

Création de copie...

After this cell executes runtime will restart to complete the install, ignore and close the message, continue running the cells below this one

```

[1] # %%capture
#After this cell executes runtime will restart to finish the install, ignore and close the crash message, continue running cells starting with !pip install numpy==1.17.5;
!pip install numpy==1.17.5;

Requirement already satisfied: numpy==1.17.5 in /usr/local/lib/python3.7/dist-packages (1.17.5)

[2] %tensorflow_version 1.x
!pip install tf_slim==1.1.0

TensorFlow 1.x selected.
Collecting tf_slim==1.1.0
  Downloading tf_slim-1.1.0-py2.py3-none-any.whl (352 kB)
    0% |████████████████████████████████| 352 kB 5.2 MB/s
Requirement already satisfied: absl-py>=0.2.2 in /usr/local/lib/python3.7/dist-packages (from tf_slim==1.1.0) (1.0.0)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from absl-py>=0.2.2->tf_slim==1.1.0) (1.15.0)
Installing collected packages: tf-slim
Successfully installed tf-slim-1.1.0

[4] # For faster training time, images should be resized to 300x300 and then annotated
# Images should contain the objects of interest at various scales, angles, lighting conditions, locations
# For acceptable results - mAP@0.5 of 0.9 the model was trained with batch size of 24
# and 5000 steps. this takes about 1h using 2 augmentations.
# using 5 augmentations it takes about 2h
num_steps = 5000 # A step means using a single batch of data. larger batch, less steps required
#Number of evaluation steps.
num_eval_steps = 50
#Batch size 24 is a setting that generally works well. can be changed higher or lower
MODELS_CONFIG = {
    'ssd_mobilenet_v2': {
        'model_name': 'ssd mobilenet_v2_coco_2018_03_29',
        'pipeline_file': 'ssd_mobilenet_v2_coco.config',
        'batch_size': 24
    }
}
selected_model = 'ssd_mobilenet_v2'

# Name of the object detection model to use.
MODEL = MODELS_CONFIG[selected_model]['model_name']

# Name of the pipeline file in tensorflow object detection API.
pipeline_file = MODELS_CONFIG[selected_model]['pipeline_file']

# Training batch size fits in Colab's GPU memory for selected model.
batch_size = MODELS_CONFIG[selected_model]['batch_size']

```

Clone the depthai-ml-training repository

```

[7] repo_url = 'https://github.com/luxonis/depthai-ml-training.git'
import os
%cd /content
repo_dir_path = "/content/depthai-ml-training"
!rm -rf {repo_dir_path}
!git clone {repo_url} {repo_dir_path}
%cd {repo_dir_path}
!git pull

/content
Cloning into '/content/depthai-ml-training'...
remote: Enumerating objects: 480, done.
remote: Counting objects: 100% (34/34), done.
remote: Compressing objects: 100% (34/34), done.
remote: Total 480 (delta 3), reused 0 (delta 0), pack-reused 455
Receiving objects: 100% (480/480), 100.97 MiB | 19.44 MiB/s, done.
Resolving deltas: 100% (235/235), done.
/content/depthai-ml-training
Already up to date.

```

Download our example dataset of PASCAL cats, dogs, and birds

You can skip this if you are using your own data.

```

[ ] %cd /content
!pip install -U --no-cache-dir gdown==4.4.0 --pre
!gdown --id 1sJ3E001OsPbPYmhC0HWTqD2wb7EMbMUG
!unzip "/content/pascal_animals.zip"
!rm -rf "/content/pascal_animals.zip"
!mv /content/sorted/* /content
!rm -rf sorted

Streaming output truncated to the last 5000 lines.
inflate: sorted/validation/2008_004265.xml
inflate: sorted/validation/2011_001959.jpg
inflate: sorted/validation/2008_003587.jpg
inflate: sorted/validation/2010_000769.xml
inflate: sorted/validation/2011_002389.xml
inflate: sorted/validation/2008_006578.xml
inflate: sorted/validation/2009_004334.xml
inflate: sorted/validation/2008_008387.xml
inflate: sorted/validation/2009_002431.xml
inflate: sorted/validation/2010_003138.jpg
inflate: sorted/validation/2009_000566.xml
inflate: sorted/validation/2009_001251.xml
inflate: sorted/validation/2008_005898.jpg
inflate: sorted/validation/2008_005735.xml
inflate: sorted/validation/2009_001075.jpg
inflate: sorted/validation/2010_002142.jpg
inflate: sorted/validation/2010_005386.xml
inflate: sorted/validation/2008_006808.xml
inflate: sorted/validation/2010_001885.jpg
inflate: sorted/validation/2011_001464.jpg
inflate: sorted/validation/2011_002433.xml
inflate: sorted/validation/2010_004970.xml
inflate: sorted/validation/2010_005222.jpg
inflate: sorted/validation/2008_001498.xml
inflate: sorted/validation/2009_005160.xml
inflate: sorted/validation/2008_000960.xml
inflate: sorted/validation/2008_004620.xml
inflate: sorted/validation/2008_000080.jpg
inflate: sorted/validation/2009_000193.jpg
inflate: sorted/validation/2008_000066.jpg
inflate: sorted/validation/2010_003823.jpg
inflate: sorted/validation/2010_000639.xml
inflate: sorted/validation/2008_005252.jpg
inflate: sorted/validation/2008_008404.jpg

```

```

inflating: sorted/validation/2008_004797.jpg
inflating: sorted/validation/2009_000684.xml
inflating: sorted/validation/2009_004713.xml
inflating: sorted/validation/2008_000620.xml
inflating: sorted/validation/2010_004476.xml
inflating: sorted/validation/2010_004244.jpg
inflating: sorted/validation/2009_004983.jpg
inflating: sorted/validation/2009_005614.jpg
inflating: sorted/validation/2009_001570.jpg
inflating: sorted/validation/2008_006032.jpg
inflating: sorted/validation/2008_001479.jpg
inflating: sorted/validation/2008_000512.jpg
inflating: sorted/validation/2008_002037.jpg
inflating: sorted/validation/2009_000029.xml
inflating: sorted/validation/2010_005511.xml
inflating: sorted/validation/2010_001351.jpg
inflating: sorted/validation/2008_001120.jpg
inflating: sorted/validation/2011_000430.jpg
inflating: sorted/validation/2010_003987.xml
inflating: sorted/validation/2010_001242.xml
inflating: sorted/validation/2009_002553.xml
inflating: sorted/validation/2008_007151.xml
inflating: sorted/validation/2010_000170.jpg

```

▼ [Optional] If you wish to use your own data, run the cells below.

