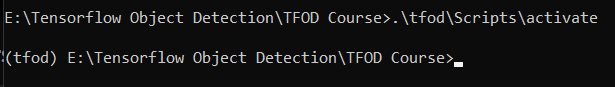


Made a virtual env using command prompt named tfod.

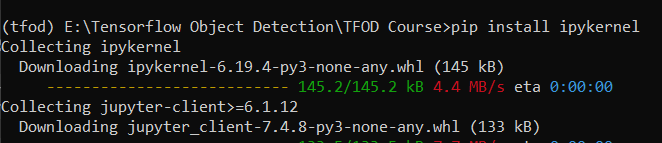


Activated the virtual env. : **.\tfod\Scripts\activate**

(if we need to get out of the env, just type ‘deactivate’ )



Updating pip : **python -m pip install --upgrade pip**

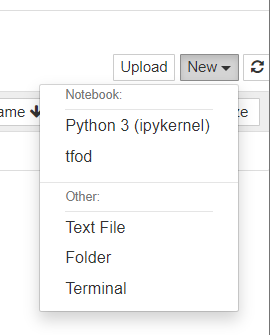


Pip install ipykernel – for jupyter notebook



Installing the virtual env into jupyter nb : **python -m ipykernel install --user --name=tfod**



 the virtual env tfod is displayed .

**LABELLING THE IMAGE**

Clone the github files from (installing the application) : <https://github.com/heartexlabs/labelImg>

**FOLDERS**:

1. TFOD Course >> made a virtual env named ‘tfod’ >> started a jupyter nb using that venv
2. ( for the collection of images)

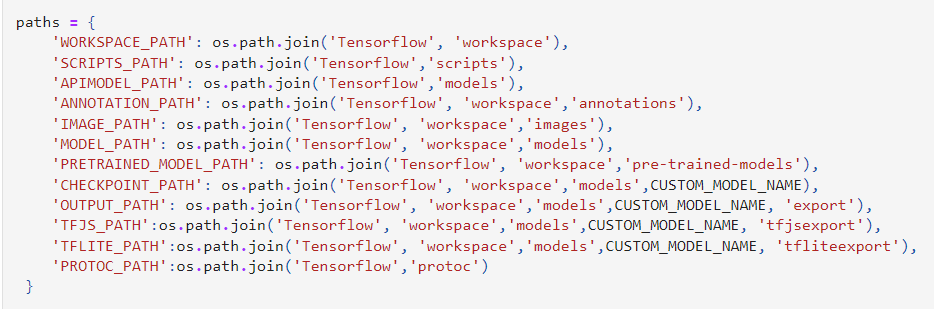
TFOD Course >> Tensorflow >> workspace >> images >> collectedimages >> *names of classes*

1. ( to download and install the github files and app of LabelImage)

Tensorflow >> LabelImg

1. ( for separating training and testing images)

images >> train & test folders

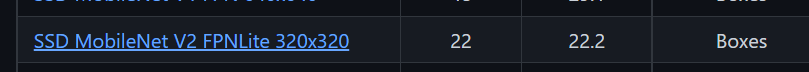


1. Then we make all the above paths in the code itself.

**INSTALLING MODELS**

**Tensorflow model zoo tf2 link:** <https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md>

**Model link that we will be using:**



<http://download.tensorflow.org/models/object_detection/tf2/20200711/ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8.tar.gz>

**Cloning a tf obj detection repo in Tensorflow >> models (APIMODEL\_PATH)** : ( TF Model Garden)

<https://github.com/tensorflow/models>

in the above link for object detection, we only need files of : select research >> object\_detection

* Download and install the models from tf model garden into:

Tensorflow >> models ie. APIMODEL\_PATH

**protoc : Protocol Buffers**

**To check what version you need to download for Tensorflow, cuda, etc**

<https://www.tensorflow.org/install/source_windows>

**TF RECORDS :**

It is a binary file format for storing data. Using a TF Record helps speed up training for our custom object detection model