load up a TensorBoard session to monitor training progress. Run the following section of code, and a TensorBoard session will appear in the browser. It won't show anything yet, because we haven't started training. Once training starts, come back and click the refresh button to see the model's overall loss.

```
%load_ext tensorboard
%tensorboard --logdir '/content/training/train'
```



Model training is performed using the "model_main_tf2.py" script from the TF Object Detection API. Training will take anywhere from 2 to 6 hours, depending on the model, batch size, and number of training steps. We've already defined all the parameters and arguments used by `model_main_tf2.py` in previous sections of this Colab. Just click Play on the following block to begin training!

> *Note: It takes a few minutes for the program to display any training messages, because it only displays logs once every 100 steps. If it seems like nothing is happening, just wait a couple minutes.*

```
# Run training!
!python /content/models/research/object_detection/model_main_tf2.py \
    --pipeline_config_path={pipeline_file} \
    --model_dir={model_dir} \
    --alsologtostderr \
    --num_train_steps={num_steps} \
    --sample_1_of_n_eval_examples=1
```

```
 'learning_rate': 0.078939244}
INFO:tensorflow:Step 4700 per-step time 0.210s
I0616 03:26:53.910124 140296422004544 model_lib_v2.py:705] Step 4700 per-step time 0.210s
INFO:tensorflow:{'Loss/classification_loss': 0.06478383,
 'Loss/localization_loss': 0.024063656,
 'Loss/regularization_loss': 0.122948974,
 'Loss/total_loss': 0.21179646,
 'learning_rate': 0.07887978}
I0616 03:26:53.910469 140296422004544 model_lib_v2.py:708] {'Loss/classification_loss': 0.06478383,
 'Loss/localization_loss': 0.024063656,
 'Loss/regularization_loss': 0.122948974,
 'Loss/total_loss': 0.21179646,
 'learning_rate': 0.07887978}
INFO:tensorflow:Step 4800 per-step time 0.209s
I0616 03:27:14.841148 140296422004544 model_lib_v2.py:705] Step 4800 per-step time 0.209s
INFO:tensorflow:{'Loss/classification_loss': 0.06462867,
 'Loss/localization_loss': 0.021625433,
 'Loss/regularization_loss': 0.122347824,
 'Loss/total_loss': 0.20860192,
 'learning_rate': 0.07881871}
I0616 03:27:14.848169 140296422004544 model_lib_v2.py:708] {'Loss/classification_loss': 0.06462867,
 'Loss/localization_loss': 0.021625433,
 'Loss/regularization_loss': 0.122347824,
 'Loss/total_loss': 0.20860192,
 'learning_rate': 0.07881871}
INFO:tensorflow:Step 4900 per-step time 0.203s
I0616 03:27:35.162303 140296422004544 model_lib_v2.py:705] Step 4900 per-step time 0.203s
INFO:tensorflow:{'Loss/classification_loss': 0.070447505,
 'Loss/localization_loss': 0.019805253,
 'Loss/regularization_loss': 0.12172723,
 'Loss/total_loss': 0.21197999,
 'learning_rate': 0.07875605}
I0616 03:27:35.162708 140296422004544 model_lib_v2.py:708] {'Loss/classification_loss': 0.070447505,
 'Loss/localization_loss': 0.019805253,
 'Loss/regularization_loss': 0.12172723,
 'Loss/total_loss': 0.21197999,
 'learning_rate': 0.07875605}
INFO:tensorflow:Step 5000 per-step time 0.204s
I0616 03:27:55.520476 140296422004544 model_lib_v2.py:705] Step 5000 per-step time 0.204s
INFO:tensorflow:{'Loss/classification_loss': 0.061745405,
 'Loss/localization_loss': 0.017726487,
 'Loss/regularization_loss': 0.12110983,
 'Loss/total_loss': 0.20058173,
 'learning_rate': 0.078691795}
I0616 03:27:55.524968 140296422004544 model_lib_v2.py:708] {'Loss/classification_loss': 0.061745405,
 'Loss/localization_loss': 0.017726487,
 'Loss/regularization_loss': 0.12110983,
 'Loss/total_loss': 0.20058173,
 'learning_rate': 0.078691795}
```

If you want to stop training early, just click Stop a couple times or right-click on the code block and select "Interrupt Execution". Otherwise, training will stop by itself once it reaches the specified number of training steps.

