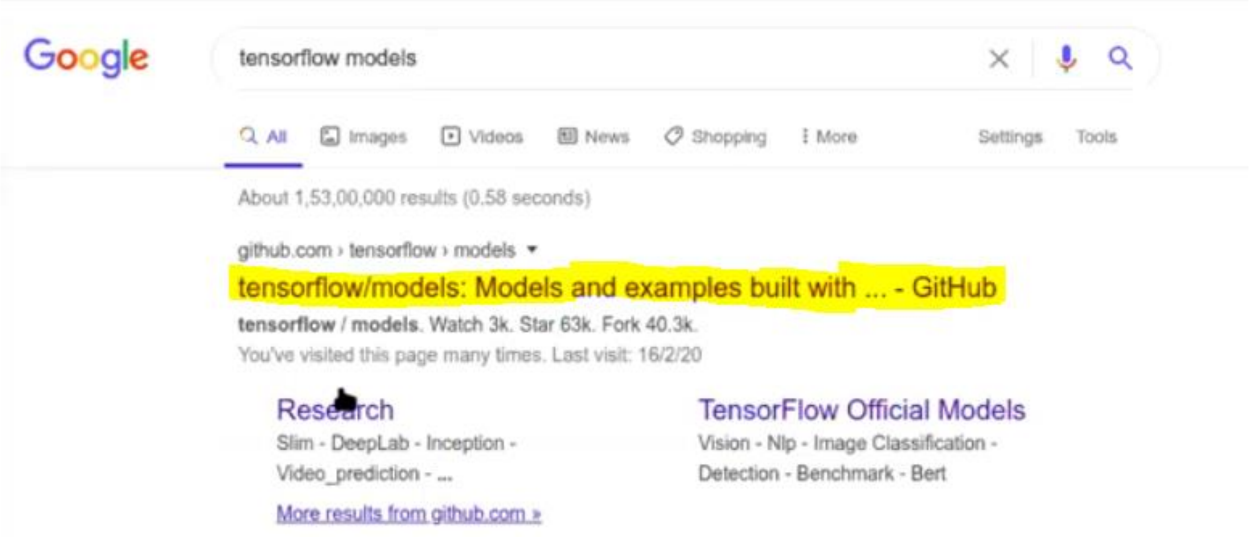
TFOD(TensorFlow Object Detection) SETUP:

A. Downloading files.
B. Installation tensor flow 1.14 – Create a virtual environment
C. Setup verification and Installation
D. Custom training data
   1. Annotation/Labelling
   2. Custom training process
   3. Xml >> csv and csv >> tfrecords

A. Downloading part

Step 1: Downloading files from the internet

Search in google: TensorFlow models



About 1,53,00,000 results (0.58 seconds)

github.com › tensorflow › models ▾

tensorflow/models: Models and examples built with ... - GitHub

tensorflow / **models**. Watch 3k. Star 63k. Fork 40.3k.

You've visited this page many times. Last visit: 16/2/20

Research
Slim - DeepLab - Inception - Video_prediction - ...

TensorFlow Official Models
Vision - Nlp - Image Classification - Detection - Benchmark - Bert

More results from github.com »

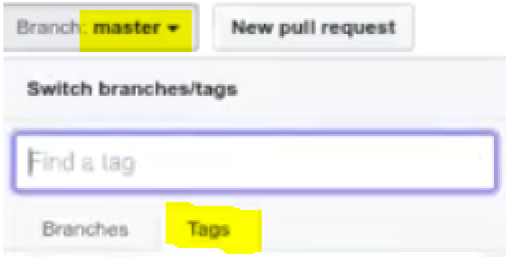
Official GitHub links for TF - https://github.com/tensorflow/models/tree/master/research/object_detection

To download TF models - https://github.com/tensorflow/models/archive/v1.13.0.zip

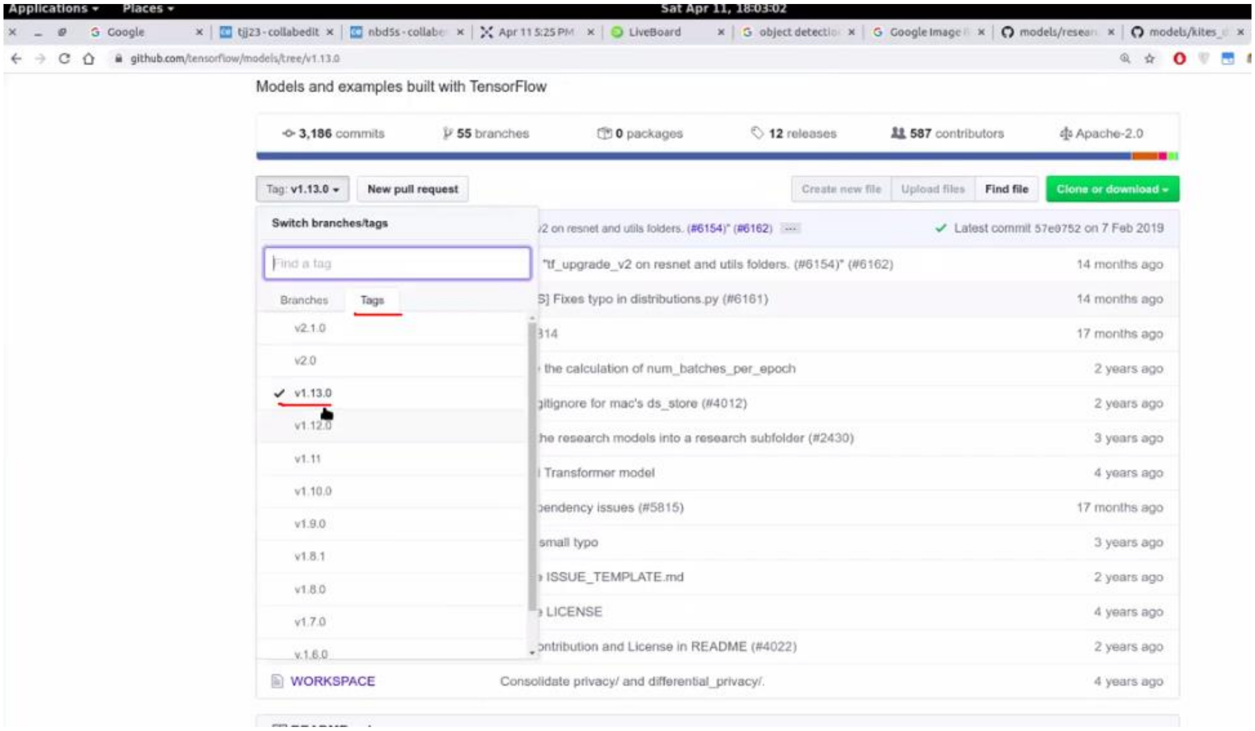
If you open the above link then automatically download starts.

Lets suppose if you want the specific TensorFlow version follow the below steps.

Click **Master** and change in to **Tags**



Then select v13.0 – stable version



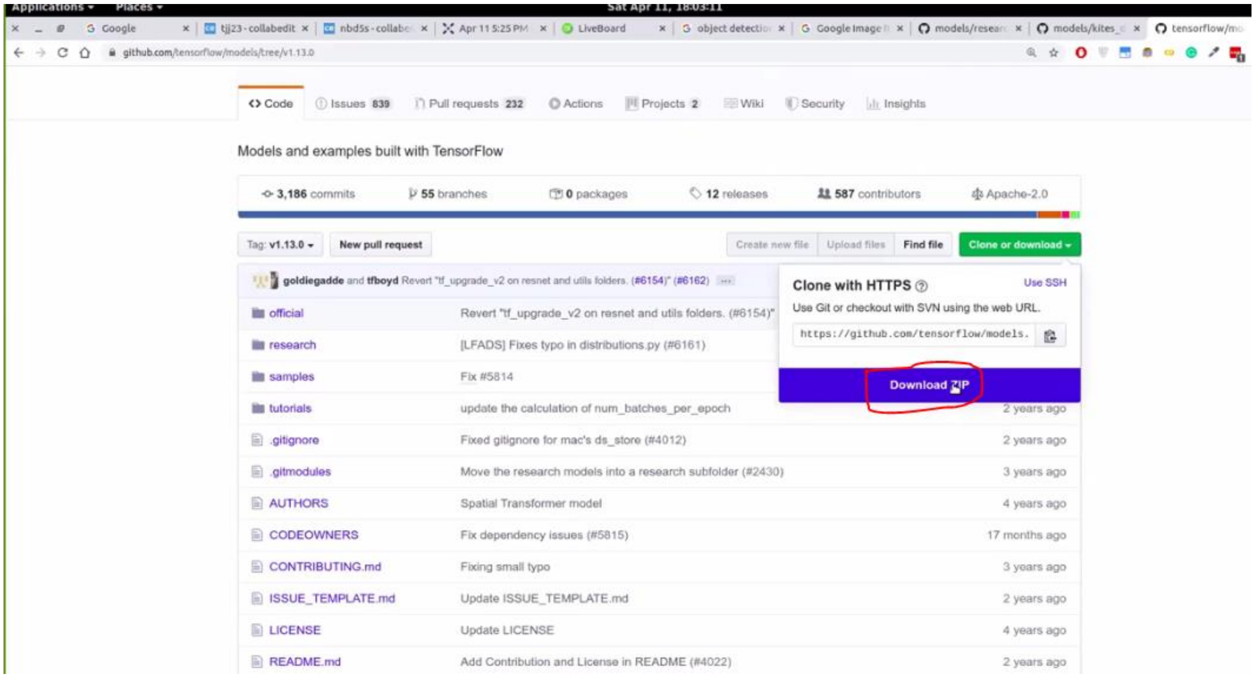
Models and examples built with TensorFlow

/2 on resnet and utils folders. (#6154)" (#6162) ··· ✓ Latest commit 57e0752 on 7 Feb 2019

"tf_upgrade_v2 on resnet and utils folders. (#6154)" (#6162)	14 months ago
S] Fixes typo in distributions.py (#6161)	14 months ago
314	17 months ago
the calculation of num_batches_per_epoch	2 years ago
jitignore for mac's ds_store (#4012)	2 years ago
he research models into a research subfolder (#2430)	3 years ago
Transformer model	4 years ago
pendency issues (#5815)	17 months ago
small typo	3 years ago
ISSUE_TEMPLATE.md	2 years ago
LICENSE	4 years ago
ontribution and License in README (#4022)	2 years ago
WORKSPACE Consolidate privacy/ and differential_privacy/.	4 years ago

Then download the file (v.13.0)

Why we have to download – Some of the files are provided by the TensorFlow community and we are going to use them.



github.com/tensorflow/models/tree/v1.13.0

Models and examples built with TensorFlow

3,186 commits · 55 branches · 0 packages · 12 releases · 587 contributors · Apache-2.0

goldiegadde and tfboyd Revert "tf_upgrade_v2 on resnet and utils folders. (#6154)" (#6162)

Clone with HTTPS

Use SSH

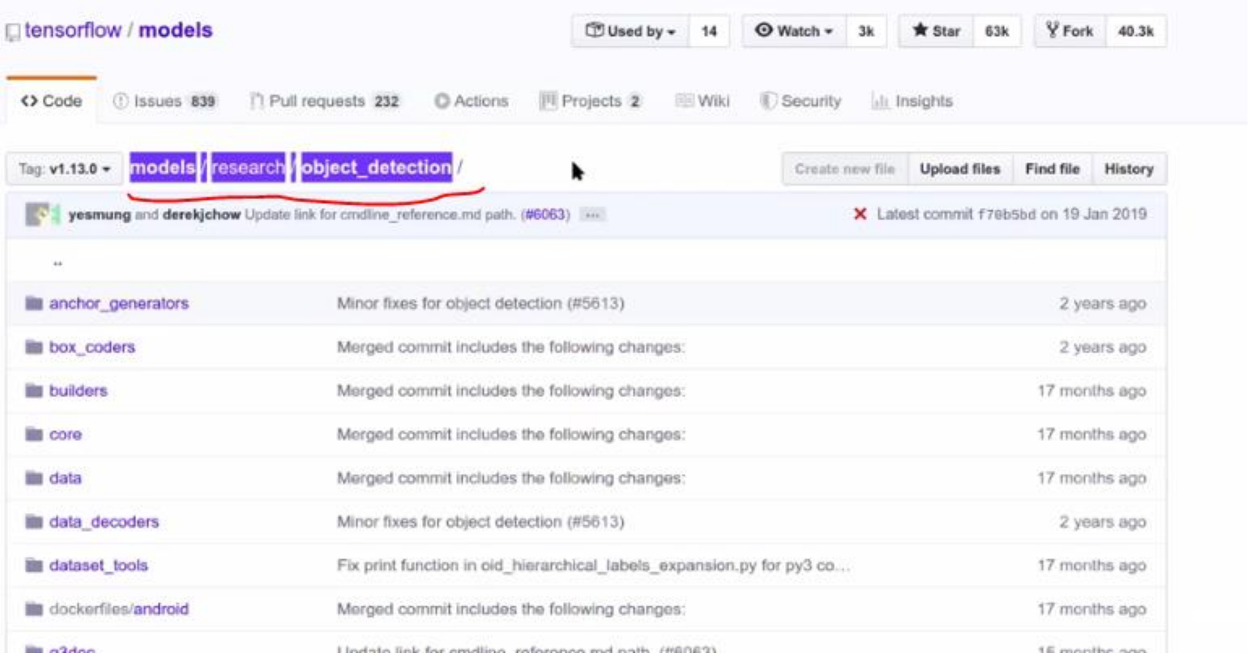
Use Git or checkout with SVN using the web URL.

```
https://github.com/tensorflow/models.
```

official	Revert "tf_upgrade_v2 on resnet and utils folders. (#6154)"	
research	[LFADS] Fixes typo in distributions.py (#6161)	
samples	Fix #5814	
tutorials	update the calculation of num_batches_per_epoch	2 years ago
.gitignore	Fixed gitignore for mac's ds_store (#4012)	2 years ago
.gitmodules	Move the research models into a research subfolder (#2430)	3 years ago
AUTHORS	Spatial Transformer model	4 years ago
CODEOWNERS	Fix dependency issues (#5815)	17 months ago
CONTRIBUTING.md	Fixing small typo	3 years ago
ISSUE_TEMPLATE.md	Update ISSUE_TEMPLATE.md	2 years ago
LICENSE	Update LICENSE	4 years ago
README.md	Add Contribution and License in README (#4022)	2 years ago

