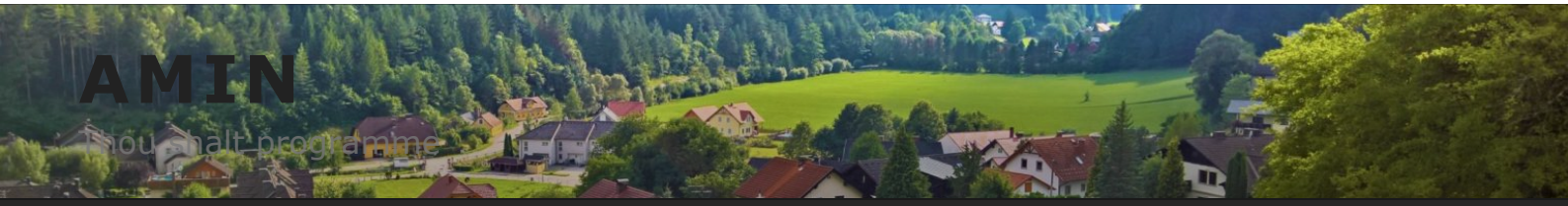


AMIN

Thou shalt programme

2020-03-07 BY AMIN

```
FROM tensorflow/tensorflow:1.15.2-py3-jupyter
RUN apt-get update
RUN apt-get upgrade -y
RUN apt-get install git cmake wget unzip -y
ADD opencv.sh /home/opencv.sh
RUN /home/opencv.sh
RUN pip install Cython
RUN pip install contextlib2
RUN pip install pillow
RUN pip install lxml
ADD tfmodels.sh /home/tfmodels.sh
RUN /home/tfmodels.sh
```

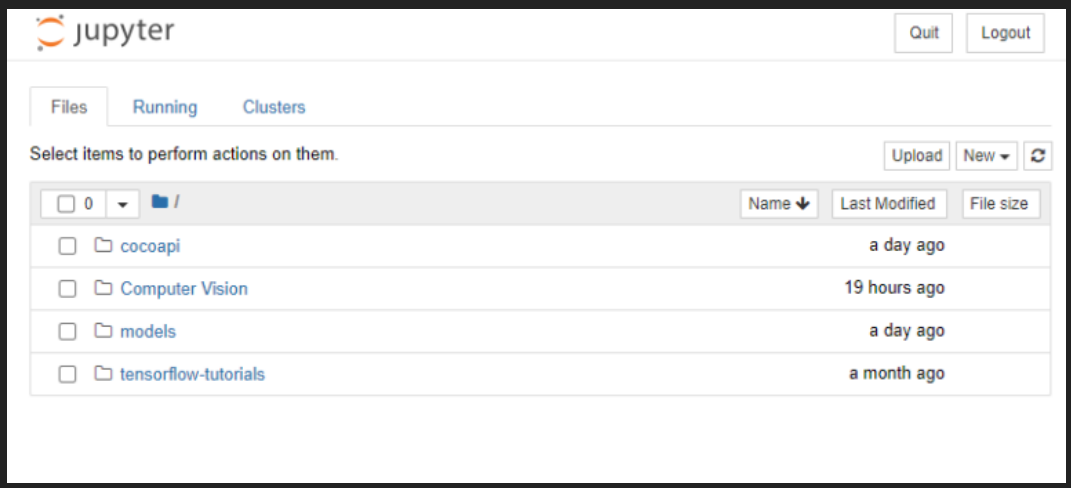
```
cd /tf/
git clone https://github.com/tensorflow/models.git --single-branch v1.13.0
mv v1.13.0 models
git clone https://github.com/cocodataset/cocoapi.git
cd cocoapi/PythonAPI
make
cp -r pycocotools /tf/models/research/
cd /tf/models/research
wget -O protobuf.zip https://github.com/google/protobuf/releases/download/v3.0.0/protoc
unzip protobuf.zip
./bin/protoc object_detection/protos/*.proto --python_out=.
export PYTHONPATH=$PYTHONPATH:`pwd`:`pwd`/slim
python object_detection/builders/model_builder_test.py
```

```
cd /
git clone https://github.com/opencv/opencv.git --single-branch 3.4.9
mv 3.4.9 cv
cd cv
mkdir build
cd build
cmake ..
make
make install
```

```
docker build --tag computervision .
```

```
docker run -p 8888:8888 computervision
```