6. Convert Model to TensorFlow Lite

Alright! Our model is all trained up and ready to be used for detecting objects. First, we need to export the model graph (a file that contains information about the architecture and weights) to a TensorFlow Lite-compatible format. We'll do this using the `export_tflite_graph_tf2.py` script.

```
# Make a directory to store the trained TFLite model
!mkdir /content/custom_model_lite
output_directory = '/content/custom_model_lite'

# Path to training directory (the conversion script automatically chooses the highest checkpoint file)
last_model_path = '/content/training'

!python /content/models/research/object_detection/export_tflite_graph_tf2.py \
    --trained_checkpoint_dir {last_model_path} \
    --output_directory {output_directory} \
    --pipeline_config_path {pipeline_file}
```

```
W0616 03:30:07.339544 139882673456960 save_impl.py:71] Skipping full serialization of Keras layer <object_detection.core.freezable_batch_norm.FreezableBatchNorm object at 0x7f38003493f0>, be
WARNING:tensorflow:Skipping full serialization of Keras layer <keras.layers.core.lambda_layer.Lambda object at 0x7f380034ba90>, because it is not built.
W0616 03:30:07.339623 139882673456960 save_impl.py:71] Skipping full serialization of Keras layer <keras.layers.core.lambda_layer.Lambda object at 0x7f380034ba90>, because it is not built.
WARNING:tensorflow:Skipping full serialization of Keras layer <object_detection.core.freezable_batch_norm.FreezableBatchNorm object at 0x7f380034a6b0>, because it is not built.
W0616 03:30:07.339696 139882673456960 save_impl.py:71] Skipping full serialization of Keras layer <object_detection.core.freezable_batch_norm.FreezableBatchNorm object at 0x7f380034a6b0>, be
WARNING:tensorflow:Skipping full serialization of Keras layer <keras.layers.core.lambda_layer.Lambda object at 0x7f380034b8b0>, because it is not built.
W0616 03:30:07.339780 139882673456960 save_impl.py:71] Skipping full serialization of Keras layer <keras.layers.core.lambda_layer.Lambda object at 0x7f380034b8b0>, because it is not built.
WARNING:tensorflow:Skipping full serialization of Keras layer <object_detection.core.freezable_batch_norm.FreezableBatchNorm object at 0x7f38003499c0>, because it is not built.
W0616 03:30:07.339867 139882673456960 save_impl.py:71] Skipping full serialization of Keras layer <object_detection.core.freezable_batch_norm.FreezableBatchNorm object at 0x7f38003499c0>, be
WARNING:tensorflow:Skipping full serialization of Keras layer <keras.layers.core.lambda_layer.Lambda object at 0x7f3800349d80>, because it is not built.
W0616 03:30:07.339942 139882673456960 save_impl.py:71] Skipping full serialization of Keras layer <keras.layers.core.lambda_layer.Lambda object at 0x7f3800349d80>, because it is not built.
WARNING:tensorflow:Skipping full serialization of Keras layer <object_detection.core.freezable_batch_norm.FreezableBatchNorm object at 0x7f38001e4df0>, because it is not built.
W0616 03:30:07.340014 139882673456960 save_impl.py:71] Skipping full serialization of Keras layer <object_detection.core.freezable_batch_norm.FreezableBatchNorm object at 0x7f38001e4df0>, be
WARNING:tensorflow:Skipping full serialization of Keras layer <keras.layers.core.lambda_layer.Lambda object at 0x7f38001e5210>, because it is not built.
W0616 03:30:07.340088 139882673456960 save_impl.py:71] Skipping full serialization of Keras layer <keras.layers.core.lambda_layer.Lambda object at 0x7f38001e5210>, because it is not built.
WARNING:tensorflow:Skipping full serialization of Keras layer <object_detection.core.freezable_batch_norm.FreezableBatchNorm object at 0x7f38001e5960>, because it is not built.
W0616 03:30:07.340162 139882673456960 save_impl.py:71] Skipping full serialization of Keras layer <object_detection.core.freezable_batch_norm.FreezableBatchNorm object at 0x7f38001e5960>, be
WARNING:tensorflow:Skipping full serialization of Keras layer <keras.layers.core.lambda_layer.Lambda object at 0x7f38001e48e0>, because it is not built.
W0616 03:30:07.340235 139882673456960 save_impl.py:71] Skipping full serialization of Keras layer <keras.layers.core.lambda_layer.Lambda object at 0x7f38001e48e0>, because it is not built.
WARNING:tensorflow:Skipping full serialization of Keras layer <object_detection.core.freezable_batch_norm.FreezableBatchNorm object at 0x7f38001e53f0>, because it is not built.
W0616 03:30:07.340309 139882673456960 save_impl.py:71] Skipping full serialization of Keras layer <object_detection.core.freezable_batch_norm.FreezableBatchNorm object at 0x7f38001e53f0>, be
WARNING:tensorflow:Skipping full serialization of Keras layer <keras.layers.core.lambda_layer.Lambda object at 0x7f38001e5270>, because it is not built.
W0616 03:30:07.387684 139882673456960 save_impl.py:71] Skipping full serialization of Keras layer <keras.layers.core.lambda_layer.Lambda object at 0x7f38001e5270>, because it is not built.
W0616 03:30:27.921396 139882673456960 save.py:260] Found untraced functions such as WeightSharedConvolutionalBoxPredictor_layer_call_fn, WeightSharedConvolutionalBoxPredictor_layer_call_and_
INFO:tensorflow:Assets written to: /content/custom_model_lite/saved_model/assets
I0616 03:30:32.606770 139882673456960 builder_impl.py:779] Assets written to: /content/custom_model_lite/saved_model/assets
```