(Otherwise the model can train with the data already present in the cloned repo)

```

✓ [8] cd /content
/content

✓ [9] !curl -L "https://app.roboflow.com/ds/Al9AavgwRi?key=IPxUeyjXRs" > roboflow.zip; unzip roboflow.zip; rm roboflow.zip
extracting: train/20220427_134317.jpg_rf_5fd5a16ea5f7c0eefdf6b6e25faed111.xml
extracting: train/20220427_134317.jpg_rf_6fcfa0e002146b5a61a18e771ff0e2e.jpg
extracting: train/20220427_134317.jpg_rf_6fcfa0e002146b5a61a18e771ff0e2e.xml
extracting: train/20220427_134334.jpg_rf_r_10b6192ea8bcfa4edecfa0addaaef0fa.jpg
extracting: train/20220427_134334.jpg_rf_r_10b6192ea8bcfa4edecfa0addaaef0fa.xml
extracting: train/20220427_134334.jpg_rf_r_10b6192ea8bcfa4edecfa0addaaef0fa.xml
extracting: train/20220427_134334.jpg_rf_r_10b60d40b3a0099a7ac1efee3d65f5f3r.jpg
extracting: train/20220427_134334.jpg_rf_r_10b60d40b3a0099a7ac1efee3d65f5f3r.xml
extracting: train/20220427_134334.jpg_rf_r_13696763f287a4a13e40379fffbe4416.jpg
extracting: train/20220427_134334.jpg_rf_r_13696763f287a4a13e40379fffbe4416.xml
extracting: train/20220427_134422.jpg_rf_r_1162c45110154043a1409963999eac2.jpg
extracting: train/20220427_134422.jpg_rf_r_1162c45110154043a1409963999eac2.xml
extracting: train/20220427_134422.jpg_rf_r_9f115341230b749de6928174b57febe.jpg
extracting: train/20220427_134422.jpg_rf_r_9f115341230b749de6928174b57febe.xml
extracting: train/20220427_134422.jpg_rf_r_b9409099342a421101592eb449e17.jpg
extracting: train/20220427_134422.jpg_rf_r_b9409099342a421101592eb449e17.xml
extracting: train/20220427_134443.jpg_rf_r_4bc182ce2fb7b23e5e943815acbcfc513.jpg
extracting: train/20220427_134443.jpg_rf_r_4bc182ce2fb7b23e5e943815acbcfc513.xml
extracting: train/20220427_134443.jpg_rf_r_8277f88fa1d60623ba96af9fee40f01.jpg
extracting: train/20220427_134443.jpg_rf_r_8277f88fa1d60623ba96af9fee40f01.xml
extracting: train/20220427_134443.jpg_rf_r_b085cf4cb0a2200d05ff791b1f80f05d.jpg
extracting: train/20220427_134443.jpg_rf_r_b085cf4cb0a2200d05ff791b1f80f05d.xml
extracting: train/20220427_134449.jpg_rf_r_402f3437ebf7f658897f22503b0fc8c.jpg
extracting: train/20220427_134449.jpg_rf_r_402f3437ebf7f658897f22503b0fc8c.xml
extracting: train/20220427_134449.jpg_rf_r_5e138e8440bd40f45011d58577519db.jpg
extracting: train/20220427_134449.jpg_rf_r_5e138e8440bd40f45011d58577519db.xml
extracting: train/20220427_134449.jpg_rf_r_6699c2c3331c3e89fd90398ba605a0c4.jpg
extracting: train/20220427_134449.jpg_rf_r_6699c2c3331c3e89fd90398ba605a0c4.xml
extracting: train/20220427_134459.jpg_rf_r_116d2e787bd793bcfe0a9231330b015.jpg
extracting: train/20220427_134459.jpg_rf_r_116d2e787bd793bcfe0a9231330b015.xml
extracting: train/20220427_134459.jpg_rf_r_43484aa0fe9940d966a9ab0b025f1fc.jpg
extracting: train/20220427_134459.jpg_rf_r_43484aa0fe9940d966a9ab0b025f1fc.xml
extracting: train/20220427_134507.jpg_rf_r_9db1a1bc500f7e1ce826baa1f6c37d3.jpg
extracting: train/20220427_134507.jpg_rf_r_9db1a1bc500f7e1ce826baa1f6c37d3.xml
extracting: train/20220427_134507.jpg_rf_r_279d84bf3a0f0ce1f02c799887b712082.jpg
extracting: train/20220427_134507.jpg_rf_r_279d84bf3a0f0ce1f02c799887b712082.xml
extracting: train/20220427_134507.jpg_rf_r_74d41091ce62474c58e92a3a41242b614.jpg
extracting: train/20220427_134507.jpg_rf_r_74d41091ce62474c58e92a3a41242b614.xml
extracting: train/20220427_134507.jpg_rf_r_77b83d96878e116c003d657f9d9bb81c.jpg
extracting: train/20220427_134507.jpg_rf_r_77b83d96878e116c003d657f9d9bb81c.xml
extracting: train/20220427_134558.jpg_rf_r_3664ad4ff0b7a3c07c77b42a0d2b7b69e33.jpg
extracting: train/20220427_134558.jpg_rf_r_3664ad4ff0b7a3c07c77b42a0d2b7b69e33.xml
extracting: train/20220427_134558.jpg_rf_r_ddb2b55a33c08075a7c92a07887a77e07.jpg
extracting: train/20220427_134558.jpg_rf_r_ddb2b55a33c08075a7c92a07887a77e07.xml
extracting: train/20220427_134603.jpg_rf_r_d51a1b8250552f0f88bd143f17f60fda.jpg
extracting: train/20220427_134603.jpg_rf_r_d51a1b8250552f0f88bd143f17f60fda.xml
extracting: train/20220427_134603.jpg_rf_r_8949eaa0467158081b6aac215bd2b240f.jpg
extracting: train/20220427_134603.jpg_rf_r_8949eaa0467158081b6aac215bd2b240f.xml
extracting: train/20220427_134603.jpg_rf_r_cba3832a4221c3c8b07735d27ff11c1e.jpg
extracting: train/20220427_134603.jpg_rf_r_cba3832a4221c3c8b07735d27ff11c1e.xml
extracting: train/20220427_134603.jpg_rf_r_ceb62e878e7914994fc5b5biad32269c.jpg
extracting: train/20220427_134603.jpg_rf_r_ceb62e878e7914994fc5b5biad32269c.xml
creating: valid
extracting: valid/20220427_134236.jpg_rf_r.8450fa7f60043da2d01b8500abfd0aa2a2.jpg
extracting: valid/20220427_134236.jpg_rf_r.8450fa7f60043da2d01b8500abfd0aa2a2.xml
extracting: valid/20220427_134243.jpg_rf_r.ac696c84bababf4cb3c680042c2ab088.jpg
extracting: valid/20220427_134243.jpg_rf_r.ac696c84bababf4cb3c680042c2ab088.xml
extracting: valid/20220427_134306.jpg_rf_r.527ef7a74854ab6ff8386606b8e53d7.jpg
extracting: valid/20220427_134306.jpg_rf_r.527ef7a74854ab6ff8386606b8e53d7.xml

```

▼ Mount your google drive

```

[ ] #mount your google drive.
#it will be visible in the file navigator on the left of this notebook
#there should be a folder in your drive with your data
from google.colab import drive
drive.mount('/content/gdrive')

-----  

MessageError: Traceback (most recent call last)
<ipython-input-7-3270c291341c> in <module>()
      3 #there should be a folder in your drive with your data
      4 from google.colab import drive
--> 5 drive.mount('/content/gdrive')

----- 3 frames -----
/usr/local/lib/python3.7/dist-packages/google/colab/_message.py in read_reply_from_input(message_id, timeout_sec)
    104         reply.get('colab_msg_id') == message_id:
    105             if 'error' in reply:
--> 106                 raise MessageError(reply['error'])
    107             return reply.get('data', None)
    108

MessageError: Error: credential propagation was unsuccessful

```

SEARCH STACK OVERFLOW

Own data should be split into 80% training (train folder), 20% for validation (test folder). The train and test folder should contain images and associated.xml annotations.

