**Use TensorFlow to detect and recognize objects**

!pip install -U --pre tensorflow=="2.\*"

!pip install pycocotools

|  |  |  |
| --- | --- | --- |
| import os | | |
|  | | | import pathlib | | | | | | | | | | | | | | | | | |
|  | | |  | | | | | | | | | | | | | | | | | |
|  | | | if "models" in pathlib.Path.cwd().parts: | | | | | | | | | | | | | | | | | |
|  | | | while "models" in pathlib.Path.cwd().parts: | | | | | | | | | | | | | | | | | |
|  | | | os.chdir('..') | | | | | | | | | | | | | | | | | |
|  | | | elif not pathlib.Path('models').exists(): | | | | | | | | | | | | | | | | | |
|  | | | !git clone --depth 1 https://github.com/tensorflow/models | | | | | | | | | | | | | | | | | |
| %%bash | | |
|  | | | cd models/research/ | | | | | | | | | | | | | | | | | |
|  | | | protoc object\_detection/protos/\*.proto --python\_out=. | | | | | | | | | | | | | | | | | |
| %%bash | |
|  | | cd models/research | | | | | | | | | | | | |
|  | | pip install | | | | | | | | | | | | |
| 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 | | | | | | | | | | | | | | | | | | |
|  | from matplotlib import pyplot as plt | | | | | | | | | | | | | | | | | | |
|  | from PIL import Image | | | | | | | | | | | | | | | | | | |
|  | from IPython.display import display | | | | | | | | | | | | | | | | | | |
| 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 | | | | | | | | |
| def load\_model(model\_name): | | | | | |
|  | | | | | | base\_url = 'http://download.tensorflow.org/models/object\_detection/' | | | | | | | | | | | | | | | |
|  | | | | | | model\_file = model\_name + '.tar.gz' | | | | | | | | | | | | | | | |
|  | | | | | | model\_dir = tf.keras.utils.get\_file( | | | | | | | | | | | | | | | |
|  | | | | | | fname=model\_name, | | | | | | | | | | | | | | | |
|  | | | | | | origin=base\_url + model\_file, | | | | | | | | | | | | | | | |
|  | | | | | | untar=True) | | | | | | | | | | | | | | | |
|  | | | | | |  | | | | | | | | | | | | | | | |
|  | | | | | | model\_dir = pathlib.Path(model\_dir)/"saved\_model" | | | | | | | | | | | | | | | |
|  | | | | | |  | | | | | | | | | | | | | | | |
|  | | | | | | model = tf.saved\_model.load(str(model\_dir)) | | | | | | | | | | | | | | | |
|  | | | | | | model = model.signatures['serving\_default'] | | | | | | | | | | | | | | | |
|  | | | | | |  | | | | | | | | | | | | | | | |
|  | | | | | | return model | | | | | | | | | | | | | | | |
| PATH\_TO\_LABELS = 'models/research/object\_detection/data/mscoco\_label\_map.pbtxt' | | | | | | | | | | | |
|  | | | | | | | | | | | | category\_index = label\_map\_util.create\_category\_index\_from\_labelmap(PATH\_TO\_LABELS, use\_display\_name=True) | | | | | | | | | | |
| PATH\_TO\_TEST\_IMAGES\_DIR = pathlib.Path('models/research/object\_detection/test\_images') | | | | | | | | | | | | |
|  | | | | | | | | | | | | | TEST\_IMAGE\_PATHS = sorted(list(PATH\_TO\_TEST\_IMAGES\_DIR.glob("\*.jpg"))) | | | | | | | | | |
|  | | | | | | | | | | | | | TEST\_IMAGE\_PATHS | | | | | | | | | |
| model\_name = 'ssd\_mobilenet\_v1\_coco\_2017\_11\_17' | | | | | | | | | |
|  | | | | | | | | | | detection\_model = load\_model(model\_name) | | | | | |
| model\_name = 'ssd\_mobilenet\_v1\_coco\_2017\_11\_17' | | | | | | | | |
|  | | | | | | | | | detection\_model = load\_model(model\_name) | | | | | | | | |
| def run\_inference\_for\_single\_image(model, image): | | | | | | | | | | |
|  | | | | | | | | | | | image = np.asarray(image) | | | | | | | | | | | |
|  | | | | | | | | | | | # The input needs to be a tensor, convert it using `tf.convert\_to\_tensor`. | | | | | | | | | | | |
|  | | | | | | | | | | | input\_tensor = tf.convert\_to\_tensor(image) | | | | | | | | | | | |