Next, we'll take the exported graph and use the `TFLiteConverter` module to convert it to `.tflite` FlatBuffer format.

```
# Convert exported graph file into TFLite model file
import tensorflow as tf

converter = tf.lite.TFLiteConverter.from_saved_model('/content/custom_model_lite/saved_model')
tflite_model = converter.convert()

with open('/content/custom_model_lite/detect.tflite', 'wb') as f:
  f.write(tflite_model)
```

7. Test TensorFlow Lite Model and Calculate mAP

We've trained our custom model and converted it to TFLite format. But how well does it actually perform at detecting objects in images? This is where the images we set aside in the **test** folder come in. The model never saw any test images during training, so its performance on these images should be representative of how it will perform on new images from the field.

7.1 Inference test images

The following code defines a function to run inference on test images. It loads the images, loads the model and labelmap, runs the model on each image, and displays the result. It also optionally saves detection results as text files so we can use them to calculate model mAP score.

This code is based off the TFLite_detection_image.py script from my TensorFlow Lite Object Detection repository on GitHub; feel free to use it as a starting point for your own application.

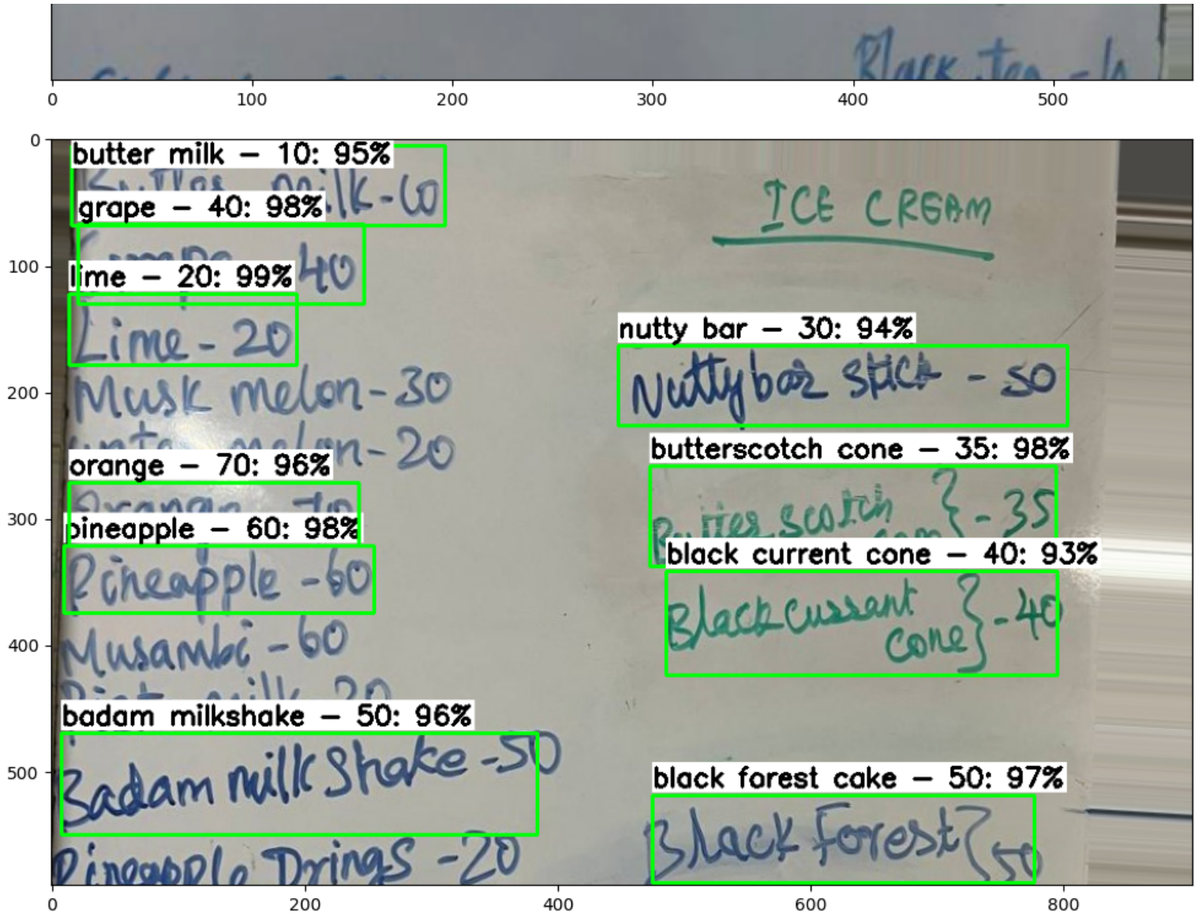
```