Have extra images for final model testing (final_test folder). These need not be annotated.

```
[ ] #copy files from gdrive to colab drive. this takes a few minutes, depending on the number of files.
#go on the file explorer on the left of this notebook and access your gdrive. find the folders
#with your train, test and final test images.
```

```

#Right click on each and copy the path. paste it btw the first " " in the corresponding lines
#training folder
# !cp -r "/path/to/data/on/Google/Drive/train" "/content/train"
#validation folder
# !cp -r "/path/to/data/on/Google/Drive/validation" "/content/validation"
#test folder
# !cp -r "/path/to/data/on/Google/Drive/test" "/content/test"

[ ] # quick check for training data files. you can also browse to the object_detection_demo_flows
# on the left and see if they were copied
!ls /content/train

```

[]

▼ Install Tensorflow Object Detection API

Clone TF models which contains the Object Detection API; also install the required dependencies

```

✓ [10] # %%capture
  %cd /content
!git clone -quiet https://github.com/tensorflow/models.git
%cd /content/models/
!git checkout 58d19c67e1d3d905ddcc6e5092348658fed80af
!apt-get update && apt-get install -y -qq protobuf-compiler python-pil python-lxml python-tk
!pip install -q cython==0.29.28 contextlib2==0.5.5 pillow==7.1.2 lxml==4.2.6 matplotlib==3.2.2
!pip uninstall -y pycocotools
!pip install --no-cache-dir pycocotools==2.0.4
%cd /content/models/research
!protoc object_detection/protos/*.proto --python_out=.
import os
os.environ['PYTHONPATH'] += ':/content/models/research:/content/models/research/slim'
!python object_detection/builders/model_builder_test.py

Selecting previously unselected package python-chardet.
Preparing to unpack .../python-chardet_3.0.4-1_all.deb ...
Unpacking python-chardet (3.0.4-1) ...
Selecting previously unselected package python-six.
Preparing to unpack .../python-six_1.11.0-2_all.deb ...
Unpacking python-six (1.11.0-2) ...
Selecting previously unselected package python-webencodings.
Preparing to unpack .../python-webencodings_0.5-2_all.deb ...
Unpacking python-webencodings (0.5-2) ...
Selecting previously unselected package python-html5lib.
Preparing to unpack .../python-html5lib_0.999999999_1_all.deb ...
Unpacking python-html5lib (0.999999999-1) ...
Selecting previously unselected package python-lxml:amd64.
Preparing to unpack .../python-lxml_4.2.1-1ubuntu0.6_amd64.deb ...
Unpacking python-lxml:amd64 (4.2.1-1ubuntu0.6) ...
Selecting previously unselected package python-olefile.
Preparing to unpack .../python-olefile_0.45.1-1_all.deb ...
Unpacking python-olefile (0.45.1-1) ...
Selecting previously unselected package python-pil:amd64.
Preparing to unpack .../python-pil_5.1.0-1ubuntu0.7_amd64.deb ...
Unpacking python-pil:amd64 (5.1.0-1ubuntu0.7) ...
Setting up python-pkg-resources (39.0.1-2) ...
Setting up python-six (1.11.0-2) ...
Setting up python-bs4 (4.6.0-1) ...
Setting up python-lxml:amd64 (4.2.1-1ubuntu0.6) ...
Setting up python-olefile (0.45.1-1) ...
Setting up python-pil:amd64 (5.1.0-1ubuntu0.7) ...
Setting up python-webencodings (0.5-2) ...
Setting up python-chardet (3.0.4-1) ...
Setting up python-html5lib (0.999999999-1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Found existing installation: pycocotools 2.0.4
Uninstalling pycocotools-2.0.4:
  Successfully uninstalled pycocotools-2.0.4
Collecting pycocotools==2.0.4
  Downloading pycocotools-2.0.4.tar.gz (106 kB)
    [██████████] 106 kB 5.6 MB/s
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
    Preparing wheel metadata ... done
Requirement already satisfied: matplotlib>=2.1.0 in /usr/local/lib/python3.7/dist-packages (from pycocotools==2.0.4) (3.2.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from pycocotools==2.0.4) (1.17.5)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib>=2.1.0>pycocotools==2.0.4) (1.4.2)
Requirement already satisfied: pyyaml!=2.0.4,!!=2.1.2,>=2.1.6,>=2.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib>=2.1.0>pycocotools==2.0.4) (3.12.2)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib>=2.1.0>pycocotools==2.0.4) (2.8.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from matplotlib>=2.1.0>pycocotools==2.0.4) (0.11.0)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packages (from kiwisolver>=1.0.1>matplotlib>=2.1.0>pycocotools==2.0.4)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil>=2.1.0>pycocotools==2.0.4)
Building wheels for collected packages: pycocotools
  Building wheel for pycocotools (PEP 517) ... done
  Created wheel for pycocotools: filename=pycocotools-2.0.4-cp37-cp37m-linux_x86_64.whl size=265209 sha256=8b5556037b945fae5e4c3f724d022a2caf22
  Stored in directory: /tmp/pip-ephem-wheel-cache-znrve2ow/wheels/a3/f5/fa/f011e578cc76e1fc5be8dce30b3eb9fd00f337e744b3bba59b
Successfully built pycocotools
Installing collected packages: pycocotools
Successfully installed pycocotools-2.0.4
/content/models/research
object_detection/protos/input_reader.proto: warning: Import object_detection/protos/image_resizer.proto but not used.

```

▼ Prepare tfrecord files

```

✓ [11] # %%capture
  %cd /content/

# Convert train folder annotation xml files to a single csv file,
# generate the 'label_map.pbtxt' file to 'data/' directory as well.
!python depthai-ml-training/helpers/xml_to_csv.py -i /content/train/ -o /content/train_labels.csv -l /content
# Convert test folder annotation xml files to a single csv.
!python depthai-ml-training/helpers/xml_to_csv.py -i /content/test/ -o /content/test_labels.csv

# Generate 'train.record'
!python depthai-ml-training/helpers/generate_tfrecord.py --csv_input=/content/train_labels.csv --output_path=/content/train.record --img_path=/content

# Generate 'test.record'
!python depthai-ml-training/helpers/generate_tfrecord.py --csv_input=/content/test_labels.csv --output_path=/content/test.record --img_path=/content

# Set the paths
test_record_fname = '/content/test.record'
train_record_fname = '/content/train.record'
label_map_pbtxt_fname = '/content/label_map.pbtxt'

/content
Successfully converted xml to csv.
Generate '/content/label_map.pbtxt'
Successfully converted xml to csv.
WARNING:tensorflow:From depthai-ml-training/helpers/generate_tfrecord.py:138: The name tf.app.run is deprecated. Please use tf.compat.v1.app.run instead.
WARNING:tensorflow:From depthai-ml-training/helpers/generate_tfrecord.py:111: The name tf.python_io.TFRecordWriter is deprecated. Please use tf.io.TFRecordWriter instead.
W0503 23:06:40.522264 140220150364032 module_wrapper.py:139] From depthai-ml-training/helpers/generate_tfrecord.py:111: The name tf.python_io.TFRE
WARNING:tensorflow:From depthai-ml-training/helpers/generate_tfrecord.py:57: The name tf.gfile.GFile is deprecated. Please use tf.io.GFile instead.