With respect to TensorFlow object detection, most important folder is “Research Folder”

models -- > research folder -- > Object detection -- >

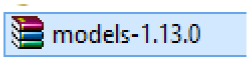

tensorflow / models

models / research / object_detection /

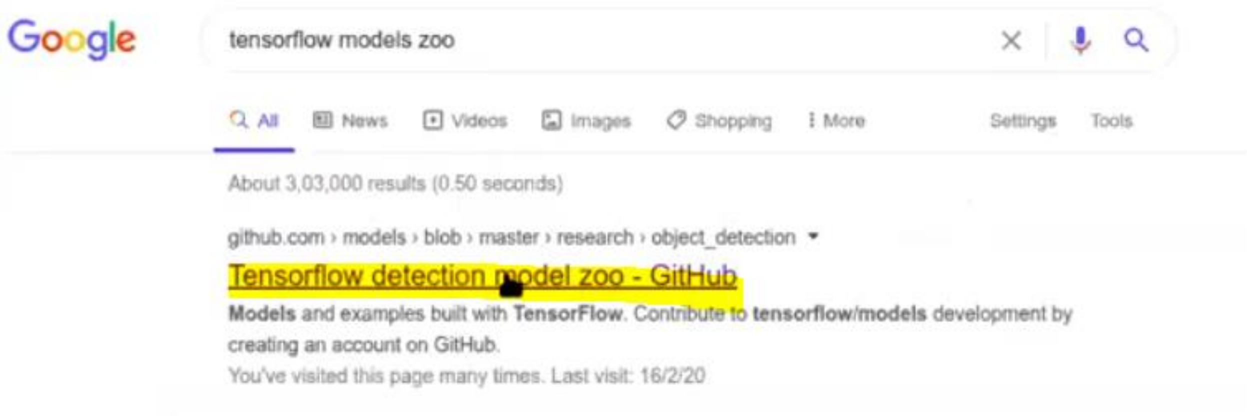
yesmung and derekjchow Update link for cmdline_reference.md path. (#6063) ... ✕ Latest commit f70b5bd on 19 Jan 2019

..		
anchor_generators	Minor fixes for object detection (#5613)	2 years ago
box_coders	Merged commit includes the following changes:	2 years ago
builders	Merged commit includes the following changes:	17 months ago
core	Merged commit includes the following changes:	17 months ago
data	Merged commit includes the following changes:	17 months ago
data_decoders	Minor fixes for object detection (#5613)	2 years ago
dataset_tools	Fix print function in oid_hierarchical_labels_expansion.py for py3 co…	17 months ago
dockerfiles/android	Merged commit includes the following changes:	17 months ago
g3doc	Update link for cmdline_reference.md path. (#6063)	15 months ago

The file looks like below after downloaded:


models-1.13.0

2. Now we are going to download the pre-trained models (Faster_rcnn_inception_V2) .And below is the link to download.



tensorflow models zoo

About 3,03,000 results (0.50 seconds)

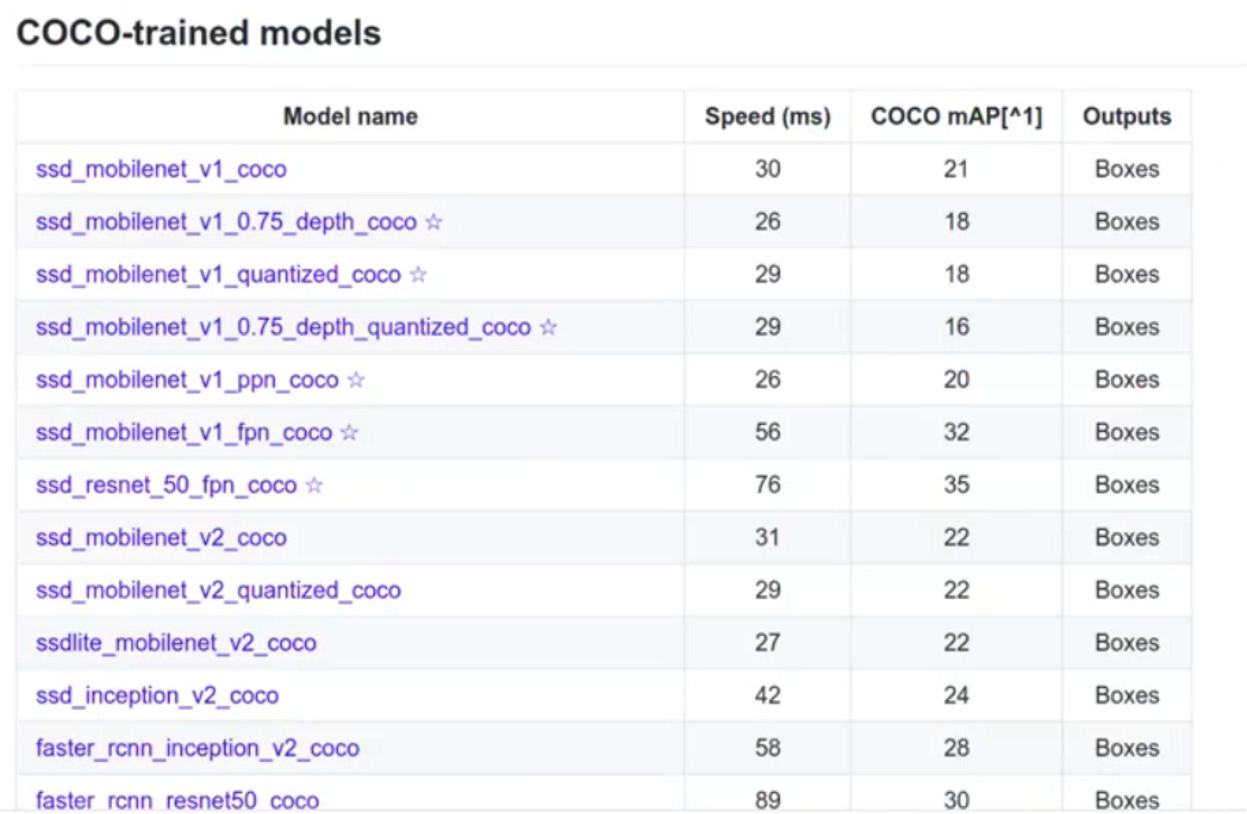
github.com › models › blob › master › research › object_detection

Tensorflow detection model zoo - GitHub

Models and examples built with **TensorFlow**. Contribute to **tensorflow/models** development by creating an account on GitHub.

You've visited this page many times. Last visit: 16/2/20

Then download the pre trained models you want



COCO-trained models

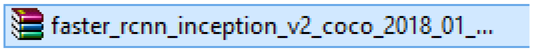
Model name	Speed (ms)	COCO mAP[^1]	Outputs
ssd_mobilenet_v1_coco	30	21	Boxes
ssd_mobilenet_v1_0.75_depth_coco ☆	26	18	Boxes
ssd_mobilenet_v1_quantized_coco ☆	29	18	Boxes
ssd_mobilenet_v1_0.75_depth_quantized_coco ☆	29	16	Boxes
ssd_mobilenet_v1_ppn_coco ☆	26	20	Boxes
ssd_mobilenet_v1_fpn_coco ☆	56	32	Boxes
ssd_resnet_50_fpn_coco ☆	76	35	Boxes
ssd_mobilenet_v2_coco	31	22	Boxes
ssd_mobilenet_v2_quantized_coco	29	22	Boxes
ssdlite_mobilenet_v2_coco	27	22	Boxes
ssd_inception_v2_coco	42	24	Boxes
faster_rcnn_inception_v2_coco	58	28	Boxes
faster_rcnn_resnet50_coco	89	30	Boxes

Here we have multiple number of models and they have trained on different types of datasets with different types of CNN architecture and various versions are also available.

Below is the direct link to download the pre- trained model:

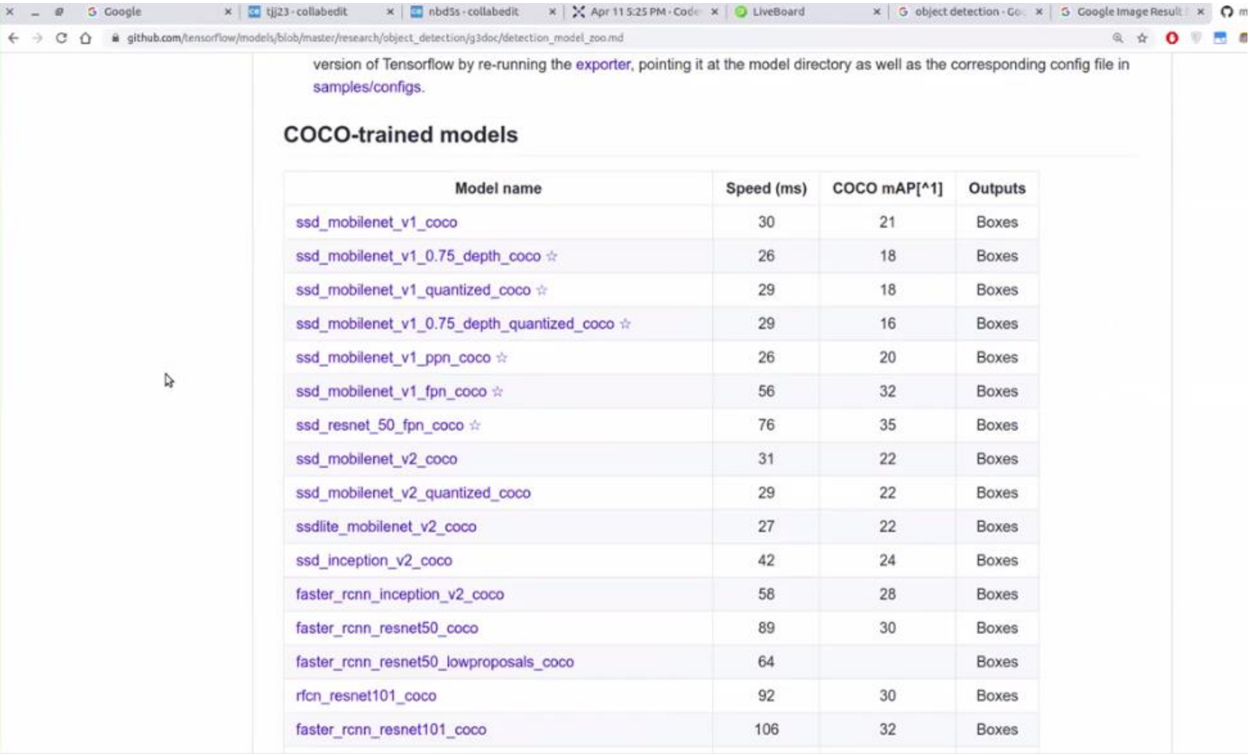
http://download.tensorflow.org/models/object_detection/faster_rcnn_inception_v2_coco_2018_01_28.tar.gz

After the file downloaded it looks like below.



faster_rcnn_inception_v2_coco_2018_01_...

In this coco models different types of datasets are already trained with the help of CNN architecture



version of Tensorflow by re-running the exporter, pointing it at the model directory as well as the corresponding config file in samples/configs.

COCO-trained models

Model name	Speed (ms)	COCO mAP[^1]	Outputs
ssd_mobilenet_v1_coco	30	21	Boxes
ssd_mobilenet_v1_0.75_depth_coco ☆	26	18	Boxes
ssd_mobilenet_v1_quantized_coco ☆	29	18	Boxes
ssd_mobilenet_v1_0.75_depth_quantized_coco ☆	29	16	Boxes
ssd_mobilenet_v1_ppn_coco ☆	26	20	Boxes
ssd_mobilenet_v1_fpn_coco ☆	56	32	Boxes
ssd_resnet_50_fpn_coco ☆	76	35	Boxes
ssd_mobilenet_v2_coco	31	22	Boxes
ssd_mobilenet_v2_quantized_coco	29	22	Boxes
ssdlite_mobilenet_v2_coco	27	22	Boxes
ssd_inception_v2_coco	42	24	Boxes
faster_rcnn_inception_v2_coco	58	28	Boxes
faster_rcnn_resnet50_coco	89	30	Boxes
faster_rcnn_resnet50_lowproposals_coco	64		Boxes
rfcn_resnet101_coco	92	30	Boxes
faster_rcnn_resnet101_coco	106	32	Boxes

Currently we using some models like

faster_rcnn_inception_v2_coco

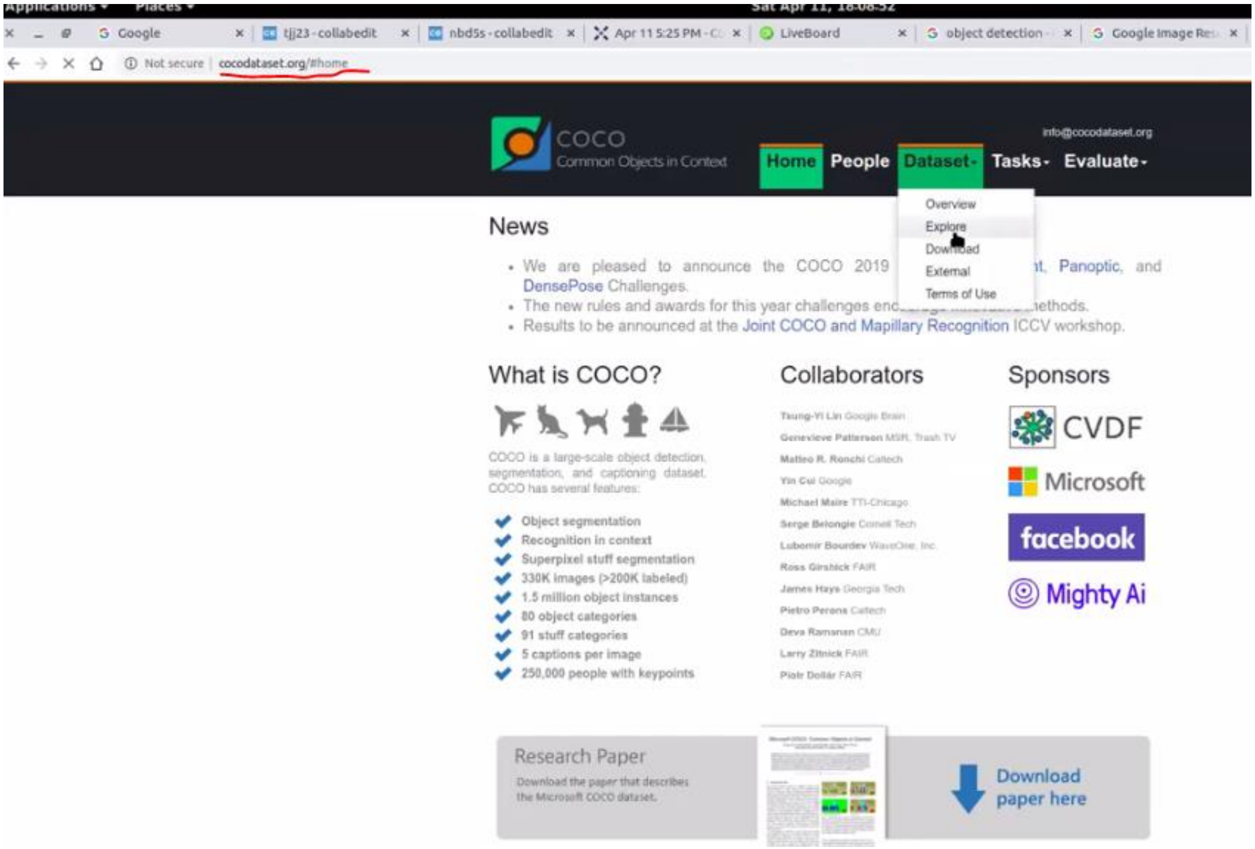
This one good for both CPU and GPU users and it is specifically used for object detection and CNN architecture is Inception which means it is the gogglenet V2 model or Inception_V2 model.