# Script to run custom TFLite model on test images to detect objects
# Source: https://github.com/EdjeElectronics/TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi/blob/master/TFLite_detection_image.py

# Import packages
import os
import cv2
import numpy as np
import sys
import glob
import random
import importlib.util
from tensorflow.lite.python.interpreter import Interpreter

import matplotlib
import matplotlib.pyplot as plt

%matplotlib inline

### Define function for inferencing with TFLite model and displaying results

def tflite_detect_images(modelpath, imgpath, lblpath, min_conf=0.5, num_test_images=10, savepath='/content/results', txt_only=False):

  # Grab filenames of all images in test folder
  images = glob.glob(imgpath + '/*.jpg') + glob.glob(imgpath + '/*.JPG') + glob.glob(imgpath + '/*.png') + glob.glob(imgpath + '/*.bmp')

  # Load the label map into memory
  with open(lblpath, 'r') as f:
      labels = [line.strip() for line in f.readlines()]

  # Load the Tensorflow Lite model into memory
  interpreter = Interpreter(model_path=modelpath)
  interpreter.allocate_tensors()

  # Get model details
  input_details = interpreter.get_input_details()
  output_details = interpreter.get_output_details()
  height = input_details[0]['shape'][1]
  width = input_details[0]['shape'][2]

  float_input = (input_details[0]['dtype'] == np.float32)

  input_mean = 127.5
  input_std = 127.5
```