```

```

W0503 23:06:40.555526 140220150364032 module_wrapper.py:139] From depthai-ml-training/helpers/generate_tfrecord.py:57: The name tf.gfile.GFile is
Successfully created the TFRecords: /content/train.record
WARNING:tensorflow:From depthai-ml-training/helpers/generate_tfrecord.py:138: The name tf.app.run is deprecated. Please use tf.compat.v1.app.run
WARNING:tensorflow:From depthai-ml-training/helpers/generate_tfrecord.py:111: The name tf.python_io.TFRecordWriter is deprecated. Please use tf.io
W0503 23:06:42.781179 139783933269888 module_wrapper.py:139] From depthai-ml-training/helpers/generate_tfrecord.py:111: The name tf.python_io.TFRE
WARNING:tensorflow:From depthai-ml-training/helpers/generate_tfrecord.py:57: The name tf.gfile.GFile is deprecated. Please use tf.io.gfile.GFile i
W0503 23:06:42.790517 139783933269888 module_wrapper.py:139] From depthai-ml-training/helpers/generate_tfrecord.py:57: The name tf.gfile.GFile is
Traceback (most recent call last):
  File "depthai-ml-training\helpers\generate_tfrecord.py", line 138, in <module>
    tf.app.run()
  File "/tensorflow-1.15.2/python3.7/tensorflow_core/python/platform/app.py", line 40, in run
    _run(main, args=args, flags_parser=parse_flags_tolerate_undef)
  File "/usr/local/lib/python3.7/dist-packages/absl/app.py", line 312, in run
    run_main(main, args)
  File "/usr/local/lib/python3.7/dist-packages/absl/app.py", line 258, in _run_main
    sys.exit(main(argv))
  File "depthai-ml-training\helpers\generate_tfrecord.py", line 129, in main
    tf_example = create_tf_example(group, path, label_map)
  File "depthai-ml-training\helpers\generate_tfrecord.py", line 58, in create_tf_example
    encoded_jpg = fid.read()
  File "/tensorflow-1.15.2/python3.7/tensorflow_core/python/lib/io/file_io.py", line 122, in read
    self._preread_check()
  File "/tensorflow-1.15.2/python3.7/tensorflow_core/python/lib/io/file_io.py", line 84, in _preread_check
    compat.as_bytes(self._name), 1024 * 512)
tensorflow.python.framework.errors_impl.NotFoundError: /content/validation/20220427_130943.jpg,rf.23efbcddfe78583596f518357050cf1.jpg; No such fi

```

▼ Download the Mobilenet SSD v2 Model

```

✓ [12] %cd /content/models/research
import os
import shutil
import glob
import urllib.request
import tarfile
MODEL_FILE = MODEL + '.tar.gz'
DOWNLOAD_BASE = 'http://download.tensorflow.org/models/object_detection/'
DEST_DIR = '/content/models/research/pretrained_model'

if not (os.path.exists(MODEL_FILE)):
    urllib.request.urlretrieve(DOWNLOAD_BASE + MODEL_FILE, MODEL_FILE)

tar = tarfile.open(MODEL_FILE)
tar.extractall()
tar.close()

os.remove(MODEL_FILE)
if (os.path.exists(DEST_DIR)):
    shutil.rmtree(DEST_DIR)
os.rename(MODEL, DEST_DIR)
!echo {DEST_DIR}
!ls -alh {DEST_DIR}

/content/models/research
/content/models/research/pretrained_model
total 135M
drwxr-xr-x 3 345018 89939 4.0K Mar 30 2018 .
drwxr-xr-x 63 root root 4.0K May 3 23:06 ..
-rw-r--r-- 1 345018 89939 77 Mar 30 2018 checkpoint
-rw-r--r-- 1 345018 89939 67M Mar 30 2018 frozen_inference_graph.pb
-rw-r--r-- 1 345018 89939 65M Mar 30 2018 model.ckpt.data-00000-of-00001
-rw-r--r-- 1 345018 89939 15K Mar 30 2018 model.ckpt.index
-rw-r--r-- 1 345018 89939 3.4M Mar 30 2018 model.ckpt.meta
-rw-r--r-- 1 345018 89939 4.2K Mar 30 2018 pipeline.config
drwxr-xr-x 3 345018 89939 4.0K Mar 30 2018 saved_model

✓ [13] #TF pretrained model checkpoint
fine_tune_checkpoint = os.path.join(DEST_DIR, "model.ckpt")
fine_tune_checkpoint
'/content/models/research/pretrained_model/model.ckpt'

```

▼ Configuring a Training Pipeline

```

✓ [14] import os
pipeline_fname = os.path.join('/content/models/research/object_detection/samples/configs/', pipeline_file)

assert os.path.isfile(pipeline_fname), '{} not exist'.format(pipeline_fname)
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())

✓ [15] from pathlib import Path
parent = Path(label_map_pbtxt_fname).parent
!ls (parent)

depthai-ml-training README.roboflow.txt test.record valid
label_map.pbtxt sample_data train
models test train_labels.csv
README.dataset.txt test_labels.csv train.record

✓ [16] import re
iou_threshold = 0.50
num_classes = get_num_classes(label_map_pbtxt_fname)
with open(pipeline_fname) as f:
    s = f.read()
    with open(pipeline_fname, 'w') as f:

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

        # tfrrecord files train and test.
        s = re.sub(
            '(input_path: ".*?")({train.record}(.*)?"', 'input_path: "{}".format({train_record_fname}), s)
        s = re.sub(
            '(input_path: ".*?")({val.record}(.*)?"', 'input_path: "{}".format({test_record_fname}), s)

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

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

f.write(s)

[ ] # Have a look at the config file with various settings
# !cat {pipeline_fname}

```

▼ Train the model

► [Optional] The cell below adds Tensorboard visualization to the training process.