Faster_rcnn is the base for object detection model.

faster_rcnn_resnet50_coco

- Only good for GPU user's and not for CPU user's

What is coco : It is basically a dataset that the architecture has been trained (Pretrained models), This community has collected all the images and they have labelled the data.



News

- We are pleased to announce the COCO 2019 … t, Panoptic, and DensePose Challenges.
- The new rules and awards for this year challenges enc… …ethods.
- Results to be announced at the Joint COCO and Mapillary Recognition ICCV workshop.

What is COCO?

COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features:

- ✓ Object segmentation
- ✓ Recognition in context
- ✓ Superpixel stuff segmentation
- ✓ 330K images (>200K labeled)
- ✓ 1.5 million object instances
- ✓ 80 object categories
- ✓ 91 stuff categories
- ✓ 5 captions per image
- ✓ 250,000 people with keypoints

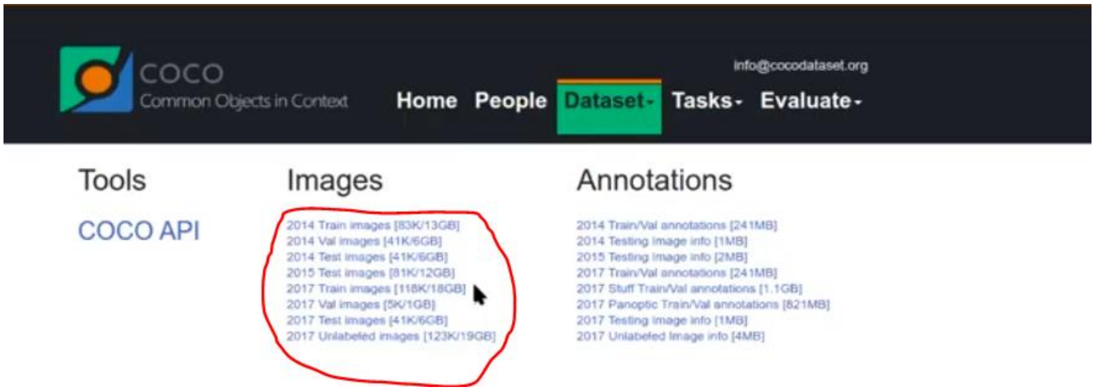
Collaborators

Tsung-Yi Lin Google Brain
Genevieve Patterson MSR, Trash TV
Matteo R. Ronchi Caltech
Yin Cui Google
Michael Maire TTI-Chicago
Serge Belongie Cornell Tech
Lubomir Bourdev WaveOne, Inc.
Ross Girshick FAIR
James Hays Georgia Tech
Pietro Perona Caltech
Deva Ramanan CMU
Larry Zitnick FAIR
Piotr Dollár FAIR

Sponsors

Research Paper

Download the paper that describes the Microsoft COCO dataset.



Tools

COCO API

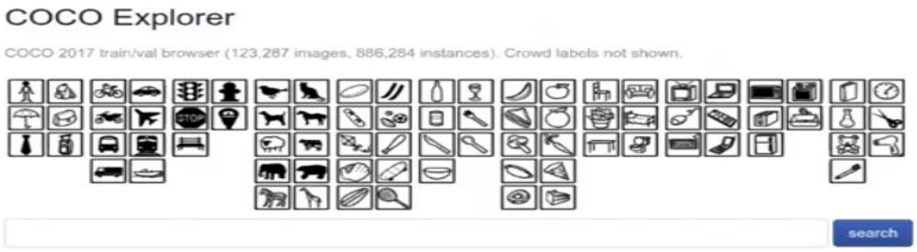
Images

2014 Train images [83K/13GB]
2014 Val images [41K/6GB]
2014 Test images [41K/6GB]
2015 Test images [81K/12GB]
2017 Train images [118K/18GB]
2017 Val images [5K/1GB]
2017 Test images [41K/6GB]
2017 Unlabeled images [123K/19GB]

Annotations

2014 Train/Val annotations [241MB]
2014 Testing image info [1MB]
2015 Testing image info [2MB]
2017 Train/Val annotations [241MB]
2017 Stuff Train/Val annotations [1.1GB]
2017 Panoptic Train/Val annotations [821MB]
2017 Testing image info [1MB]
2017 Unlabeled image info [4MB]

In this coco dataset we have 90 classes are there. On top of these 90 classes, they have trained the images with various architecture and here, we are going to reuse the model called **Faster_rcnn_Inception_v2_Coco.**

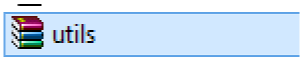

COCO Explorer

COCO 2017 train/val browser (123,287 images, 886,284 instances). Crowd labels not shown.

3. The next thing is we need to download some utility files and the link is below to download

https://drive.google.com/file/d/12F5oGAuQg7qBM_267TCMt_rlorV-M7gf/view?usp=sharing

After downloaded the file looks like below



utils

4. The next thing is we need to download the annotation tool for labeling the images and below is the link to download

https://tzutalin.github.io/labelImg/

- Windows_v1.8.0

This is the version for Window’s user.

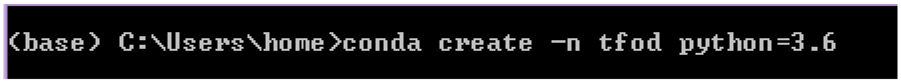
windows_v1.8.0

The file looks like above.

Here the all the downloading parts are completed.

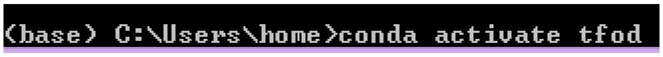
B. Create a tensorflow 1.14 environment

1. To create a new environment open the Anaconda prompt



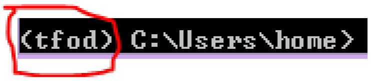
```
(base) C:\Users\home>conda create -n tfod python=3.6
```

2. Activate the environment



```
(base) C:\Users\home>conda activate tfod
```

Now the environment is activated

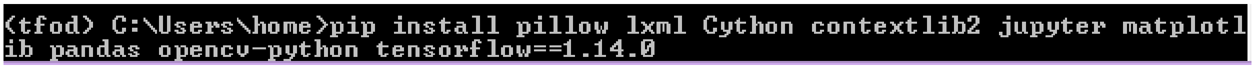


```
(tfod) C:\Users\home>
```

3. Now we need to install some of the packages

For CPU user’s only:

pip install pillow lxml Cython contextlib2 jupyter matplotlib pandas opencv-python tensorflow==1.14.0



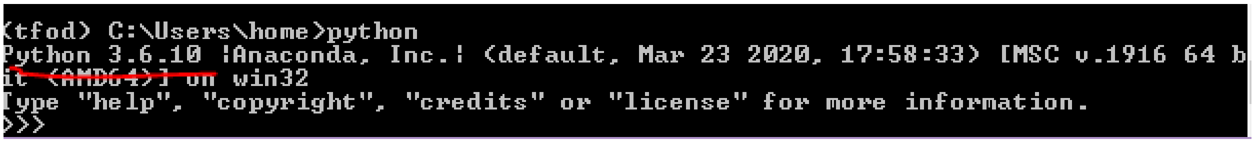
```
(tfod) C:\Users\home>pip install pillow lxml Cython contextlib2 jupyter matplotl
ib pandas opencv-python tensorflow==1.14.0
```

GPU user’s - make sure you have to install the CUDA toolkit and NVDIA toolkit.

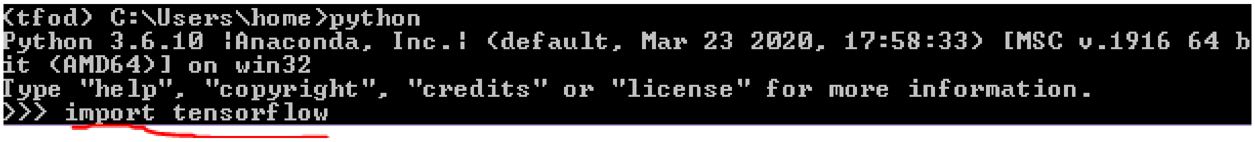
GPU User's:

pip install pillow lxml Cython contextlib2 jupyter matplotlib pandas opencv-python tensorflow-gpu==1.14.0

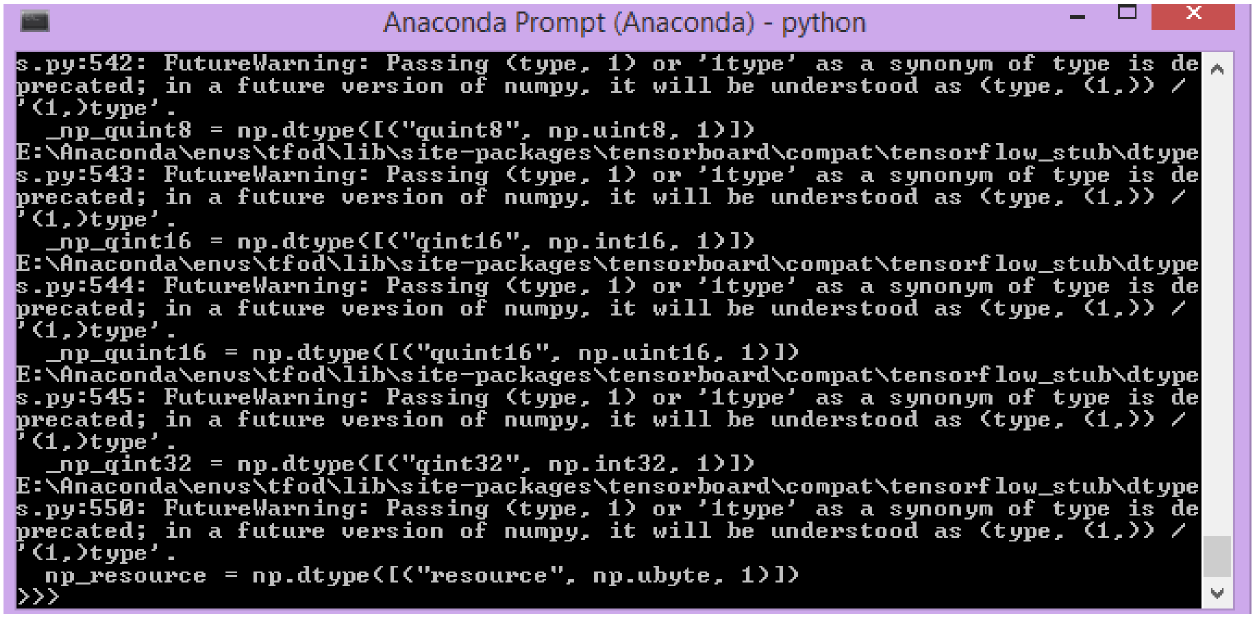
After successful installation, we need to verify. To verify just open the python shell and you will be to see the python version as 3.6



```
(tfod) C:\Users\home>python
Python 3.6.10 |Anaconda, Inc.| (default, Mar 23 2020, 17:58:33) [MSC v.1916 64 b
it (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```



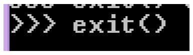
```
(tfod) C:\Users\home>python
Python 3.6.10 |Anaconda, Inc.| (default, Mar 23 2020, 17:58:33) [MSC v.1916 64 b
it (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow
```



```
s.py:542: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is de
precated; in a future version of numpy, it will be understood as (type, (1,)) /
'(1,)type'.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
E:\Anaconda\envs\tfod\lib\site-packages\tensorboard\compat\tensorflow_stub\dtype
s.py:543: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is de
precated; in a future version of numpy, it will be understood as (type, (1,)) /
'(1,)type'.
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
E:\Anaconda\envs\tfod\lib\site-packages\tensorboard\compat\tensorflow_stub\dtype
s.py:544: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is de
precated; in a future version of numpy, it will be understood as (type, (1,)) /
'(1,)type'.
  _np_quint16 = np.dtype([("quint16", np.uint16, 1)])
E:\Anaconda\envs\tfod\lib\site-packages\tensorboard\compat\tensorflow_stub\dtype
s.py:545: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is de
precated; in a future version of numpy, it will be understood as (type, (1,)) /
'(1,)type'.
  _np_qint32 = np.dtype([("qint32", np.int32, 1)])
E:\Anaconda\envs\tfod\lib\site-packages\tensorboard\compat\tensorflow_stub\dtype
s.py:550: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is de
precated; in a future version of numpy, it will be understood as (type, (1,)) /
'(1,)type'.
  np_resource = np.dtype([("resource", np.ubyte, 1)])
>>>
```

It shows some deprecated warning and apart from that everything is working fine here. If you face any problem then we need to create the new environment again and we need to do re-install the packages.

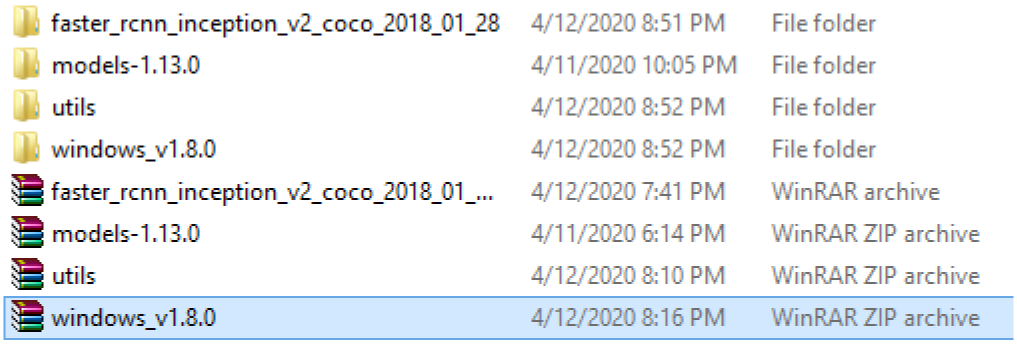
Exit from the python shell after verification