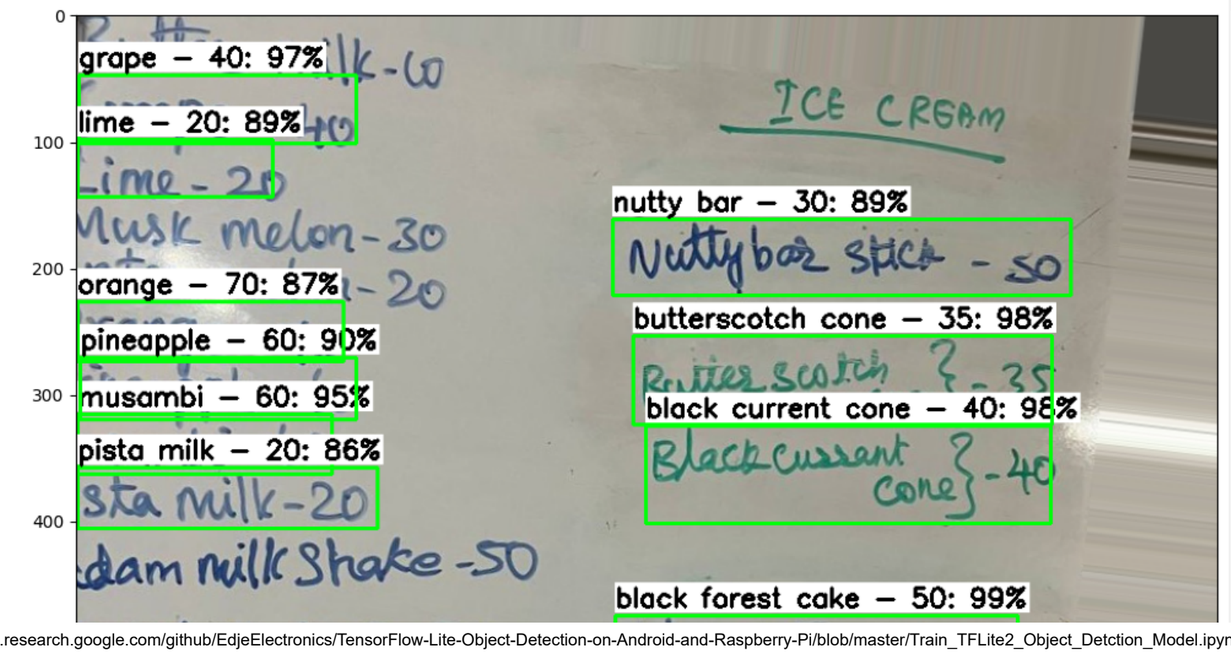
```
# Randomly select test images
images_to_test = random.sample(images, num_test_images)

# Loop over every image and perform detection
for image_path in images_to_test:

    # Load image and resize to expected shape [1xHxWx3]
    image = cv2.imread(image_path)
    image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    imH, imW, _ = image.shape
    image_resized = cv2.resize(image_rgb, (width, height))
    input_data = np.expand_dims(image_resized, axis=0)

    # Normalize pixel values if using a floating model (i.e. if model is non-quantized)
    if float_input:
        input_data = (np.float32(input_data) - input_mean) / input_std

    # Perform the actual detection by running the model with the image as input
    interpreter.set_tensor(input_details[0]['index'],input_data)
    interpreter.invoke()

    # Retrieve detection results
    boxes = interpreter.get_tensor(output_details[1]['index'])[0] # Bounding box coordinates of detected objects
    classes = interpreter.get_tensor(output_details[3]['index'])[0] # Class index of detected objects
    scores = interpreter.get_tensor(output_details[0]['index'])[0] # Confidence of detected objects

    detections = []

    # Loop over all detections and draw detection box if confidence is above minimum threshold
    for i in range(len(scores)):
        if ((scores[i] > min_conf) and (scores[i] <= 1.0)):

            # Get bounding box coordinates and draw box
            # Interpreter can return coordinates that are outside of image dimensions, need to force them to be within image using max() and min()
            ymin = int(max(1,(boxes[i][0] * imH)))
            xmin = int(max(1,(boxes[i][1] * imW)))
            ymax = int(min(imH,(boxes[i][2] * imH)))
            xmax = int(min(imW,(boxes[i][3] * imW)))

            cv2.rectangle(image, (xmin,ymin), (xmax,ymax), (10, 255, 0), 2)

            # Draw label
            object_name = labels[int(classes[i])] # Look up object name from "labels" array using class index
            label = '%s: %d%%' % (object_name, int(scores[i]*100)) # Example: 'person: 72%'
            labelSize, baseLine = cv2.getTextSize(label, cv2.FONT_HERSHEY_SIMPLEX, 0.7, 2) # Get font size
            label_ymin = max(ymin, labelSize[1] + 10) # Make sure not to draw label too close to top of window
            cv2.rectangle(image, (xmin, label_ymin-labelSize[1]-10), (xmin+labelSize[0], label_ymin+baseLine-10), (255, 255, 255), cv2.FILLED) # Draw white box to put label text in
            cv2.putText(image, label, (xmin, label_ymin-7), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 0), 2) # Draw label text

            detections.append([object_name, scores[i], xmin, ymin, xmax, ymax])
```