Will open in new tab.

```
[ ] 41 cellule masquée
```

▼ Start the training

```

✓ [17] model_dir = 'training/'

# Optionally remove content in output model directory for a fresh start.
# !rm -rf {model_dir}
# os.makedirs(model_dir, exist_ok=True)
!python /content/models/research/object_detection/model_main.py \
--pipeline_config_path={pipeline_fname} \
--model_dir={model_dir} \
--alsologtostderr \
--num_train_steps={num_steps} \
--num_eval_steps={num_eval_steps}

I0504 00:08:21.725332 140046207149952 convolutional_box_predictor:py:156] depth of additional conv before box predictor: 0
INFO:tensorflow:depth of additional conv before box predictor: 0
I0504 00:08:21.764342 140046207149952 convolutional_box_predictor:py:156] depth of additional conv before box predictor: 0
INFO:tensorflow:depth of additional conv before box predictor: 0
I0504 00:08:21.803318 140046207149952 convolutional_box_predictor:py:156] depth of additional conv before box predictor: 0
INFO:tensorflow:depth of additional conv before box predictor: 0
I0504 00:08:21.850974 140046207149952 convolutional_box_predictor:py:156] depth of additional conv before box predictor: 0
INFO:tensorflow:depth of additional conv before box predictor: 0
I0504 00:08:21.890399 140046207149952 convolutional_box_predictor:py:156] depth of additional conv before box predictor: 0
INFO:tensorflow:Done calling model_fn.
I0504 00:08:22.967236 140046207149952 estimator.py:1150] Done calling model_fn.
WARNING:tensorflow:From /tensorflow/1.15.2/python3.7/tensorflow_core/python/saved_model/signature_def_utils_impl.py:201: build_tensor_info (from Instructions for updating:
This function will only be available through the v1 compatibility library as tf.compat.v1.saved_model.utils.build_tensor_info or tf.compat.v1.saved_model.Signatures INCLUDED in export for Classify: None
W0504 00:08:22.967541 140046207149952 deprecation.py:323] From /tensorflow/1.15.2/python3.7/tensorflow_core/python/saved_model/signature_def_utils_impl.py:201: build_tensor_info (from Instructions for updating:
This function will only be available through the v1 compatibility library as tf.compat.v1.saved_model.utils.build_tensor_info or tf.compat.v1.saved_model.Signatures INCLUDED in export for Classify: None
INFO:tensorflow:Signatures INCLUDED in export for Regress: None
I0504 00:08:22.968488 140046207149952 export_utils.py:170] Signatures INCLUDED in export for Regress: None
INFO:tensorflow:Signatures INCLUDED in export for Predict: ['tensorflow/serving/predict', 'serving_default']
I0504 00:08:22.968623 140046207149952 export_utils.py:170] Signatures INCLUDED in export for Predict: ['tensorflow/serving/predict', 'serving_default']
INFO:tensorflow:Signatures INCLUDED in export for Train: None
I0504 00:08:22.968745 140046207149952 export_utils.py:170] Signatures INCLUDED in export for Train: None
INFO:tensorflow:Signatures INCLUDED in export for Eval: None
I0504 00:08:22.968894 140046207149952 export_utils.py:170] Signatures INCLUDED in export for Eval: None
2022-05-04 00:08:22.969457: I tensorflow/stream_executor/cuda/cuda_gpu_executor:cc:983] successful NUMA node read from SysFS had negative value
2022-05-04 00:08:22.970068: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1639] Found device 0 with properties:
name: Tesla K80 major: 3 minor: 7 memoryClockRate(GHz): 0.8235
pciBusID: 0000:00:04.0
2022-05-04 00:08:22.970159: I tensorflow/stream_executor/default/dso_loader.cc:44] Successfully opened dynamic library libcudart.so.10.
2022-05-04 00:08:22.970212: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcublas.so.10
2022-05-04 00:08:22.970285: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcurlft.so.10
2022-05-04 00:08:22.970342: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcurlrand.so.10
2022-05-04 00:08:22.970395: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcurlsolver.so.1
2022-05-04 00:08:22.970446: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcusparse.so.1
2022-05-04 00:08:22.970495: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcudnn.so.7
2022-05-04 00:08:22.970593: I tensorflow/stream_executor/cuda/cuda_gpu_executor:cc:983] successful NUMA node read from SysFS had negative value
2022-05-04 00:08:22.971161: I tensorflow/stream_executor/cuda/cuda_gpu_executor:cc:983] successful NUMA node read from SysFS had negative value
2022-05-04 00:08:22.971618: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1180] Adding visible gpu devices: 0
2022-05-04 00:08:22.971704: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1186] 0
2022-05-04 00:08:22.971723: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1199] 0: N
2022-05-04 00:08:22.971873: I tensorflow/stream_executor/cuda/cuda_gpu_executor:cc:983] successful NUMA node read from SysFS had negative value
2022-05-04 00:08:22.972498: I tensorflow/stream_executor/cuda/cuda_gpu_executor:cc:983] successful NUMA node read from SysFS had negative value
2022-05-04 00:08:22.973016: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1325] Created TensorFlow device (/job:localhost/replica:0/task:0)
INFO:tensorflow:Restoring parameters from training/model.ckpt-5000
I0504 00:08:22.975748 140046207149952 saver.py:1284] Restoring parameters from training/model.ckpt-5000
INFO:tensorflow:Assets added to graph.
INFO:tensorflow:No assets to write.
I0504 00:08:23.449465 140046207149952 builder_impl.py:460] No assets to write.
INFO:tensorflow:SavedModel written to: training/export/Servo/temp-b'1651622898'/saved_model.pb
I0504 00:08:24.236282 140046207149952 builder_impl.py:425] SavedModel written to: training/export/Servo/temp-b'1651622898'/saved_model.pb
INFO:tensorflow:Loss for final step: 0.7604284.
I0504 00:08:24.669312 140046207149952 estimator.py:371] Loss for final step: 0.7604284.

```

```

✓ [19] ls {model_dir}

```

```

checkpoint
eval_0
events.out.tfevents.1651619284.26e37d6c46a8
export
graph.pbtxt
model.ckpt-1673.data-00000-of-00001
model.ckpt-1673.index
model.ckpt-1673.meta
model.ckpt-2518.data-00000-of-00001
model.ckpt-2518.index
model.ckpt-2518.meta
model.ckpt-3355.data-00000-of-00001
model.ckpt-3355.index
model.ckpt-3355.meta
model.ckpt-4193.data-00000-of-00001
model.ckpt-4193.index
model.ckpt-4193.meta
model.ckpt-5000.data-00000-of-00001
model.ckpt-5000.index
model.ckpt-5000.meta

```

▼ Export a Trained Inference Graph

Once your training job is complete, you need to extract the newly trained inference graph, which will be later used to perform the object detection. This can be done as follows:

```
[ ] #clean output_directory if necessary to start fresh:  
# !rm -rf /content/object_detection_demo/fine_tuned_model/  
# os.makedirs('/content/object_detection_demo_flow/fine_tuned_model/', exist_ok=True)  
  
✓ [20] %%capture  
import re  
import numpy as np  
  
output_directory = './fine_tuned_model'  
# output_directory = '/content/gdrive/My\ Drive/data/'  
  
lst = os.listdir(model_dir)  
lst = [l for l in lst if 'model.ckpt' in l and '.meta' in l]  
steps=np.array([int(re.findall('\d+', l[0])) for l in lst])  
last_model = lst[steps.argmax()].replace('.meta', '')  
  
last_model_path = os.path.join(model_dir, last_model)  
print(last_model_path)  
!python /content/models/research/object_detection/export_inference_graph.py \  
--input_type=image_tensor \  
--pipeline_config_path=pipeline_fname \  
--output_directory=output_directory \  
--trained_checkpoint_prefix={last_model_path}  
  
✓ [21] #export directory check  
!ls {output_directory}  
  
checkpoint          model.ckpt.index  saved_model  
frozen_inference_graph.pb    model.ckpt.meta  
model.ckpt.data-00000-of-00001 pipeline.config  
  
✓ [22] import os  
pb_fname = os.path.abspath(output_directory), "frozen_inference_graph.pb")  
assert os.path.isfile(pb_fname), '{} not exist'.format(pb_fname)  
  
# !ls -alh {pb_fname}
```