```
>>> exit()
```

Install the downloaded packages which we are downloaded early.

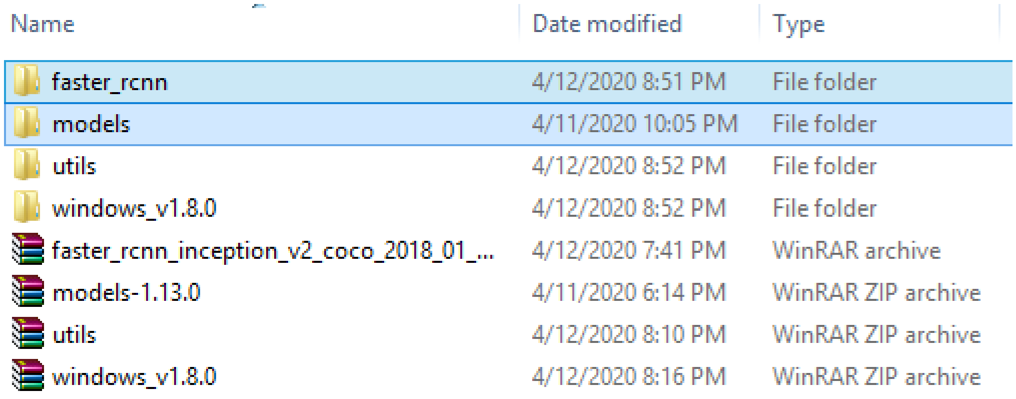
1. First unzip all the files.



faster_rcnn_inception_v2_coco_2018_01_28	4/12/2020 8:51 PM	File folder
models-1.13.0	4/11/2020 10:05 PM	File folder
utils	4/12/2020 8:52 PM	File folder
windows_v1.8.0	4/12/2020 8:52 PM	File folder
faster_rcnn_inception_v2_coco_2018_01_...	4/12/2020 7:41 PM	WinRAR archive
models-1.13.0	4/11/2020 6:14 PM	WinRAR ZIP archive
utils	4/12/2020 8:10 PM	WinRAR ZIP archive
windows_v1.8.0	4/12/2020 8:16 PM	WinRAR ZIP archive

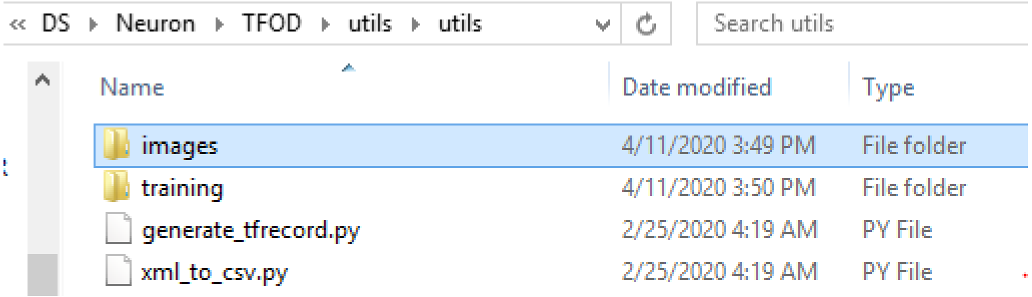
I done unzip all the files.

2. Now I'm going to rename the folder models-1.13.0 to models and faster_rcnn also



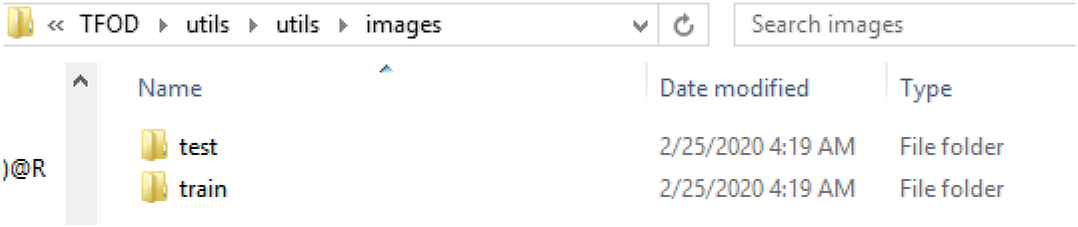
Name	Date modified	Type
faster_rcnn	4/12/2020 8:51 PM	File folder
models	4/11/2020 10:05 PM	File folder
utils	4/12/2020 8:52 PM	File folder
windows_v1.8.0	4/12/2020 8:52 PM	File folder
faster_rcnn_inception_v2_coco_2018_01_...	4/12/2020 7:41 PM	WinRAR archive
models-1.13.0	4/11/2020 6:14 PM	WinRAR ZIP archive
utils	4/12/2020 8:10 PM	WinRAR ZIP archive
windows_v1.8.0	4/12/2020 8:16 PM	WinRAR ZIP archive

In this **Utils** folder we have basically a dataset Images -- > Test and Data



« DS › Neuron › TFOD › utils › utils

Name	Date modified	Type
images	4/11/2020 3:49 PM	File folder
training	4/11/2020 3:50 PM	File folder
generate_tfrecord.py	2/25/2020 4:19 AM	PY File
xml_to_csv.py	2/25/2020 4:19 AM	PY File



« TFOD › utils › utils › images

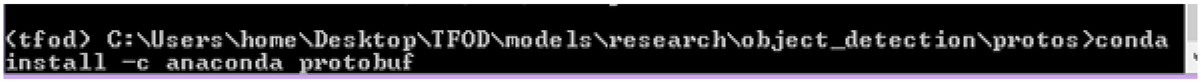
Name	Date modified	Type
test	2/25/2020 4:19 AM	File folder
train	2/25/2020 4:19 AM	File folder

3. Now we need to install one more library in our newly created environment. In order to install this protobuf package, make sure we have to navigate to the below path and install the package.

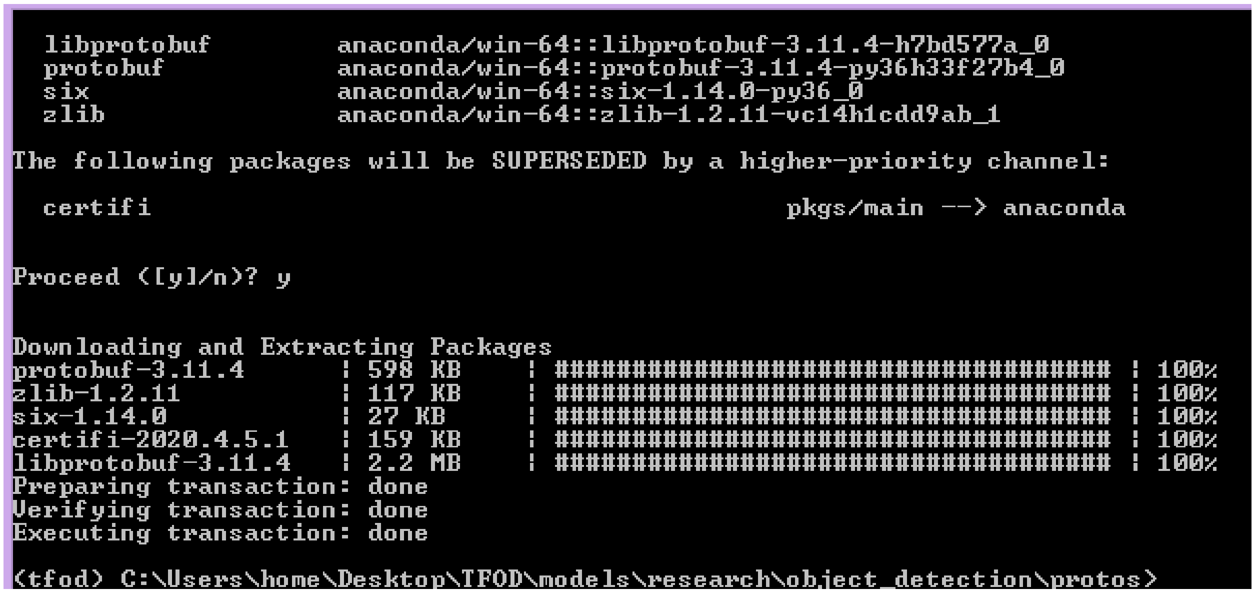
“\Desktop\TFOD(our folder name that we have created early)\models\research\object_detection\protos\ conda install -c anaconda protobuf ”

Command : conda install -c anaconda protobuf

This is a kind of package which compile the protobuf file to our python files, because in our protos folder all the files have extension of .proto. So, our python compiler wont be able to understand the protobuf files. In this object detection most of the files are written in the form of google protobuf.



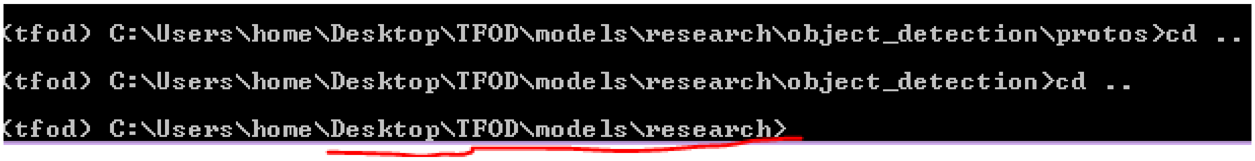
```
(tfod) C:\Users\home\Desktop\TFOD\models\research\object_detection\protos>conda
install -c anaconda protobuf
```



```
  libprotobuf          anaconda/win-64::libprotobuf-3.11.4-h7bd577a_0
  protobuf             anaconda/win-64::protobuf-3.11.4-py36h33f27b4_0
  six                  anaconda/win-64::six-1.14.0-py36_0
  zlib                 anaconda/win-64::zlib-1.2.11-vc14h1cdd9ab_1

The following packages will be SUPERSEDED by a higher-priority channel:

  certifi                                            pkgs/main --> anaconda


Proceed ([y]/n)? y


Downloading and Extracting Packages
protobuf-3.11.4      | 598 KB    | #################################### | 100%
zlib-1.2.11          | 117 KB    | #################################### | 100%
six-1.14.0           | 27 KB     | #################################### | 100%
certifi-2020.4.5.1   | 159 KB    | #################################### | 100%
libprotobuf-3.11.4   | 2.2 MB    | #################################### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(tfod) C:\Users\home\Desktop\TFOD\models\research\object_detection\protos>
```

Installation completed Successfully.

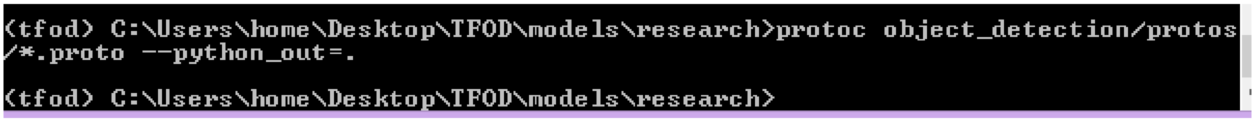
4. Now I need to do a conversion of this protos file to python file. For that we need to switch back to the research folder like below.



```
(tfod) C:\Users\home\Desktop\TFOD\models\research\object_detection\protos>cd ..

(tfod) C:\Users\home\Desktop\TFOD\models\research\object_detection>cd ..

(tfod) C:\Users\home\Desktop\TFOD\models\research>
```

Command : protoc object_detection/protos/*.proto --python_out=.

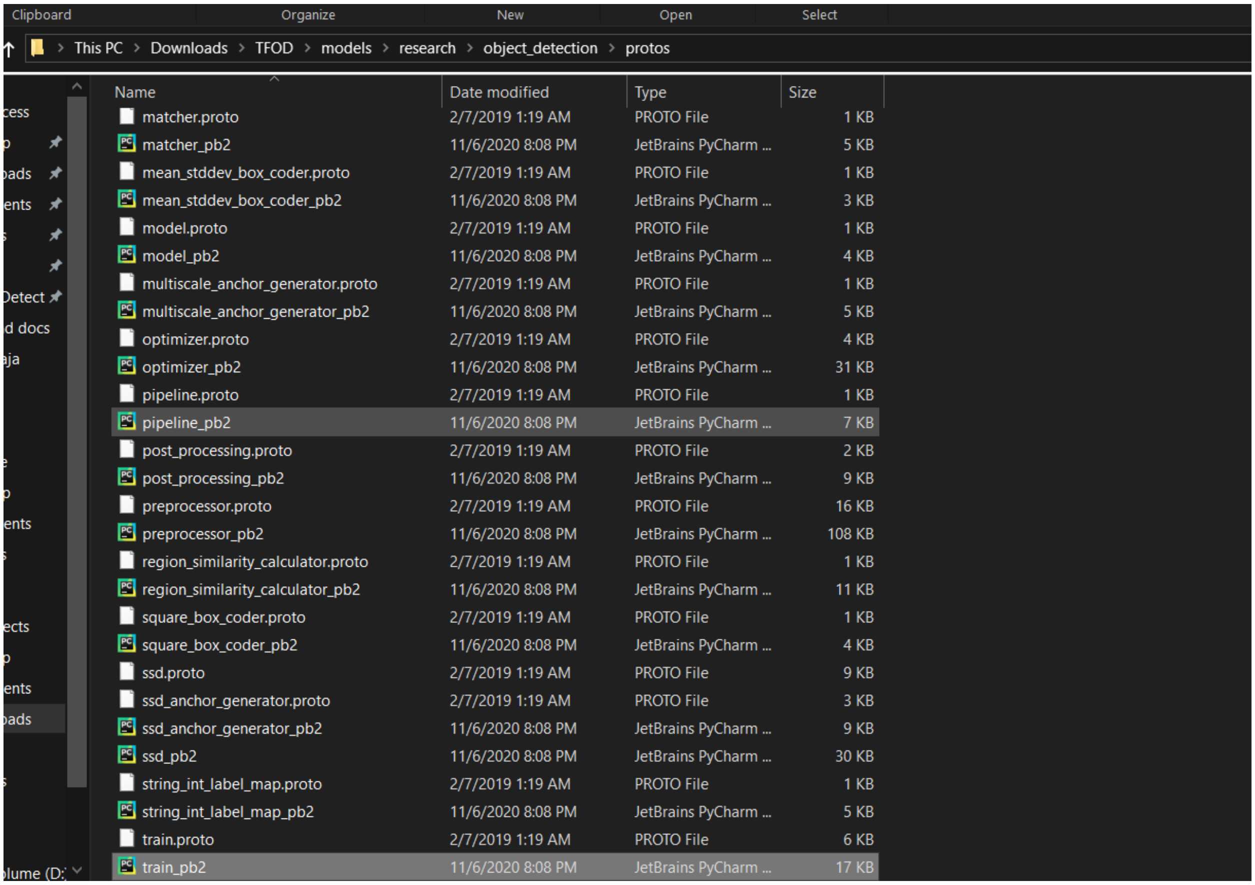
```
protoc object_detection/protos/*.proto --python_out=.
```



```
(tfod) C:\Users\home\Desktop\TFOD\models\research>protoc object_detection/protos
/*.proto --python_out=.