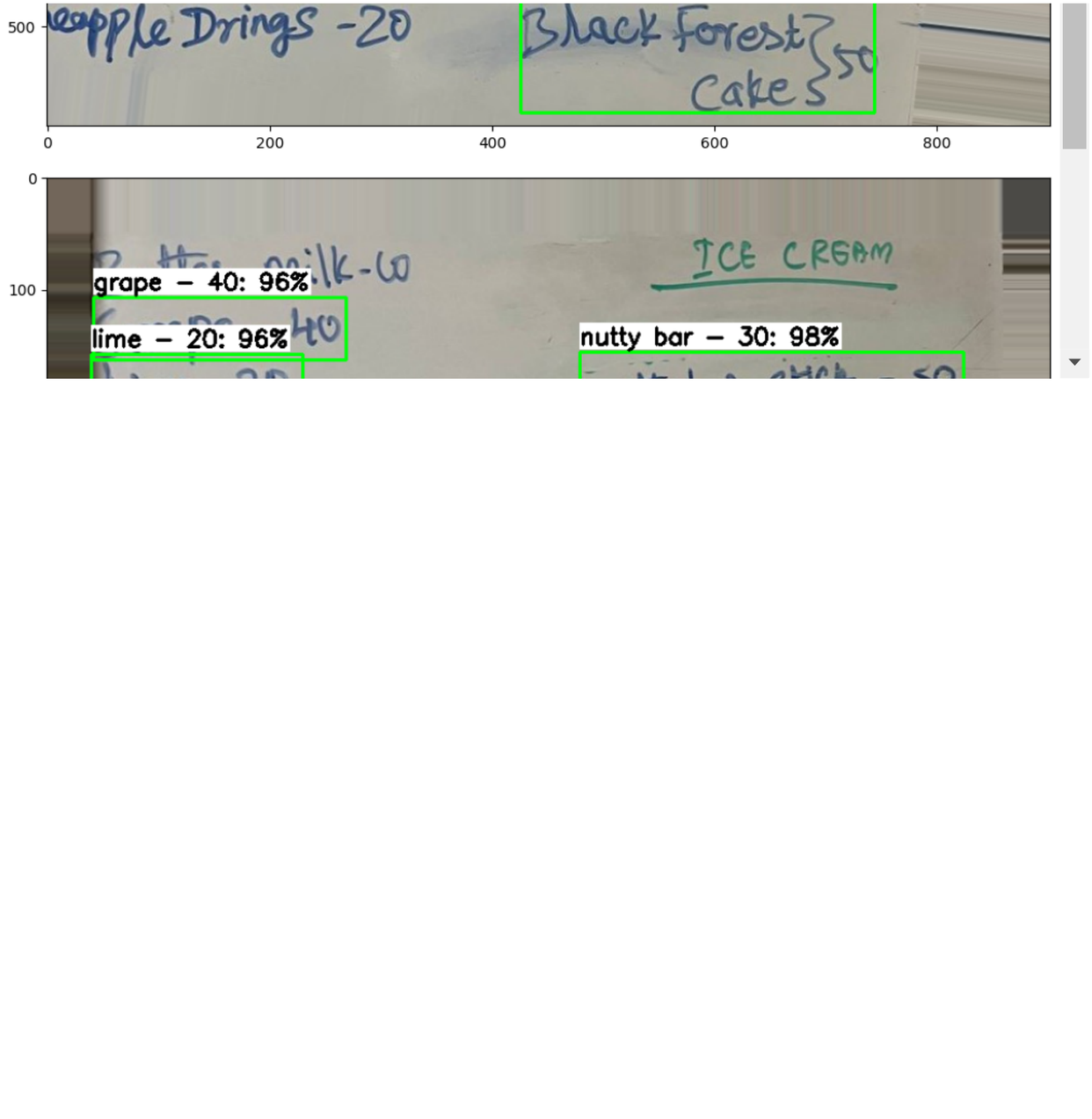
```
    # All the results have been drawn on the image, now display the image
    if txt_only == False: # "text_only" controls whether we want to display the image results or just save them in .txt files
      image = cv2.cvtColor(image,cv2.COLOR_BGR2RGB)
      plt.figure(figsize=(12,16))
      plt.imshow(image)
      plt.show()

    # Save detection results in .txt files (for calculating mAP)
    elif txt_only == True:

      # Get filenames and paths
      image_fn = os.path.basename(image_path)
      base_fn, ext = os.path.splitext(image_fn)
      txt_result_fn = base_fn +'.txt'
      txt_savepath = os.path.join(savepath, txt_result_fn)

      # Write results to text file
      # (Using format defined by https://github.com/Cartucho/mAP, which will make it easy to calculate mAP)
      with open(txt_savepath,'w') as f:
          for detection in detections:
              f.write('%s %.4f %d %d %d %d\n' % (detection[0], detection[1], detection[2], detection[3], detection[4], detection[5]))

return
```