▼ Running Inference: Checking what the trained model can detect

Test with images in repository `object_detection_demo_flow`/data/images/final test directory.

```
✓ [23] import os  
import glob  
  
# Path to frozen detection graph. This is the actual model that is used for the object detection.  
PATH_TO_CKPT = pb_fname  
  
# List of the strings that is used to add correct label for each box.  
PATH_TO_LABELS = label_map_pbtxt_fname  
  
# If you want to test the code with your images, just add images files to the PATH_TO_TEST_IMAGES_DIR.  
PATH_TO_TEST_IMAGES_DIR = "/content/test/mytest"  
  
assert os.path.isfile(pb_fname)  
assert os.path.isfile(PATH_TO_LABELS)  
TEST_IMAGE_PATHS = glob.glob(os.path.join(PATH_TO_TEST_IMAGES_DIR, "*.*"))  
assert len(TEST_IMAGE_PATHS) > 0, 'No image found in {}'.format(PATH_TO_TEST_IMAGES_DIR)  
  
✓ [30] %cd /content/models/research/object_detection  
  
import numpy as np  
import os  
import six.moves.urllib as urllib  
import sys  
import tarfile  
import tensorflow as tf  
import zipfile  
  
from collections import defaultdict  
from io import StringIO  
# This is needed to display the images.  
%matplotlib inline  
import matplotlib.pyplot as plt  
from PIL import Image  
  
# This is needed since the notebook is stored in the object_detection folder.  
sys.path.append("..")  
from object_detection.utils import ops as utils_ops  
  
from object_detection.utils import label_map_util  
  
from object_detection.utils import visualization_utils as vis_util  
  
detection_graph = tf.Graph()  
with detection_graph.as_default():  
    od_graph_def = tf.GraphDef()  
    with tf.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:  
        serialized_graph = fid.read()  
        od_graph_def.ParseFromString(serialized_graph)  
        tf.import_graph_def(od_graph_def, name='')  
  
label_map = label_map_util.load_labelmap(PATH_TO_LABELS)  
categories = label_map_util.convert_label_map_to_categories(  
    label_map, max_num_classes=num_classes, use_display_name=True)  
category_index = label_map_util.create_category_index(categories)  
  
def load_image_into_numpy_array(image):  
    (im_width, im_height) = image.size  
    return np.array(image.getdata()).reshape(  
        (im_height, im_width, 3)).astype(np.uint8)  
  
# Size, in inches, of the output images.  
IMAGE_SIZE = (12, 8)  
  
def run_inference_for_single_image(image, graph):  
    with graph.as_default():  
        with tf.Session() as sess:  
            # Get handles to input and output tensors  
            ops = tf.get_default_graph().get_operations()  
            all_tensor_names = [  
                output.name for op in ops for output in op.outputs]  
            tensor_dict = {}  
            for key in [  
                'num_detections', 'detection_boxes', 'detection_scores',
```

```

        'detection_classes', 'detection_masks'
    ];
    tensor_name = key + ':0'
    if tensor_name in all_tensor_names:
        tensor_dict[key] = tf.get_default_graph().get_tensor_by_name(
            tensor_name)
if 'detection_masks' in tensor_dict:
    # The following processing is only for single image
    detection_boxes = tf.squeeze(
        tensor_dict['detection_boxes'], [0])
    detection_masks = tf.squeeze(
        tensor_dict['detection_masks'], [0])
    # Reframe is required to translate mask from box coordinates to image coordinates and fit the image size.
    real_num_detection = tf.cast(
        tensor_dict['num_detections'][0], tf.int32)
    detection_boxes = tf.slice(detection_boxes, [0, 0, [
        real_num_detection, -1]])
    detection_masks = tf.slice(detection_masks, [0, 0, 0], [
        real_num_detection, -1, -1])
    detection_masks_reframed = utils_ops.reframe_box_masks_to_image_masks(
        detection_masks, detection_boxes, image.shape[0], image.shape[1])
    detection_masks_reframed = tf.cast(
        tf.greater(detection_masks_reframed, 0.5), tf.uint8)
    # Follow the convention by adding back the batch dimension
    tensor_dict['detection_masks'] = tf.expand_dims(
        detection_masks_reframed, 0)
image_tensor = tf.get_default_graph().get_tensor_by_name('image_tensor:0')

# Run inference
output_dict = sess.run(tensor_dict,
                      feed_dict={image_tensor: np.expand_dims(image, 0)})

# all outputs are float32 numpy arrays, so convert types as appropriate
output_dict['num_detections'] = int(
    output_dict['num_detections'][0])
output_dict['detection_classes'] = output_dict[
    'detection_classes'][0].astype(np.uint8)
output_dict['detection_boxes'] = output_dict['detection_boxes'][0]
output_dict['detection_scores'] = output_dict['detection_scores'][0]
if 'detection_masks' in output_dict:
    output_dict['detection_masks'] = output_dict['detection_masks'][0]
return output_dict

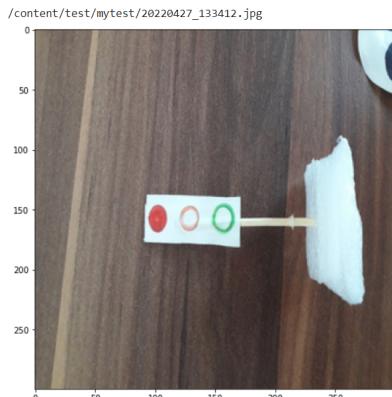
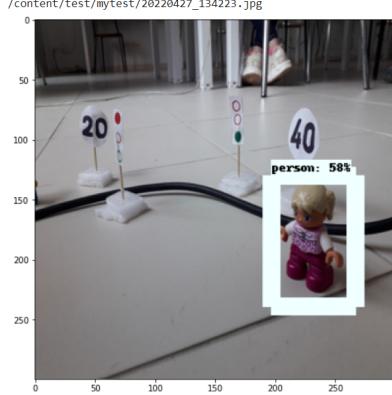
```

/content/models/research/object_detection

```

✓ [32] # visualize the results
%matplotlib inline
plt.closer('all')
for image_path in TEST_IMAGE_PATHS[21]:
    image = Image.open(image_path)
    print(image_path)
    # the array based representation of the image will be used later in order to prepare the
    # result image with boxes and labels on it.
    image_np = load_image_into_numpy_array(image)
    # Expand dimensions since the model expects images to have shape: [1, None, None, 3]
    image_np_expanded = np.expand_dims(image_np, axis=0)
    # Actual detection.
    output_dict = run_inference_for_single_image(image_np, detection_graph)
    # Visualization of the results of a detection.
    vis_util.visualize_boxes_and_labels_on_image_array(
        image_np,
        output_dict['detection_boxes'],
        output_dict['detection_classes'],
        output_dict['detection_scores'],
        category_index,
        instance_masks=output_dict.get('detection_masks'),
        use_normalized_coordinates=True,
        line_thickness=15)
    plt.figure(figsize=IMAGE_SIZE)
    plt.imshow(image_np)
    plt.show()

```



This can be used to run inference on OpenVINO.

In order to run the model on DepthAI modules, we then compile the IR obtained above to a .blob (via a server we set up just for that)