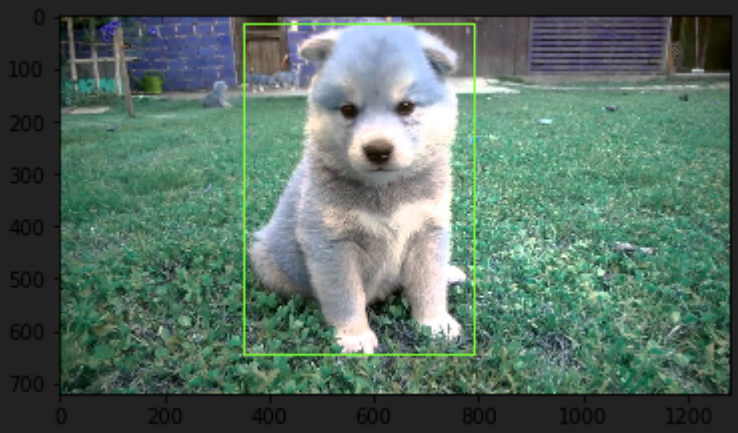
```
http://127.0.0.1:8888/?token=487a8ed5ac0cb13b0a57080917cc679db9242180a5181a52
```




```
!wget "http://download.tensorflow.org/models/object_detection/ssd_mobilenet_v2_coco_201
!tar -xvzf "ssd_mobilenet_v2_coco_2018_03_29.tar.gz"
!wget "https://somewebsite.com/test.jpg"
```

```
import numpy as np
import tensorflow as tf
import cv2 as cv

# Read the graph.
with tf.gfile.FastGFile('ssd_mobilenet_v2_coco_2018_03_29/frozen_inference_graph.pb', '
    graph_def = tf.GraphDef()
    graph_def.ParseFromString(f.read())

with tf.Session() as sess:
    # Restore session
    sess.graph.as_default()
    tf.import_graph_def(graph_def, name='')

    # Read and preprocess an image.
    img = cv.imread('test.jpg')
    rows = img.shape[0]
    cols = img.shape[1]
    inp = cv.resize(img, (300, 300))
    inp = inp[:, :, [2, 1, 0]]  # BGR2RGB

    # Run the model
    out = sess.run([sess.graph.get_tensor_by_name('num_detections:0'),
                    sess.graph.get_tensor_by_name('detection_scores:0'),
                    sess.graph.get_tensor_by_name('detection_boxes:0'),
                    sess.graph.get_tensor_by_name('detection_classes:0')],
                   feed_dict={'image_tensor:0': inp.reshape(1, inp.shape[0], inp.shape[

    # Visualize detected bounding boxes.
    num_detections = int(out[0][0])
    for i in range(num_detections):
        classId = int(out[3][0][i])
        score = float(out[1][0][i])
        bbox = [float(v) for v in out[2][0][i]]
        if score > 0.3:
            x = bbox[1] * cols
            y = bbox[0] * rows
            right = bbox[3] * cols
            bottom = bbox[2] * rows
            cv.rectangle(img, (int(x), int(y)), (int(right), int(bottom)), (125, 255, 5
```

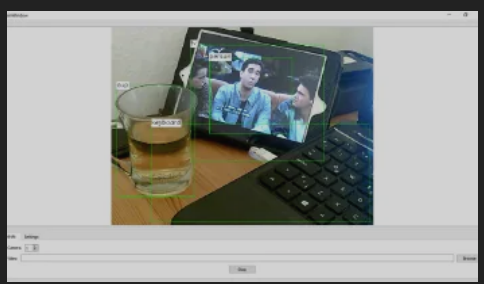
```
import matplotlib.pyplot as plt
plt.imshow(img)
```



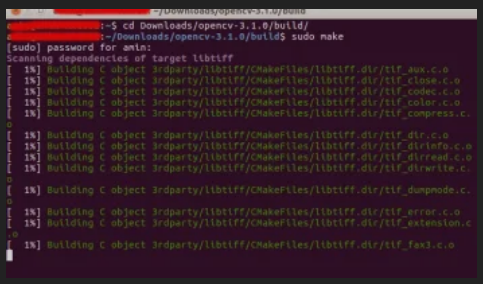
Share this:

WhatsApp Email Print LinkedIn Reddit Facebook Twitter Pinterest Pocket Skype Tumblr Telegram

Related




Hands-On
Algorithms for
Computer Vision
Computer Vision
with OpenCV 3
and Qt5


:~$ cd Downloads/opencv-3.1.0/build/
:~/Downloads/opencv-3.1.0/build$ sudo make
[sudo] password for anln:
Scanning dependencies of target libtiff
[1%] Building C object 3rdparty/libtiff/CMakeFiles/libtiff.dir/tif_aux.c.o
[1%] Building C object 3rdparty/libtiff/CMakeFiles/libtiff.dir/tif_close.c.o
[1%] Building C object 3rdparty/libtiff/CMakeFiles/libtiff.dir/tif_codec.c.o
[1%] Building C object 3rdparty/libtiff/CMakeFiles/libtiff.dir/tif_color.c.o
[1%] Building C object 3rdparty/libtiff/CMakeFiles/libtiff.dir/tif_compress.c.
o
[1%] Building C object 3rdparty/libtiff/CMakeFiles/libtiff.dir/tif_dir.c.o
[1%] Building C object 3rdparty/libtiff/CMakeFiles/libtiff.dir/tif_dirinfo.c.o
[1%] Building C object 3rdparty/libtiff/CMakeFiles/libtiff.dir/tif_dirread.c.o
[1%] Building C object 3rdparty/libtiff/CMakeFiles/libtiff.dir/tif_dirwrite.c.
o
[1%] Building C object 3rdparty/libtiff/CMakeFiles/libtiff.dir/tif_dumpmode.c.
o
[1%] Building C object 3rdparty/libtiff/CMakeFiles/libtiff.dir/tif_error.c.o
[1%] Building C object 3rdparty/libtiff/CMakeFiles/libtiff.dir/tif_extension.c
.o
[1%] Building C object 3rdparty/libtiff/CMakeFiles/libtiff.dir/tif_fax3.c.o

/