The next block sets the paths to the test images and models and then runs the inferencing function. If you want to use more than 10 images, change the `images_to_test` variable. Click play to run inferencing!

```
# Set up variables for running user's model
PATH_TO_IMAGES='/content/images/test'   # Path to test images folder
PATH_TO_MODEL='/content/custom_model_lite/detect.tflite'   # Path to .tflite model file
PATH_TO_LABELS='/content/labelmap.txt'   # Path to labelmap.txt file
min_conf_threshold=0.5   # Confidence threshold (try changing this to 0.01 if you don't see any detection results)
images_to_test = 5   # Number of images to run detection on

# Run inferencing function!
tflite_detect_images(PATH_TO_MODEL, PATH_TO_IMAGES, PATH_TO_LABELS, min_conf_threshold, images_to_test)
```


butter milk – 10: 95%
grape – 40: 98%
lime – 20: 99%
orange – 70: 96%
pineapple – 60: 98%
badam milkshake – 50: 96%
nutty bar – 30: 94%
butterscotch cone – 35: 98%
black current cone – 40: 93%
black forest cake – 50: 97%
ICE CREAM


grape – 40: 97%
lime – 20: 89%
orange – 70: 87%
pineapple – 60: 90%
musambi – 60: 95%
pista milk – 20: 86%
nutty bar – 30: 89%
butterscotch cone – 35: 98%
black current cone – 40: 98%
black forest cake – 50: 99%
ICE CREAM


500
0
200
400
600
800
0
100
grape – 40: 96%
lime – 20: 96%
nutty bar – 30: 98%
ICE CREAM

7.2 Calculate mAP

Now we have a visual sense of how our model performs on test images, but how can we quantitatively measure its accuracy?

One popular methord for measuring object detection model accuracy is "mean average precision" (mAP). Basically, the higher the mAP score, the better your model is at detecting objects in images. To learn more about mAP, read through this article from Roboflow.

We'll use the mAP calculator tool at https://github.com/Cartucho/mAP to determine our model's mAP score. First, we need to clone the repository and remove its existing example data. We'll also download a script I wrote for interfacing with the calculator.

```
%%bash
git clone https://github.com/Cartucho/mAP /content/mAP
cd /content/mAP
rm input/detection-results/*
rm input/ground-truth/*
rm input/images-optional/*
wget https://raw.githubusercontent.com/EdjeElectronics/TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi/master/util_scripts/calculate_map_cartucho.py
```

```
Cloning into '/content/mAP'...
--2023-06-16 03:40:41--  https://raw.githubusercontent.com/EdjeElectronics/TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi/master/util_scripts/calculate_map_cartucho.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5397 (5.3K) [text/plain]
Saving to: ‘calculate_map_cartucho.py’

     0K .....                                                 100% 52.2M=0s

2023-06-16 03:40:41 (52.2 MB/s) - ‘calculate_map_cartucho.py’ saved [5397/5397]
```

Next, we'll copy the images and annotation data from the **test** folder to the appropriate folders inside the cloned repository. These will be used as the "ground truth data" that our model's detection results will be compared to.

```
!cp /content/images/test/* /content/mAP/input/images-optional # Copy images and xml files
!mv /content/mAP/input/images-optional/*.xml /content/mAP/input/ground-truth/  # Move xml files to the appropriate folder
```

The calculator tool expects annotation data in a format that's different from the Pascal VOC .xml file format we're using. Fortunately, it provides an easy script, `convert_gt_xml.py`, for converting to the expected .txt format.

```
!python /content/mAP/scripts/extra/convert_gt_xml.py
```

```
Conversion completed!
```

Okay, we've set up the ground truth data, but now we need actual detection results from our model. The detection results will be compared to the ground truth data to calculate the model's accuracy in mAP.

The inference function we defined in Step 7.1 can be used to generate detection data for all the images in the **test** folder. We'll use it the same as before, except this time we'll tell it to save detection results into the `detection-results` folder.

Click Play to run the following code block!