- First, we install Open Vino 21.03

```
✓ [33] %cd ../..
/content/models

✓ [34] import os
from urllib.parse import urlparse

## install tools. Open Vino takes some time to download - it's ~400MB
!sudo apt-get install -y pcutils cpio
!sudo apt autoremove

## download installation files
url = "https://registrationcenter-download.intel.com/akdlm/irc_nas/17662/1_openvino_toolkit_p_2021.3.394.tgz"
!wget {url}

## Get the name of the tgz
parsed = urlparse(url)
openvino_tgz = os.path.basename(parsed.path)
openvino_folder = os.path.splitext(openvino_tgz)[0]

## Extract & install openvino
!tar xf {openvino_tgz}
!cd {openvino_folder}
!./install_openvino_dependencies.sh && \
    sed -i 's/decline/accept/g' silent.cfg && \
    ./install.sh --silent silent.cfg

Setting up libcdparanoia0:amd64 (3.10.2+debian-13) ...
Setting up libvo-aacenc0:amd64 (0.1.3-1) ...
Setting up libgssdp-1.0-3:amd64 (1.0.2-2) ...
Setting up libwidmidi-config (0.4.2-1) ...
Setting up libas2-0.7.4:amd64 (0.7.4-19) ...
Setting up libmpegutilis-2.1-0 (1:2.1.0+debian-5) ...
Setting up libvo-amvenc0:amd64 (0.1.3-1) ...
Setting up libaa1:amd64 (1.4p-4d4build2) ...
Setting up python3-pip (9.0.1-2.3~ubuntu18.04.5) ...
Setting up libfoa0:amd64 (0.9.3-15) ...
Setting up libvisual-0.4-0:amd64 (0.4.0-11) ...
Setting up libopencore-amrnb0:amd64 (0.1.3-2.1) ...
Setting up liborc-0.4-0:amd64 (1:0.4.28-1) ...
Setting up libsrtp2-1:amd64 (2.1.0-1) ...
Setting up libsoundtouch1:amd64 (1.9.2-3) ...
Setting up python3-setuptools (39.0.1-2) ...
Setting up libdc0:amd64 (0.0.5-10) ...
Setting up libserd-0.0:amd64 (0.28.0-dfsg0-1) ...
Setting up libopencore-amrnb0:amd64 (0.1.3-2.1) ...
Setting up libshout3:amd64 (2.4.1-2build1) ...
Setting up libfluidsynth1:amd64 (1.1.9-1) ...
Setting up libplex2-2.1-0 (1:2.1.0+debian-5) ...
Setting up libdv4:amd64 (1.0.0-11) ...
Setting up libtag1v5-vanilla:amd64 (1.11.1+dfsg.1-0.2build2) ...
Setting up libvdvread4:amd64 (6.0.0-1) ...
Setting up python3-venv (3.6.7-1-18.04) ...
Setting up libmpeg2encpp-2.1-0 (1:2.1.0+debian-5) ...
Setting up libtag1v5:amd64 (1.11.1+dfsg.1-0.2build2) ...
Setting up libgstreamer1.0-0:amd64 (1.14.5-0ubuntu1-18.04.2) ...
Setcap worked! gst-pty-helper is not suid!
Setting up libgupnp-1.0-4:amd64 (1.0.2-2) ...
Setting up libv4l-0:amd64 (1.14.2-1) ...
Setting up flex (2.6.4-6) ...
Setting up libvdnav4:amd64 (6.0.0-1) ...
Setting up libisor0-0:amd64 (0.16.0-dfsg0-1) ...
Setting up libwidmidi2:amd64 (0.4.2-1) ...
Setting up libgupnp-igd-1.0-4:amd64 (0.2.5-1) ...
Setting up gstreamer1.0-tools (1.14.5-0ubuntu1-18.04.2) ...
Setting up libnice10:amd64 (0.1.14-1) ...
Setting up libtag-extras1 (1.0.1-3.1) ...
Setting up libgstreamer-plugins-base1.0-0:amd64 (1.14.5-0ubuntu1-18.04.3) ...
Setting up libratoam-0-0:amd64 (0.6.0-dfsg0-1) ...
Setting up gstreamer1.0-plugins-base:amd64 (1.14.5-0ubuntu1-18.04.3) ...
Setting up liblilyv-0-0 (0.24.2-dfsg0-1) ...
Setting up libzbar0:amd64 (0.10+doc-10.1build2) ...
Setting up libgstreamer-glib-0-0:amd64 (1.14.5-0ubuntu1-18.04.3) ...
Setting up libgstreamer-plugins-bad1.0-0:amd64 (1.14.5-0ubuntu1-18.04.1) ...
Setting up gstreamer1.0-vaapi:amd64 (1.14.5-0ubuntu1-18.04.1) ...
Setting up gstreamer1.0-ugly:amd64 (1.14.5-0ubuntu1-18.04.1) ...
Setting up libgstreamer-plugins-good1.0-0:amd64 (1.14.5-0ubuntu1-18.04.2) ...
Setting up gstreamer1.0-alsa:amd64 (1.14.5-0ubuntu1-18.04.3) ...
Setting up gstreamer1.0-plugins-bad:amd64 (1.14.5-0ubuntu1-18.04.1) ...
Setting up gstreamer1.0-plugins-good:amd64 (1.14.5-0ubuntu1-18.04.2) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1.3) ...
Processing triggers for libmkl-dnn.so.0 is not a symbolic link
/sbin/ldconfig.real: /usr/local/lib/python3.7/dist-packages/ideep4py/lib/libmkldnn.so.0 is not a symbolic link

✓ [35] !ls
EULA.txt      install_openvino_dependencies.sh  pset             rpm
install_GUI.sh  install.sh                      PUBLIC_KEY.PUB  silent.cfg
```

[Optional] Open Vino install check, generally not needed

```
[ ] # !source /opt/intel/openvino/bin/setupvars.sh && \
#     /opt/intel/openvino/deployment_tools/demo/demo_squeezeenet_download_convert_run.sh
```

Here we run some modifications in the ssd2 OpenVINO extension for TF so that our MobileNet SSDv2 model can convert successfully to the IR

```
✓ [ ] %cd /opt/intel/openvino_2021/deployment_tools/model_optimizer/extensions/front/tf/
#openvino fixes: edit
# Read in the file, make sure the .json corresponds to the model!!!
with open('ssd_v2_support.json', 'r') as file:
    filedata = file.read()

# Replace the target string
filedata = filedata.replace('"Postprocessor/ToFloat"', '"Postprocessor/Cast_1"')

# Write the file out again
with open('ssd_v2_support.json', 'w') as file:
    file.write(filedata)
```

```
/opt/intel/openvino_2021.3.394/deployment_tools/model_optimizer/extensions/front/t
```

- ▼ Convert TF model to Open Vino Intermediate Representation

If using own model, please change to your desired name for output directory --output_dir "choose name"

```
[37] #CONVERT TF MODEL TO OPEN VINO IRv10. saved in IR_v10.fruits_mnssd_v2_6k directory or
#choose own name for -output_dir "choose name"
&cd /content/models/research/fine_tuned_model/
!source /opt/intel/openvino_2021/bin/setupvars.sh && \
    python /opt/intel/openvino_2021/deployment_tools/model_optimizer/mo.py \
        --input_model frozen_inference_graph.pb \
        --tensorflow_use_custom_operations_config /opt/intel/openvino_2021/deployment_tools/model_optimizer/extensions/front/tf/ssd_v2_support.json \
        --tensorflow_object_detection_api_pipeline_config pipeline.config \
        --reverse_input_channels \
        --output_dir ./pascal_animals \
        --data_type FP16

/content/models/research/fine_tuned_model
error: XDG_RUNTIME_DIR not set in the environment.
[setupvars.sh] OpenVINO environment initialized
[ WARNING ] Use of deprecated cli option --tensorflow_use_custom_operations_config detected. Option use in the following releases will be fatal.
Model Optimizer arguments:
Common parameters:
- Path to the Input Model: /content/models/research/fine_tuned_model/frozen_inference_graph.pb
- Path for generated IR: /content/models/research/fine_tuned_model./pascal_animals
- IR output name: frozen_inference_graph
- Log level: ERROR
- Batch: Not specified, inherited from the model
- Input layers: Not specified, inherited from the model
- Output layers: Not specified, inherited from the model
- Input shapes: Not specified, inherited from the model
- Mean values: Not specified
- Scale values: Not specified
- Scale factor: Not specified
- Precision of IR: FP16
- Enable fusing: True
- Enable grouped convolutions fusing: True
- Move mean values to preprocess section: None
- Reverse input channels: True
Tensorflow specific parameters:
- Input model in text protobuf format: False
- Path to model dump for Tensorboard: None
- List of shared libraries with Tensorflow custom layers implementation: None
- Update the configuration file with input/output node names: None
- Use configuration file used to generate the model with Object Detection API: /content/models/research/fine_tuned_model/pipeline.config
- Use the config file: /opt/intel/openvino_2021/deployment_tools/model_optimizer/extensions/front/tf/ssd_v2_support.json
- Inference Engine found in: /opt/intel/openvino_2021/python/python3.7/openvino
Inference Engine version: 2.1.2021.3.0-2787-60059f2c755-releases/2021/3
Model Optimizer version: 2021.3.0-2787-60059f2c755-releases/2021/3
[ WARNING ]
Detected not satisfied dependencies:
test-generator: not installed, required: == 0.1.1

Please install required versions of components or use install_prerequisites script
/opt/intel/openvino_2021.3.394/deployment_tools/model_optimizer/install_prerequisites/install_prerequisites_tf.sh
Note that install prerequisites scripts may install additional components.
The Preprocessor block has been removed. Only nodes performing mean value subtraction and scaling (if applicable) are kept.
[ SUCCESS ] Generated IR version 10 model.
[ SUCCESS ] XML file: /content/models/research/fine_tuned_model/pascal_animals/frozen_inference_graph.xml
[ SUCCESS ] BIN file: /content/models/research/fine_tuned_model/pascal_animals/frozen_inference_graph.bin
[ SUCCESS ] Total execution time: 43.82 seconds.
[ SUCCESS ] Memory consumed: 2023 MB.
It's been a while, check for a new version of Intel(R) Distribution of OpenVINO(TM) toolkit here https://software.intel.com/content/www/us/en/develop工具/OpenVINO.html

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

[ ] #check directory containing the exported TF trained model and the IRv10 folder
!ls pascal_animals/

frozen_inference_graph.bin      frozen_inference_graph.xml
frozen_inference_graph.mapping
```