|  | | | | | | | | | | | # The model expects a batch of images, so add an axis with `tf.newaxis`. | | | | | | | | | | | |
|  | | | | | | | | | | | input\_tensor = input\_tensor[tf.newaxis,...] | | | | | | | | | | | |
|  | | | | | | | | | | |  | | | | | | | | | | | |
|  | | | | | | | | | | | # Run inference | | | | | | | | | | | |
|  | | | | | | | | | | | output\_dict = model(input\_tensor) | | | | | | | | | | | |
|  | | | | | | | | | | |  | | | | | | | | | | | |
|  | | | | | | | | | | | # All outputs are batches tensors. | | | | | | | | | | | |
|  | | | | | | | | | | | # Convert to numpy arrays, and take index [0] to remove the batch dimension. | | | | | | | | | | | |
|  | | | | | | | | | | | # We're only interested in the first num\_detections. | | | | | | | | | | | |
|  | | | | | | | | | | | num\_detections = int(output\_dict.pop('num\_detections')) | | | | | | | | | | | |
|  | | | | | | | | | | | output\_dict = {key:value[0, :num\_detections].numpy() | | | | | | | | | | | |
|  | | | | | | | | | | | for key,value in output\_dict.items()} | | | | | | | | | | | |
|  | | | | | | | | | | | output\_dict['num\_detections'] = num\_detections | | | | | | | | | | | |
|  | | | | | | | | | | |  | | | | | | | | | | | |
|  | | | | | | | | | | | # detection\_classes should be ints. | | | | | | | | | | | |
|  | | | | | | | | | | | output\_dict['detection\_classes'] = output\_dict['detection\_classes'].astype(np.int64) | | | | | | | | | | | |
|  | | | | | | | | | | |  | | | | | | | | | | | |
|  | | | | | | | | | | | # Handle models with masks: | | | | | | | | | | | |
|  | | | | | | | | | | | if 'detection\_masks' in output\_dict: | | | | | | | | | | | |
|  | | | | | | | | | | | # Reframe the the bbox mask to the image size. | | | | | | | | | | | |
|  | | | | | | | | | | | detection\_masks\_reframed = utils\_ops.reframe\_box\_masks\_to\_image\_masks( | | | | | | | | | | | |
|  | | | | | | | | | | | output\_dict['detection\_masks'], output\_dict['detection\_boxes'], | | | | | | | | | | | |
|  | | | | | | | | | | | image.shape[0], image.shape[1]) | | | | | | | | | | | |
|  | | | | | | | | | | | detection\_masks\_reframed = tf.cast(detection\_masks\_reframed > 0.5, | | | | | | | | | | | |
|  | | | | | | | | | | | tf.uint8) | | | | | | | | | | | |
|  | | | | | | | | | | | output\_dict['detection\_masks\_reframed'] = detection\_masks\_reframed.numpy() | | | | | | | | | | | |
|  | | | | | | | | | | |  | | | | | | | | | | | |
|  | | | | | | | | | | | return output\_dict | | | | | | | | | | | |
| def show\_inference(model, 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 = np.array(Image.open(image\_path)) | | | | | | | | | | | | | | | | | |
|  | | | | | # Actual detection. | | | | | | | | | | | | | | | | | |
|  | | | | | output\_dict = run\_inference\_for\_single\_image(model, image\_np) | | | | | | | | | | | | | | | | | |
|  | | | | | # 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\_reframed', None), | | | | | | | | | | | | | | | | | |
|  | | | | | use\_normalized\_coordinates=True, | | | | | | | | | | | | | | | | | |
|  | | | | | line\_thickness=8) | | | | | | | | | | | | | | | | | |
|  | | | | |  | | | | | | | | | | | | | | | | | |
|  | | | | | display(Image.fromarray(image\_np)) | | | | | | | | | | | | | | | | | |
| for image\_path in TEST\_IMAGE\_PATHS: | | | |
|  | | | | show\_inference(detection\_model, image\_path) | | | | | | | | | | |
| model\_name = 'ssd\_inception\_v1\_coco\_2017\_11\_17' | | | | | | | |
|  | | | | | | | | detection\_model = load\_model(model\_name) | | | | | | | | | | |
| model\_name = 'faster\_rcnn\_resnet101\_coco' | | | | | | |
|  | | | | | | | detection\_model = load\_model(model\_name) | | | | | | | | | |