(tfod) C:\Users\home\Desktop\TFOD\models\research>
```

The command is successfully executed.

After the command got executed we will be able to see for each and every protos file corresponding .py files are generated.

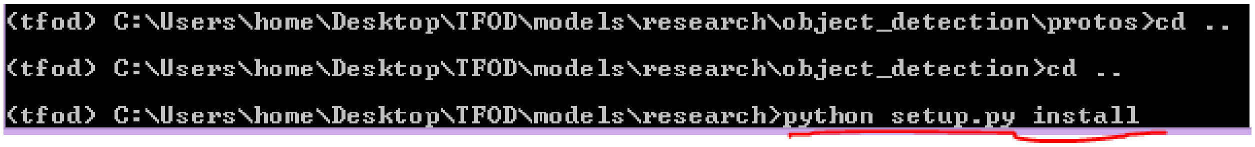


Clipboard
Organize
New
Open
Select
This PC > Downloads > TFOD > models > research > object_detection > protos

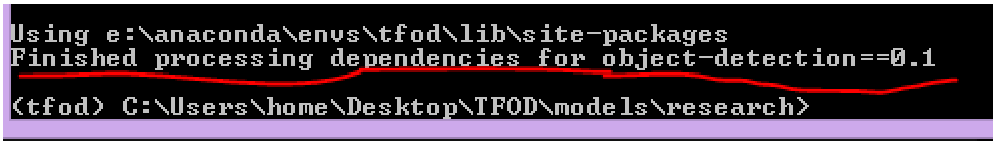
Name	Date modified	Type	Size
matcher.proto	2/7/2019 1:19 AM	PROTO File	1 KB
matcher_pb2	11/6/2020 8:08 PM	JetBrains PyCharm ...	5 KB
mean_stddev_box_coder.proto	2/7/2019 1:19 AM	PROTO File	1 KB
mean_stddev_box_coder_pb2	11/6/2020 8:08 PM	JetBrains PyCharm ...	3 KB
model.proto	2/7/2019 1:19 AM	PROTO File	1 KB
model_pb2	11/6/2020 8:08 PM	JetBrains PyCharm ...	4 KB
multiscale_anchor_generator.proto	2/7/2019 1:19 AM	PROTO File	1 KB
multiscale_anchor_generator_pb2	11/6/2020 8:08 PM	JetBrains PyCharm ...	5 KB
optimizer.proto	2/7/2019 1:19 AM	PROTO File	4 KB
optimizer_pb2	11/6/2020 8:08 PM	JetBrains PyCharm ...	31 KB
pipeline.proto	2/7/2019 1:19 AM	PROTO File	1 KB
pipeline_pb2	11/6/2020 8:08 PM	JetBrains PyCharm ...	7 KB
post_processing.proto	2/7/2019 1:19 AM	PROTO File	2 KB
post_processing_pb2	11/6/2020 8:08 PM	JetBrains PyCharm ...	9 KB
preprocessor.proto	2/7/2019 1:19 AM	PROTO File	16 KB
preprocessor_pb2	11/6/2020 8:08 PM	JetBrains PyCharm ...	108 KB
region_similarity_calculator.proto	2/7/2019 1:19 AM	PROTO File	1 KB
region_similarity_calculator_pb2	11/6/2020 8:08 PM	JetBrains PyCharm ...	11 KB
square_box_coder.proto	2/7/2019 1:19 AM	PROTO File	1 KB
square_box_coder_pb2	11/6/2020 8:08 PM	JetBrains PyCharm ...	4 KB
ssd.proto	2/7/2019 1:19 AM	PROTO File	9 KB
ssd_anchor_generator.proto	2/7/2019 1:19 AM	PROTO File	3 KB
ssd_anchor_generator_pb2	11/6/2020 8:08 PM	JetBrains PyCharm ...	9 KB
ssd_pb2	11/6/2020 8:08 PM	JetBrains PyCharm ...	30 KB
string_int_label_map.proto	2/7/2019 1:19 AM	PROTO File	1 KB
string_int_label_map_pb2	11/6/2020 8:08 PM	JetBrains PyCharm ...	5 KB
train.proto	2/7/2019 1:19 AM	PROTO File	6 KB
train_pb2	11/6/2020 8:08 PM	JetBrains PyCharm ...	17 KB

5. The Next thing is we need to install the file(setup.py).That file is available inside the research folder.

Cmd : **python setup.py install**



```
(tfod) C:\Users\home\Desktop\TFOD\models\research\object_detection\protos>cd ..

(tfod) C:\Users\home\Desktop\TFOD\models\research\object_detection>cd ..

(tfod) C:\Users\home\Desktop\TFOD\models\research>python setup.py install
```



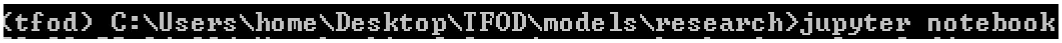
```
Using e:\anaconda\envs\tfod\lib\site-packages
Finished processing dependencies for object-detection==0.1

(tfod) C:\Users\home\Desktop\TFOD\models\research>
```

So, the object detection has been installed in my system successfully.

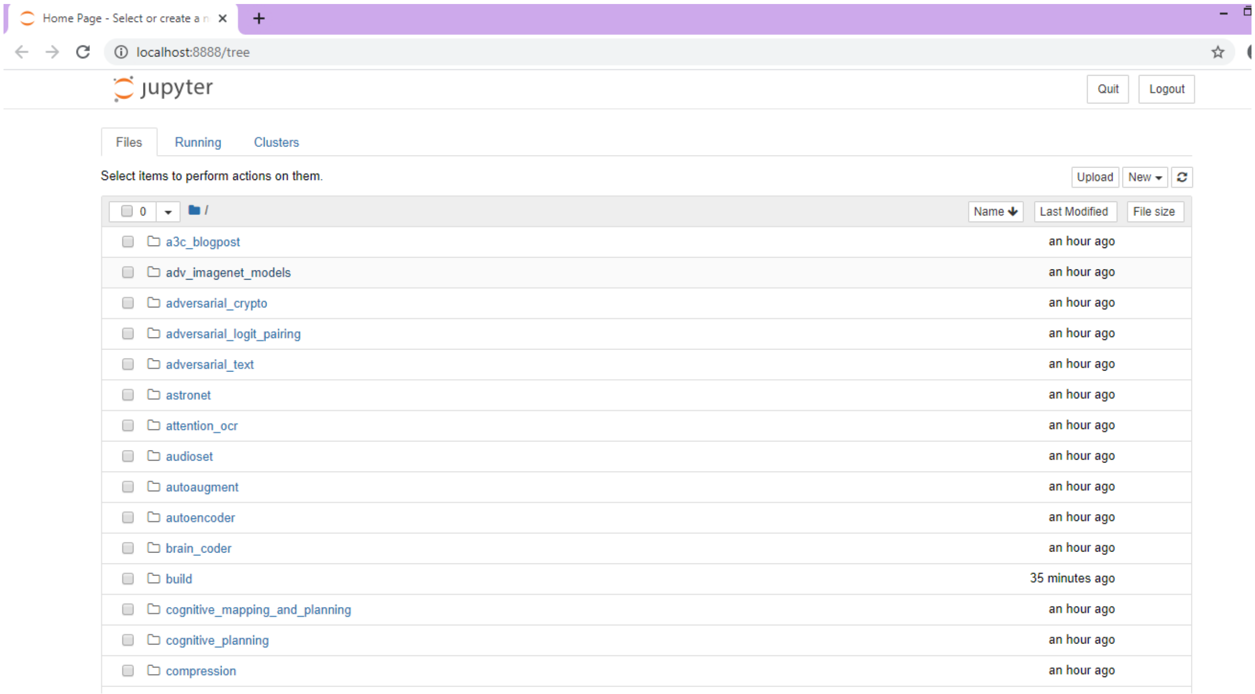
6. Now we need to do the verification whether really those files got converted or not. For that we need to do the Visual inspection through command prompt.

C. Verification

To verify that open the Jupyter notebook through anaconda prompt



```
(tfod) C:\Users\home\Desktop\TFOD\models\research>jupyter notebook
```

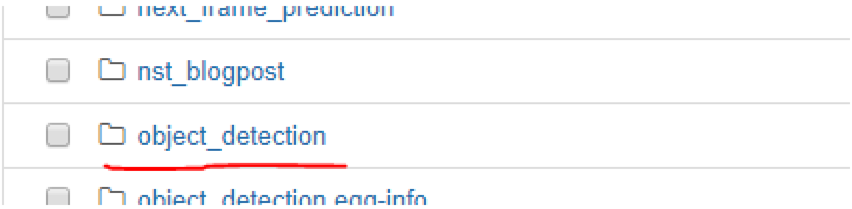


Select items to perform actions on them.

Name	Last Modified
a3c_blogpost	an hour ago
adv_imagenet_models	an hour ago
adversarial_crypto	an hour ago
adversarial_logit_pairing	an hour ago
adversarial_text	an hour ago
astronet	an hour ago
attention_ocr	an hour ago
audioset	an hour ago
autoaugment	an hour ago
autoencoder	an hour ago
brain_coder	an hour ago
build	35 minutes ago
cognitive_mapping_and_planning	an hour ago
cognitive_planning	an hour ago
compression	an hour ago

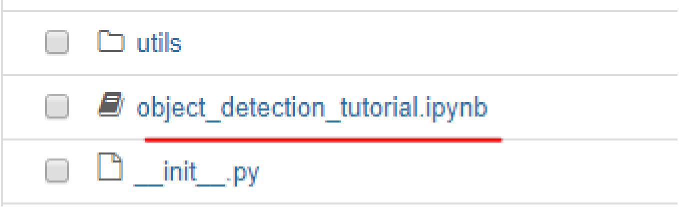
It automatically open's the research folder.

Now open the object detection folder.



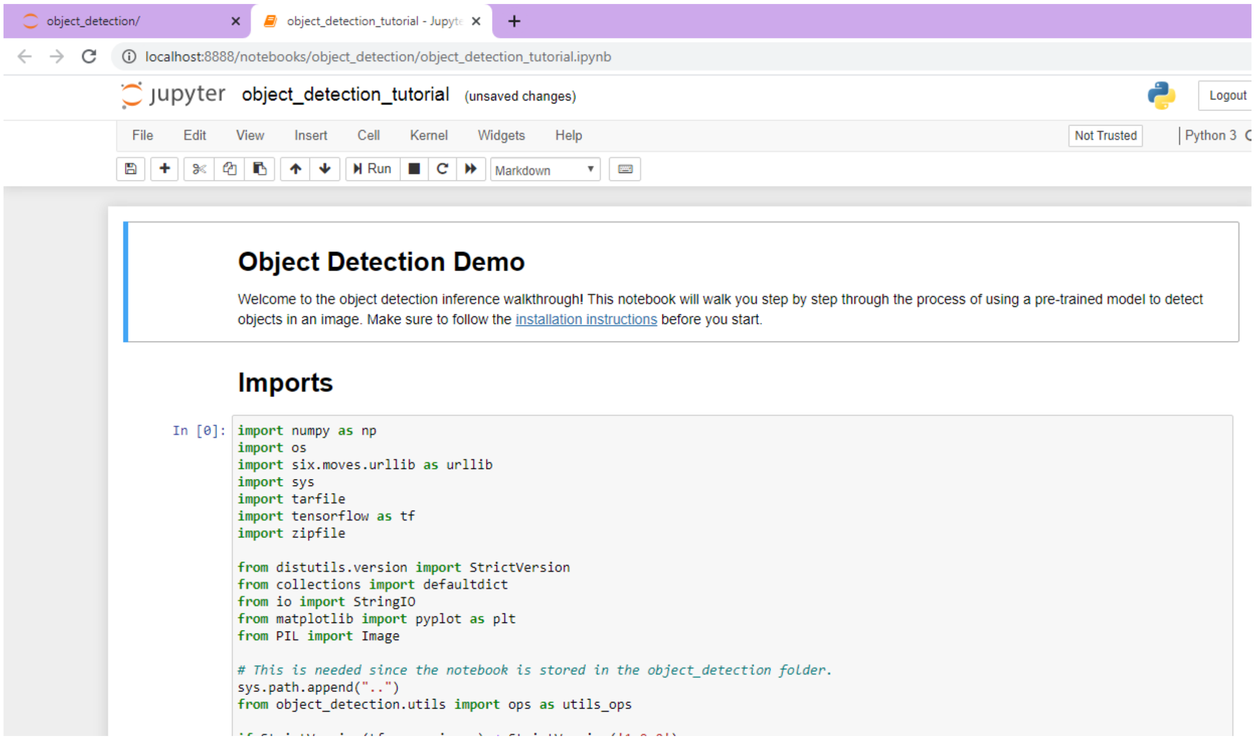
next_frame_prediction
nst_blogpost
object_detection
object_detection.egg-info

Inside the object detection folder there will be a Jupyter notebook file like below.



utils
object_detection_tutorial.ipynb
__init__.py

Open that file.


object_detection/
object_detection_tutorial - Jupyte
localhost:8888/notebooks/object_detection/object_detection_tutorial.ipynb
jupyter
object_detection_tutorial
(unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3
Run
Markdown

Object Detection Demo

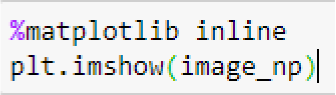
Welcome to the object detection inference walkthrough! This notebook will walk you step by step through the process of using a pre-trained model to detect objects in an image. Make sure to follow the installation instructions before you start.

Imports

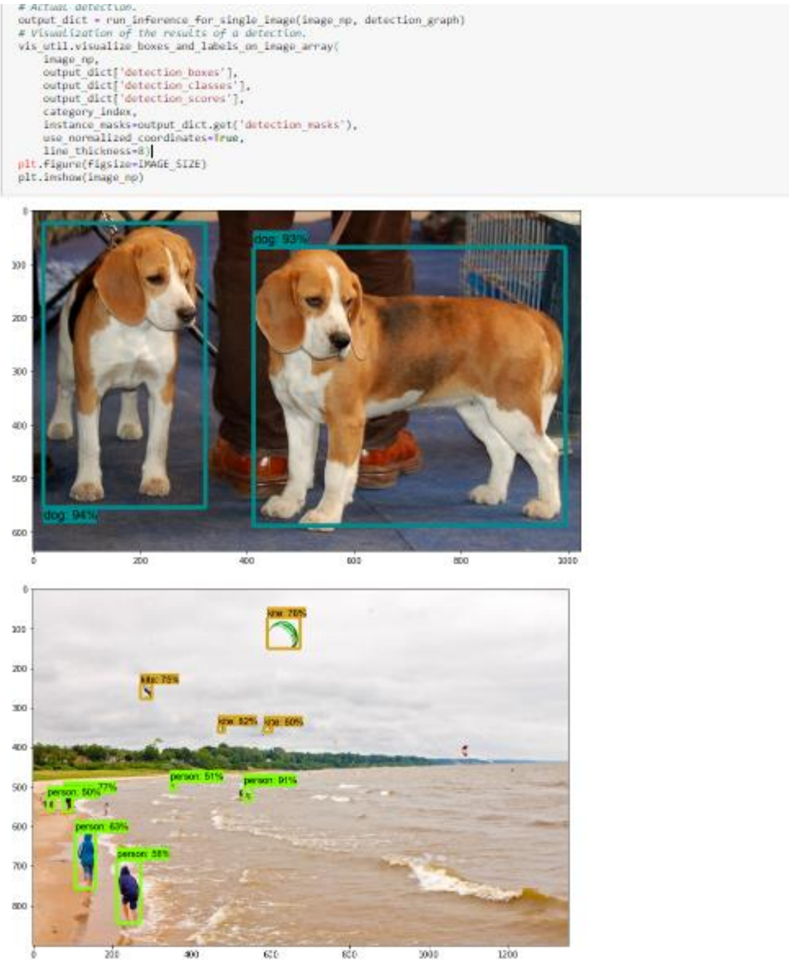
In [0]:
```
import numpy as np
import os
import six.moves.urllib as urllib
import sys
import tarfile
import tensorflow as tf
import zipfile