- Now we compile the IR model to a .blob for use on DepthAI modules/platforms

- ▼ We save the blob in the IR directory from above, corresponding to --output_dir parameter above

The blob filename will be `frozen_inference_graph.blob`

```
[ ] #No changes needed here unless using custom data.
#CHOOSE the directory where you would like to save the blob.
# I use the same --output_dir as above for the IR conversion
blob_dir = "/content/models/research/fine_tuned_model/pascal_animals/"

#Copy the path of your .xml and .bin files. For that, you can look at the IR
#conversion output cell, select and copy from:
#[SUCCESS] XML file and bin file paths.
#Or you can choose to compile other .xml .bin files from a different location
#
xmlfile = "/content/models/research/fine_tuned_model/pascal_animals/frozen_inference_graph.xml"
binfile = "/content/models/research/fine_tuned_model/pascal_animals/frozen_inference_graph.bin"
!python -m pip install boto3==1.17.39
!python -m pip install blobconverter==1.2.9

import blobconverter
blob_path = blobconverter.from_openvino(
    xml=xmlfile,
    bin=binfile,
    data_type="FP16",
    shaves=6,
)
from google.colab import files
files.download(blob_path)

Requirement already satisfied: boto3==1.17.39 in /usr/local/lib/python3.7/dist-packages (1.17.39)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /usr/local/lib/python3.7/dist-packages (from boto3==1.17.39) (0.10.0)
Requirement already satisfied: botocore<1.21.0,>=1.20.39 in /usr/local/lib/python3.7/dist-packages (from boto3==1.17.39) (1.20.112)
Requirement already satisfied: s3transfer<0.4.0,>=0.3.0 in /usr/local/lib/python2.7/dist-packages (from boto3==1.17.39) (0.3.7)
Requirement already satisfied: python-dateutil<3.0.0,>>2.1 in /usr/local/lib/python3.7/dist-packages (from botocore<1.21.0,>=1.20.39->boto3==1.17.39) (1)
Requirement already satisfied: six<1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil<3.0.0,>>2.1->botocore<1.21.0,>=1.20.39->boto3==1.17.39)
Requirement already satisfied: blobconverter==1.2.9 in /usr/local/lib/python3.7/dist-packages (1.2.9)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from blobconverter==1.2.9) (2.23.0)
Requirement already satisfied: PyYAML in /usr/local/lib/python3.7/dist-packages (from blobconverter==1.2.9) (3.13)
Requirement already satisfied: boto3 in /usr/local/lib/python3.7/dist-packages (from blobconverter==1.2.9) (1.17.39)
Requirement already satisfied: s3transfer<0.4.0,>=0.3 in /usr/local/lib/python3.7/dist-packages (from boto3->blobconverter==1.2.9) (0.3.7)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /usr/local/lib/python3.7/dist-packages (from boto3->blobconverter==1.2.9) (0.10.0)
Requirement already satisfied: botocore<1.21.0,>=1.20.39 in /usr/local/lib/python3.7/dist-packages (from boto3->blobconverter==1.2.9) (1.20.112)
Requirement already satisfied: python-dateutil<3.0.0,>>2.1 in /usr/local/lib/python3.7/dist-packages (from botocore<1.21.0,>=1.20.39->boto3->blobconverter==1.2.9) (1)
Requirement already satisfied: urllib3<1.27,>=1.25.4 in /usr/local/lib/python3.7/dist-packages (from botocore<1.21.0,>=1.20.39->boto3->blobconverter==1.2.9) (1.25.4)
Requirement already satisfied: six<1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil<3.0.0,>>2.1->botocore<1.21.0,>=1.20.39->boto3->blobconverter==1.2.9) (1)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests->blobconverter==1.2.9) (2017.10.8)
Requirement already satisfied: idna<3,>>2.5 in /usr/local/lib/python3.7/dist-packages (from requests->blobconverter==1.2.9) (2.10).
```

```
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests->blobconverter==1.2.9) (3.0.4)
```

Now you can download your .blob file and run it on the DepthAI module/platform

To download locally, use the file explorer on the left to locate the file in the --output_dir folder, then right click download. Colab takes a few seconds to prepare the file, then the download prompt will appear.

Run the blob on DepthAI

To use this blob with DepthAI:

- Clone depthai demo script

```
$ git clone https://github.com/luxonis/depthai.git
```

- Install requirements

```
$ python3 install_requirements.py
```

- Create a new model directory and insert the blob

```
$ mkdir resources/nm/custom_mobilenet  
$ mv <downloaded_blob> resources/nm/custom_mobilenet/
```

- Copy config file from mobilenet-ssd model

```
$ cp resources/nm/mobilenet-ssd/mobilenet-ssd.json resources/nm/custom_mobilenet/custom_mobilenet.json
```

- Run the demo script

```
$ python3 depthai_demo.py -cnn custom_mobilenet
```

You can read more details on deployment [here](#)

```
[ ] #Compress the folder with the Tensorflow model and OpenVINO IR_V10 folder and download it locally.  
#These are in content/models/research/fine_tuned_model, so we compress the fine_tuned_model folder.  
!tar czvf fine_tuned_model.tar.gz {blob_dir}
```

```
tar: Removing leading `/' from member names  
/content/models/research/fine_tuned_model/pascal_animals/  
/content/models/research/fine_tuned_model/pascal_animals/frozen_inference_graph.bin  
/content/models/research/fine_tuned_model/pascal_animals/frozen_inference_graph.mapping  
/content/models/research/fine_tuned_model/pascal_animals/frozen_inference_graph.xml
```

```
❶ #download the compressed IRv10 folder locally  
#or can use file navigator on the left to move it to your gdrive  
from google.colab import files  
files.download("fine_tuned_model.tar.gz")
```

✓ 45 s terminée à 01:35

● ✕