

Assignment8

```
!pip install tensorflow-gpu
import tensorflow as tf
print(tf.__version__)
!git clone https://github.com/tensorflow/models.git
pwd
cd /content/models/research
pwd
!protoc object_detection/protos/*.proto --python_out=.
!pip install cython
!pip install git+https://github.com/philferriere/cocoapi.git#subdirecto
ry=PythonAPI
pwd
cp object_detection/packages/tf2/setup.py .
!python -m pip install .
!python object_detection/builders/model_builder_tf2_test.py
cd /content/training_demo/pre-trained-models/
!wget http://download.tensorflow.org/models/object_detection/tf2/202007
11/ssd_resnet101_v1_fpn_1024x1024_coco17_tpu-8.tar.gz
!tar -xvf ssd_resnet101_v1_fpn_1024x1024_coco17_tpu-8.tar.gz
pwd
cd /content/training_demo
# Create train data:
python generate_tfrecord.py -x /content/training_demo/images/train -
l /content/training_demo/annotations/label_map.pbtxt -
o /content/training_demo/annotations/train.record

# Create test data:
python generate_tfrecord.py -x /content/training_demo/images/test -
l /content/training_demo/annotations/label_map.pbtxt -
o /content/training_demo/annotations/test.record
pwd
ls
!python model_main_tf2.py --model_dir=models/my_ssd_resnet50_v1_fpn --
pipeline_config_path=models/my_ssd_resnet50_v1_fpn/pipeline.config
!python .\exporter_main_v2.py --input_type image_tensor --
pipeline_config_path .\models\my_efficientdet_d1\pipeline.config --
trained_checkpoint_dir .\models\my_efficientdet_d1\ --
output_directory .\trained-inference-graph
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
import pathlib
import tensorflow as tf
import cv2
import argparse
from google.colab.patches import cv2_imshow
```

```

gpus = tf.config.experimental.list_physical_devices('GPU')
for gpu in gpus:
    tf.config.experimental.set_memory_growth(gpu, True)

IMAGE_PATHS = '/content/workspace/images/train/image18.jpg'
PATH_TO_MODEL_DIR = '/content/workspace/exported-models/my_model'
PATH_TO_LABELS = '/content/workspace/annotations/label.pbtxt'
MIN_CONF_THRESH = float(0.60)

import time
from object_detection.utils import label_map_util
from object_detection.utils import visualization.utils as viz_utils

PATH_TO_SAVED_MODEL = PATH_TO_MODEL_DIR + " /saved_model"

print('loading_model...' , end='')
start_time=time.time()

detect_fn = tf.saved_model.load(PATH_TO_SAVED_MODEL)

end_time = time.time()
elapsed_time = end_time - start_time
print('Done! took{} seconds'.format(elapsed_time))

category_index = label_map_util.create_category_index_from_label_map(PA
TH_TO_LABELS,
                                                                    us
e_display_name = True)

import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
def load_image_into_numpy_array(path):
    """Load an image from file into a numpy array
    Puts image into numpy array to feed into tensorflowgraph
    Note that by convention we put in into a numpy array with shape
    (height,width,channels), where channels = 3 for RGB
    """
    return np.array(Image.open(path))

print('Running inference for{}...'.format(IMAGE_PATHS),end = '')

image = cv2.imread(IMAGE_PATHS)
image_rgb = cv2.cvtColor(image,cv2.COLOR_BGR2RGB)
image_expanded = np.expand_dims(image_rgb,axis = 0)

```

```

input_tensor = tf.convert_to_tensor(image)
input_tensor = input_tensor[tf.newaxis,...]

detections = detect_fn(input_tensor)

num_detections = int(detection.pop('num_detections'))
detections = {key:value[0,:num_detections].numpy()
               for key,value in detections.items()}
detections['num_detections']= num_detections

detections['detection_classes']= detections['detection_classes'].astype
(np.int64)

image_with_detections = image.copy()

viz_utils.visualize_boxes_and_labels_on_image_array(
    image_with_detections,
    detections['detection_boxes'],
    detections['detections_classes'],
    detections['detections_scores'],
    category_index,
    use_normalized_coordinates= True,
    max_boxes_to_draw = 200,
    min_score_thresh = 0.5,
    agnostic_mode = False)
print('done')

cv2_imshow(image_with_detections)

```