from distutils.version import StrictVersion
from collections import defaultdict
from io import StringIO
from matplotlib import 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
```

Now run all the cells.

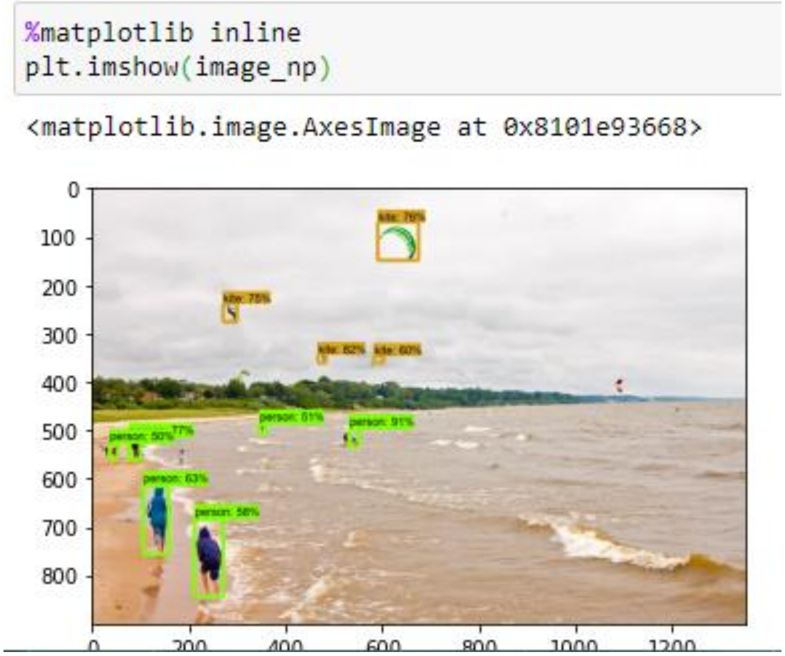
If you get any errors in any of the cells means there is a problem. If all the cells executed successfully then we need to check whether we getting the results or not. For that we need to add one cell in the same notebook file.



```
%matplotlib inline
plt.imshow(image_np)
```



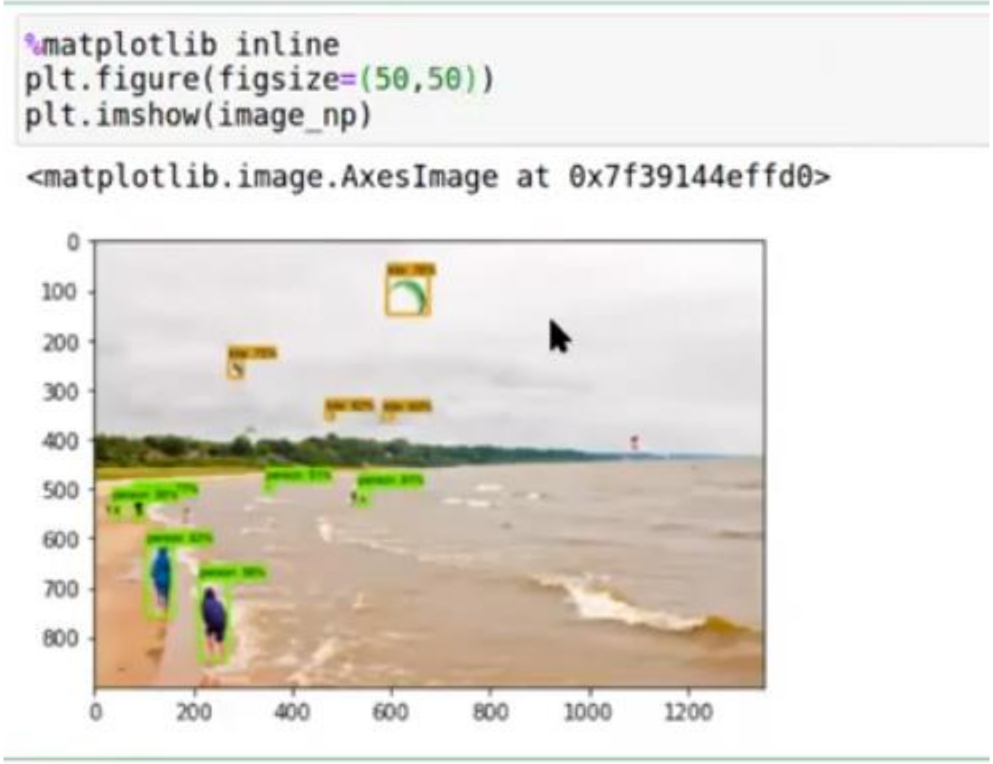
```
# Actual detection.
output_dict = run_inference_for_single_image(image_np, detection_graph)
# Visualization of the results of a detection.
vis_util.visualize_boxes_and_labels_on_image_array(
    image_np,
    output_dict['detection_boxes'],
    output_dict['detection_classes'],
    output_dict['detection_scores'],
    category_index,
    instance_masks=output_dict.get('detection_masks'),
    use_normalized_coordinates=True,
    line_thickness=8)
plt.figure(figsize=IMAGE_SIZE)
plt.imshow(image_np)
```



```
%matplotlib inline
plt.imshow(image_np)
```

```
<matplotlib.image.AxesImage at 0x8101e93668>
```

If you want to change the width of the picture



```
%matplotlib inline
plt.figure(figsize=(50,50))
plt.imshow(image_np)
```

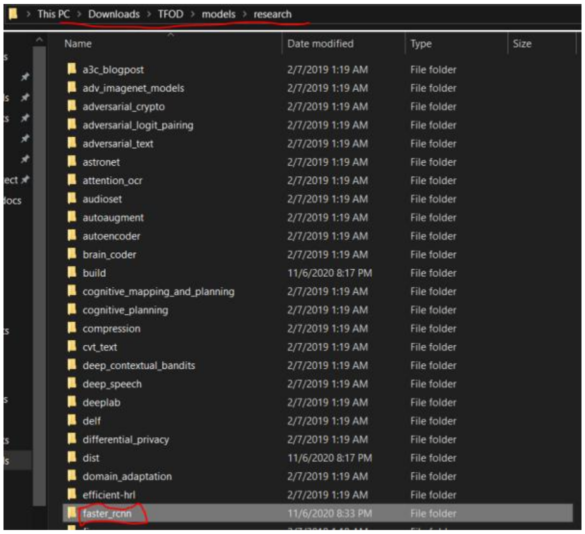
```
<matplotlib.image.AxesImage at 0x7f39144effd0>
```

Then you will be able to see those images, so every step that I have done is correct. Now we are good to go for custom training.

Move files to the research folder

Switch back to TFOD directory in your local system

a) Move the **faster_rcnn folder to the research folder**

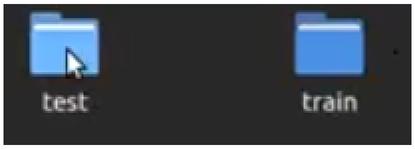


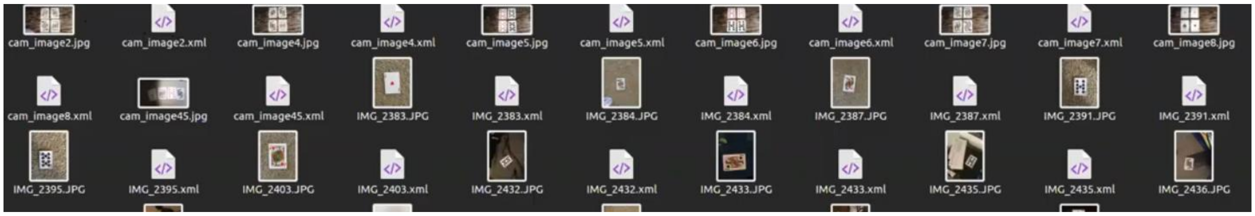
This PC › Downloads › TFOD › models › research

Name	Date modified	Type	Size
a3c_blogpost	2/7/2019 1:19 AM	File folder	
adv_imagenet_models	2/7/2019 1:19 AM	File folder	
adversarial_crypto	2/7/2019 1:19 AM	File folder	
adversarial_logit_pairing	2/7/2019 1:19 AM	File folder	
adversarial_text	2/7/2019 1:19 AM	File folder	
astronet	2/7/2019 1:19 AM	File folder	
attention_ocr	2/7/2019 1:19 AM	File folder	
audioset	2/7/2019 1:19 AM	File folder	
autoaugment	2/7/2019 1:19 AM	File folder	
autoencoder	2/7/2019 1:19 AM	File folder	
brain_coder	2/7/2019 1:19 AM	File folder	
build	11/6/2020 8:17 PM	File folder	
cognitive_mapping_and_planning	2/7/2019 1:19 AM	File folder	
cognitive_planning	2/7/2019 1:19 AM	File folder	
compression	2/7/2019 1:19 AM	File folder	
cvt_text	2/7/2019 1:19 AM	File folder	
deep_contextual_bandits	2/7/2019 1:19 AM	File folder	
deep_speech	2/7/2019 1:19 AM	File folder	
deeplab	2/7/2019 1:19 AM	File folder	
delf	2/7/2019 1:19 AM	File folder	
differential_privacy	2/7/2019 1:19 AM	File folder	
dist	11/6/2020 8:17 PM	File folder	
domain_adaptation	2/7/2019 1:19 AM	File folder	
efficient-hrl	2/7/2019 1:19 AM	File folder	
faster_rcnn	11/6/2020 8:33 PM	File folder	

b) Copy the all the subfolders (images, training, generate_tfrecord.py, xml_to_csv.py) which are inside the **utils folder** and paste it in to the tfod-- > models -- > research folder.

Images Folder:
In our utils -- > Images folder





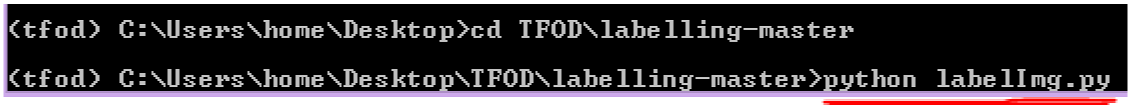
Here, Each and every image have .XML file, which basically represents, I have done the annotation or labelling for each and every images in my test and train folders.

D. How to do an Annotation

1. Annotation/Labelling

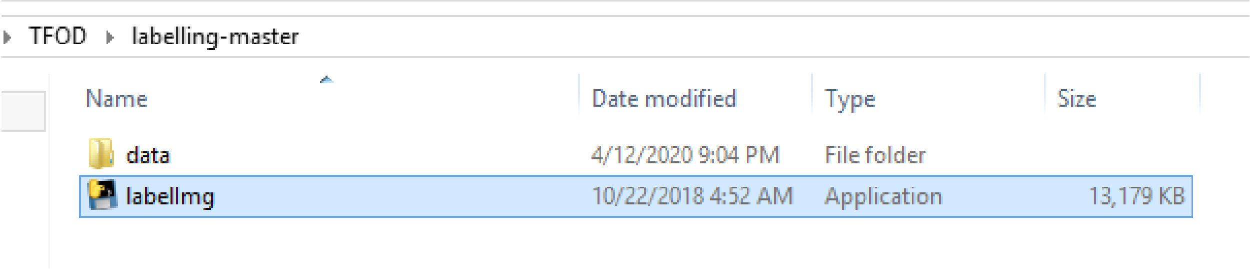
Rename the file **Windows_v1.8.0** to labelling-master.

Open the terminal.

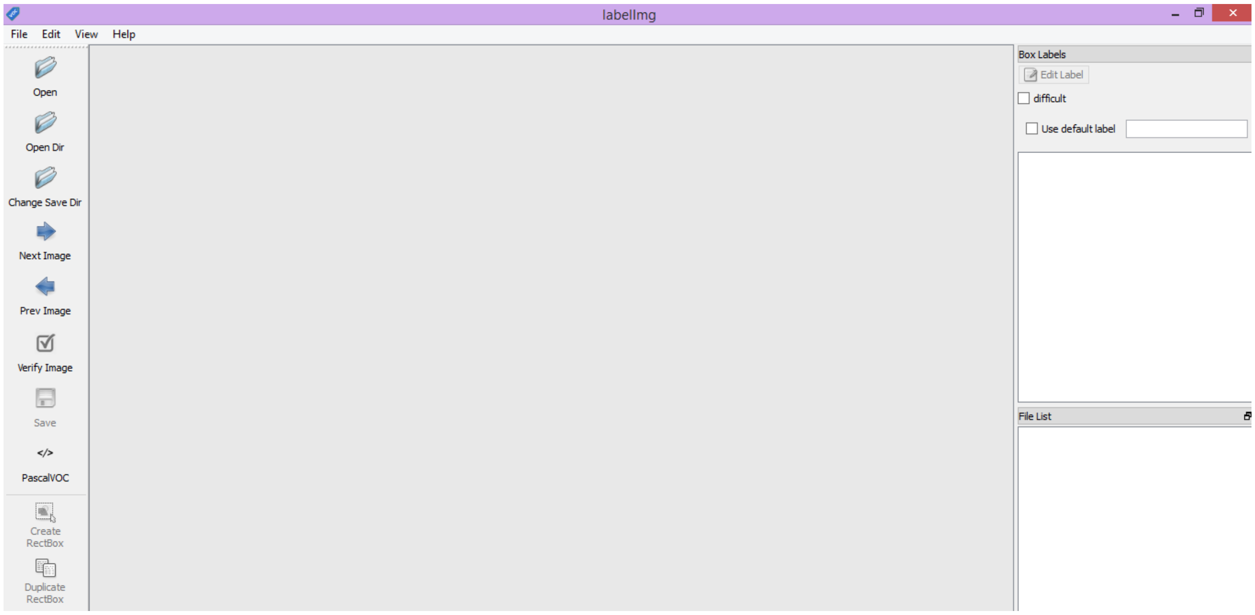


```
(tfod) C:\Users\home\Desktop>cd TFOD\labelling-master
(tfod) C:\Users\home\Desktop\TFOD\labelling-master>python labelImg.py
```

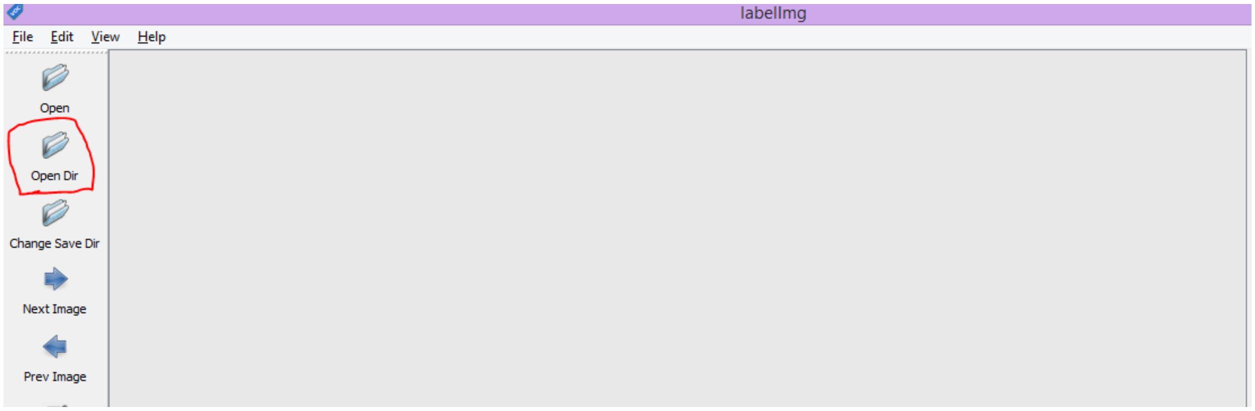
Or just go the folder directly and open the application.



```
▸ TFOD ▸ labelling-master

Name                              Date modified          Type            Size
data                              4/12/2020 9:04 PM      File folder
labelImg                          10/22/2018 4:52 AM     Application     13,179 KB
```



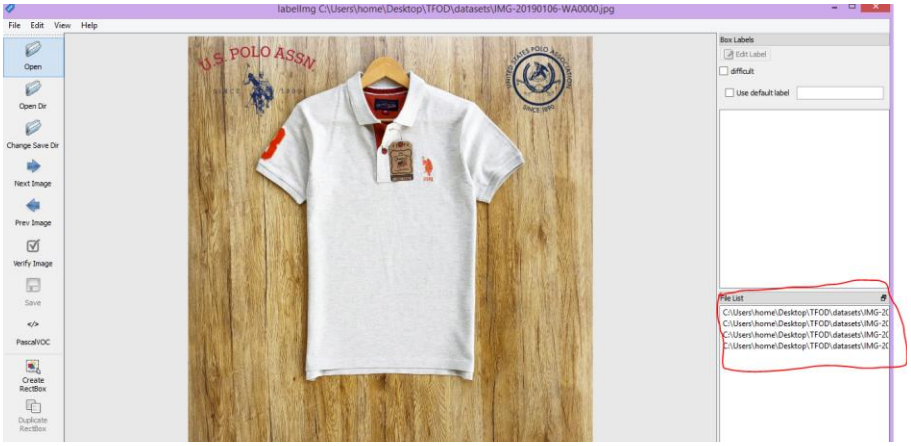
This tool is mainly used for annotations which means



TFOD --- > Utils ---- > Images ----- > train - - Do annotation for all the images

TFOD --- > Utils ---- > Images ----- > test --- Do annotation for all the images

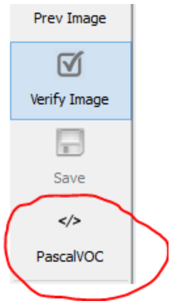
Sample Annotations Below:


labelImg C:\Users\home\Desktop\TFOD\datasets\IMG-20190106-WA0000.jpg

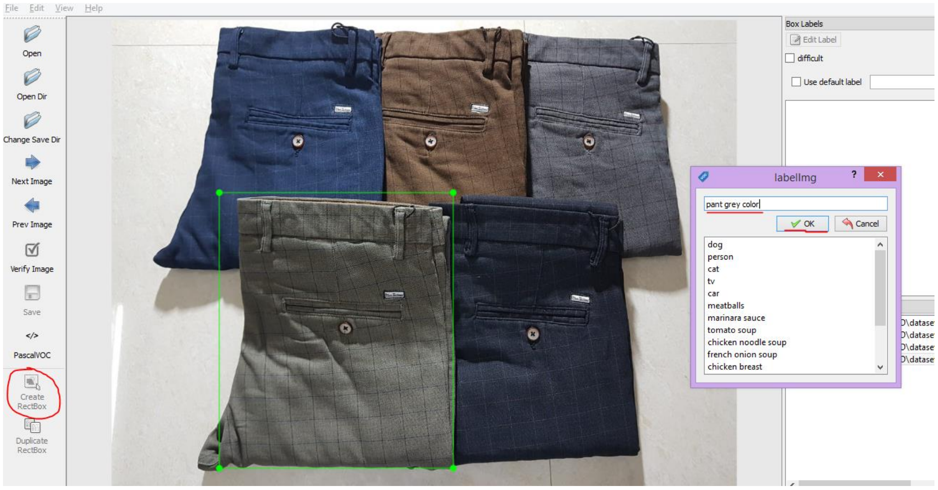
Total images in the dataset are loaded in this tool.

Now we are going to do the labelling of each image.

Make sure when you are doing the labelling for TFOD we need to select the **PascalVOC** option and don't select YOLO.

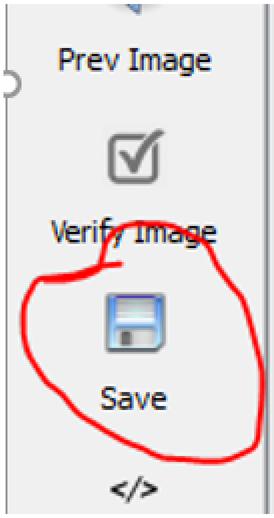


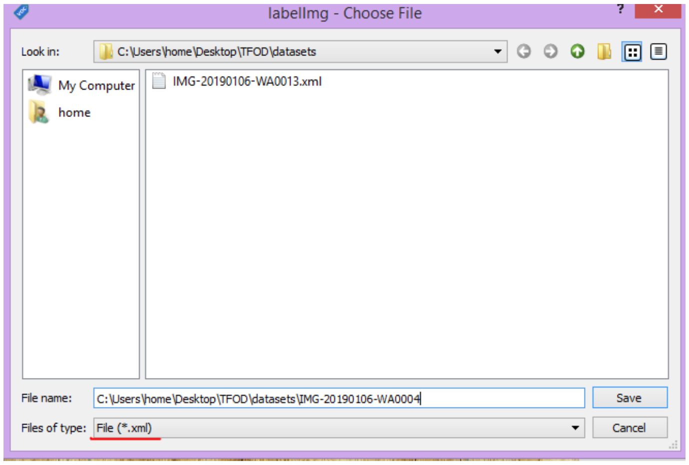
Now how to do the labelling. For that click **CreateRectBox**


pant grey color

dog
person
cat
tv
car
meatballs
marinara sauce
tomato soup
chicken noodle soup
french onion soup
chicken breast

For One-by-one image, we need to save the file after the labelling part was done



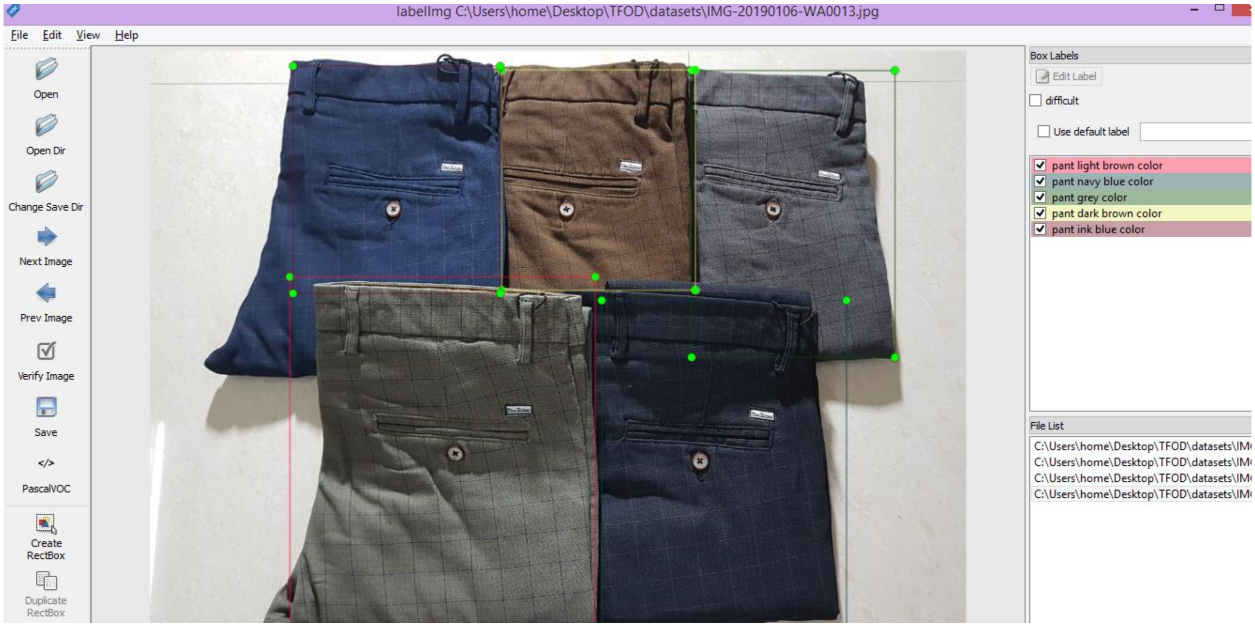


Look in: C:\Users\home\Desktop\TFOD\datasets

IMG-20190106-WA0013.xml

File name: C:\Users\home\Desktop\TFOD\datasets\IMG-20190106-WA0004

Files of type: File (*.xml)



labelImg C:\Users\home\Desktop\TFOD\datasets\IMG-20190106-WA0013.jpg

□ difficult

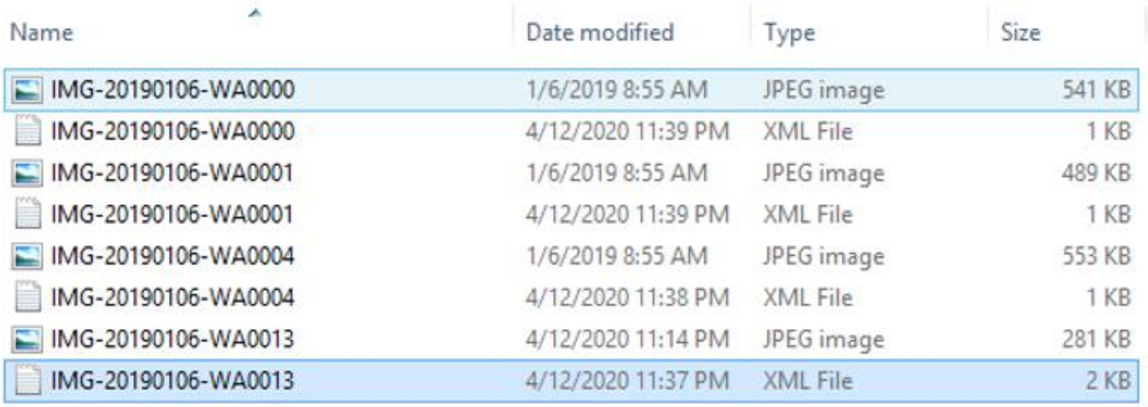
□ Use default label

☑ pant light brown color
☑ pant navy blue color
☑ pant grey color
☑ pant dark brown color
☑ pant ink blue color

C:\Users\home\Desktop\TFOD\datasets\IM
C:\Users\home\Desktop\TFOD\datasets\IM
C:\Users\home\Desktop\TFOD\datasets\IM
C:\Users\home\Desktop\TFOD\datasets\IM

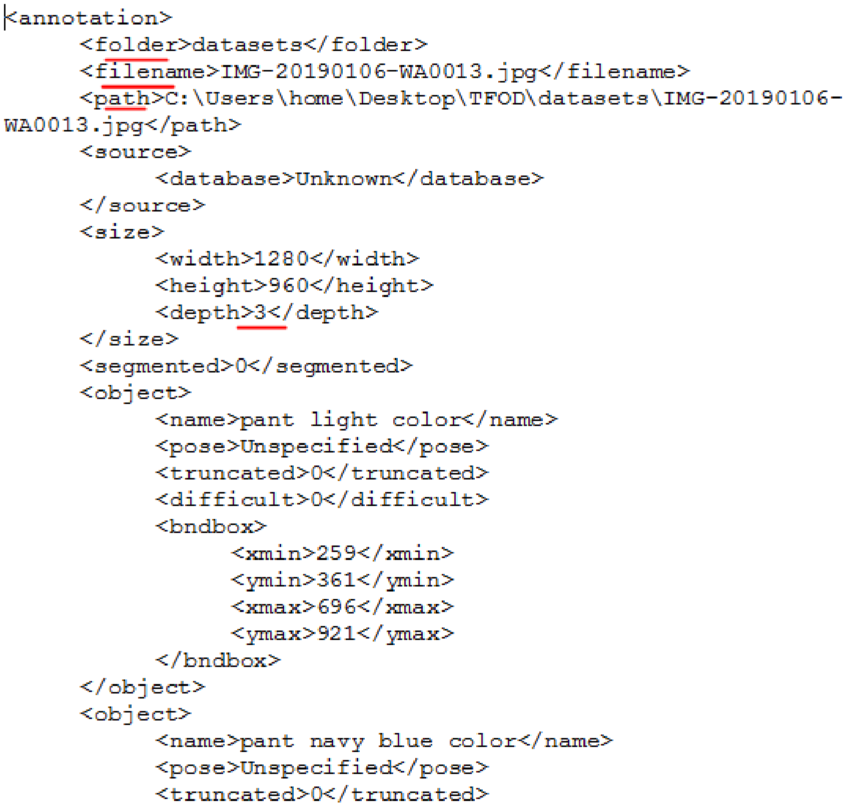
I had done the labelling part for each and every images.

Here, for each and every image .XML files got generated, which basically represents, I have done the annotation or labelling for each and every image in my test and train folders.



Name	Date modified	Type	Size
IMG-20190106-WA0000	1/6/2019 8:55 AM	JPEG image	541 KB
IMG-20190106-WA0000	4/12/2020 11:39 PM	XML File	1 KB
IMG-20190106-WA0001	1/6/2019 8:55 AM	JPEG image	489 KB
IMG-20190106-WA0001	4/12/2020 11:39 PM	XML File	1 KB
IMG-20190106-WA0004	1/6/2019 8:55 AM	JPEG image	553 KB
IMG-20190106-WA0004	4/12/2020 11:38 PM	XML File	1 KB
IMG-20190106-WA0013	4/12/2020 11:14 PM	JPEG image	281 KB
IMG-20190106-WA0013	4/12/2020 11:37 PM	XML File	2 KB

Sample XML File:

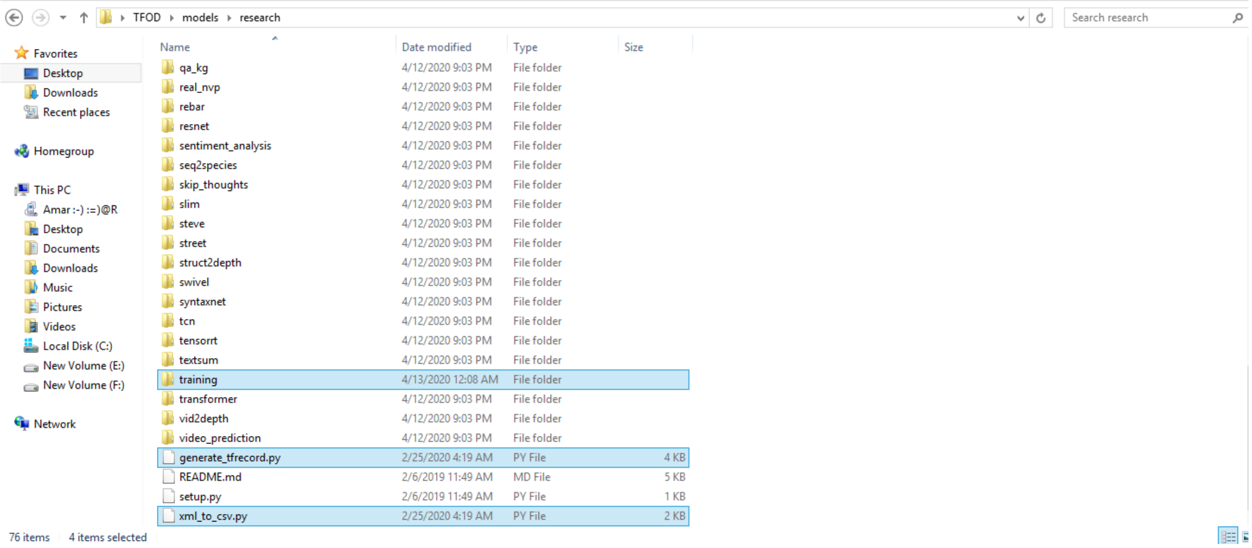


```
<annotation>
	<folder>datasets</folder>
	<filename>IMG-20190106-WA0013.jpg</filename>
	<path>C:\Users\home\Desktop\TFOD\datasets\IMG-20190106-
WA0013.jpg</path>
	<source>
		<database>Unknown</database>
	</source>
	<size>
		<width>1280</width>
		<height>960</height>
		<depth>3</depth>
	</size>
	<segmented>0</segmented>
	<object>
		<name>pant light color</name>
		<pose>Unspecified</pose>
		<truncated>0</truncated>
		<difficult>0</difficult>
		<bndbox>
			<xmin>259</xmin>
			<ymin>361</ymin>
			<xmax>696</xmax>
			<ymax>921</ymax>
		</bndbox>
	</object>
	<object>
		<name>pant navy blue color</name>
		<pose>Unspecified</pose>
		<truncated>0</truncated>
```

Here the depth is the very important and which is basically represents RGB images or colorful images, if the depth is less than 3 then we need to delete that image. Here we have coordinates. (xmin, ymin , xmax, ymax) for each and every images.

2. Custom training process

We need to move some of the files, whatever files in the Utils copy all the files and paste it into the research folder.



Name	Date modified	Type	Size
qa_kg	4/12/2020 9:03 PM	File folder	
real_nvp	4/12/2020 9:03 PM	File folder	
rebar	4/12/2020 9:03 PM	File folder	
resnet	4/12/2020 9:03 PM	File folder	
sentiment_analysis	4/12/2020 9:03 PM	File folder	
seq2species	4/12/2020 9:03 PM	File folder	
skip_thoughts	4/12/2020 9:03 PM	File folder	
slim	4/12/2020 9:03 PM	File folder	
steve	4/12/2020 9:03 PM	File folder	
street	4/12/2020 9:03 PM	File folder	
struct2depth	4/12/2020 9:03 PM	File folder	
swivel	4/12/2020 9:03 PM	File folder	
syntaxnet	4/12/2020 9:03 PM	File folder	
tcn	4/12/2020 9:03 PM	File folder	
tensorrt	4/12/2020 9:03 PM	File folder	
textsum	4/12/2020 9:03 PM	File folder	
training	4/13/2020 12:08 AM	File folder	
transformer	4/12/2020 9:03 PM	File folder	
vid2depth	4/12/2020 9:03 PM	File folder	
video_prediction	4/12/2020 9:03 PM	File folder	
generate_tfrecord.py	2/25/2020 4:19 AM	PY File	4 KB
README.md	2/6/2019 11:49 AM	MD File	5 KB
setup.py	2/6/2019 11:49 AM	PY File	1 KB
xml_to_csv.py	2/25/2020 4:19 AM	PY File	2 KB

76 items 4 items selected

Conversion of XML files to tf records (1.XML to CSV, 2. CSV to .tf records):

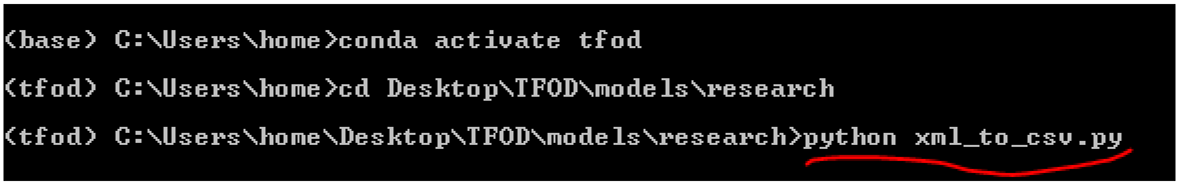
Now we are going to convert **XML to csv and again csv to tfrecords** (tf records are basically a binary format) for training the model.

Once we done the labelling part, for each and every image in our dataset we have corresponding XML files are generated.

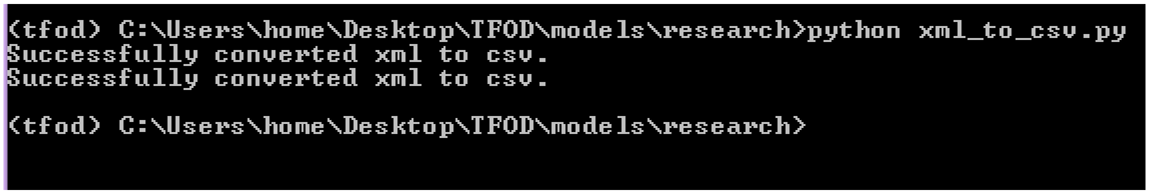
1. Convert all the XML files into a single CSV file.

Open the terminal ----- > Navigate to the research folder

Command : **python xml_to_csv.py**

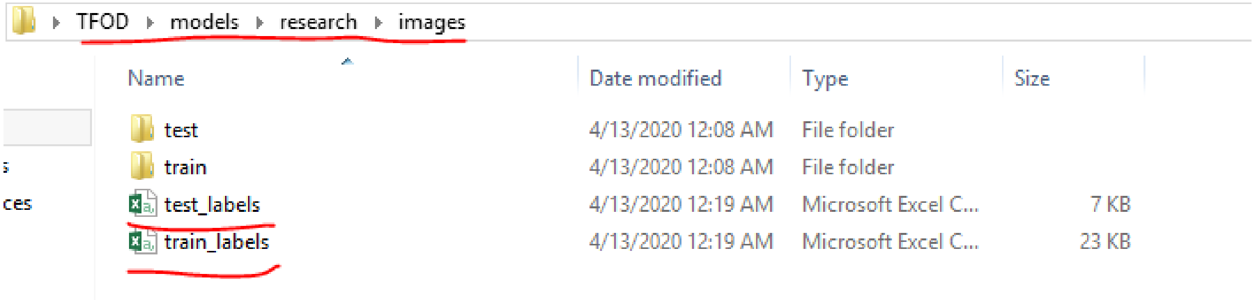


```
(base) C:\Users\home>conda activate tfod
(tfod) C:\Users\home>cd Desktop\TFOD\models\research
(tfod) C:\Users\home\Desktop\TFOD\models\research>python xml_to_csv.py
```

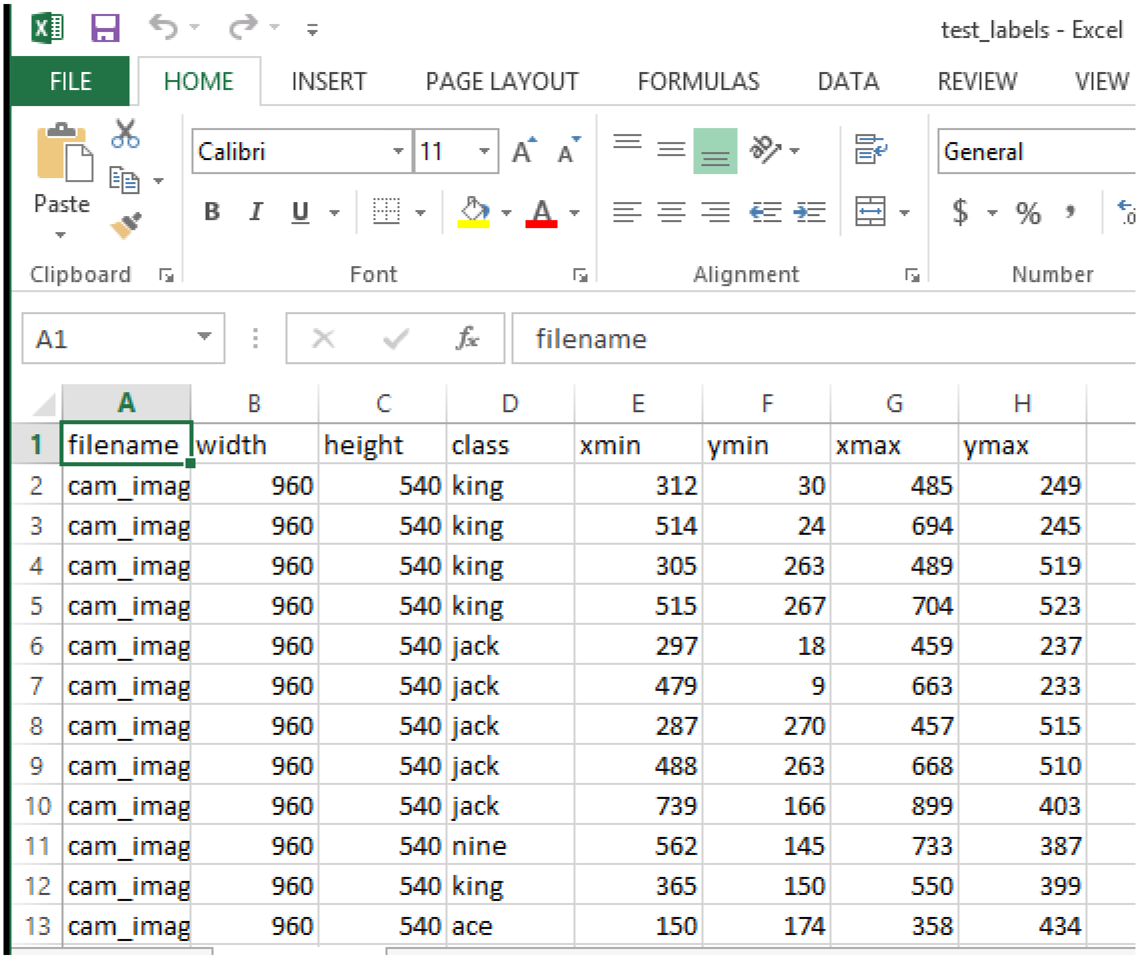


```
(tfod) C:\Users\home\Desktop\TFOD\models\research>python xml_to_csv.py
Successfully converted xml to csv.
Successfully converted xml to csv.

(tfod) C:\Users\home\Desktop\TFOD\models\research>
```

CSV files are created inside the images folder.



Name	Date modified	Type	Size
test	4/13/2020 12:08 AM	File folder	
train	4/13/2020 12:08 AM	File folder	
test_labels	4/13/2020 12:19 AM	Microsoft Excel C...	7 KB
train_labels	4/13/2020 12:19 AM	Microsoft Excel C...	23 KB



	A	B	C	D	E	F	G	H
1	filename	width	height	class	xmin	ymin	xmax	ymax
2	cam_imag	960	540	king	312	30	485	249
3	cam_imag	960	540	king	514	24	694	245
4	cam_imag	960	540	king	305	263	489	519
5	cam_imag	960	540	king	515	267	704	523
6	cam_imag	960	540	jack	297	18	459	237
7	cam_imag	960	540	jack	479	9	663	233
8	cam_imag	960	540	jack	287	270	457	515
9	cam_imag	960	540	jack	488	263	668	510
10	cam_imag	960	540	jack	739	166	899	403
11	cam_imag	960	540	nine	562	145	733	387
12	cam_imag	960	540	king	365	150	550	399
13	cam_imag	960	540	ace	150	174	358	434

So, all the coordinates are captured in the csv file and kept inside the single place.

CSV to tf records:

Navigate to the research folder and we need to generate the csv record from XML file for the “Test dataset Xml file”.

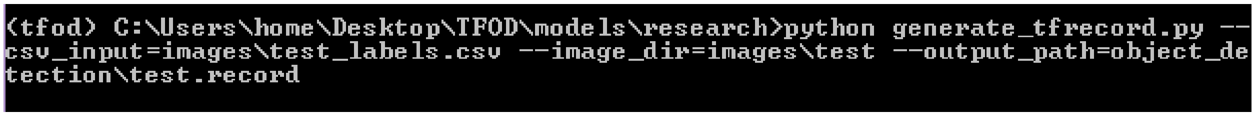
Command: test folder xml file: `python generate_tfrecord.py --csv_input=images/test_labels.csv --image_dir=images/test --output_path=test.record`

`Command: train` **folder xml file**: `python generate_tfrecord.py --csv_input=images/train_labels.csv --image_dir=images/train --output_path=train.record`

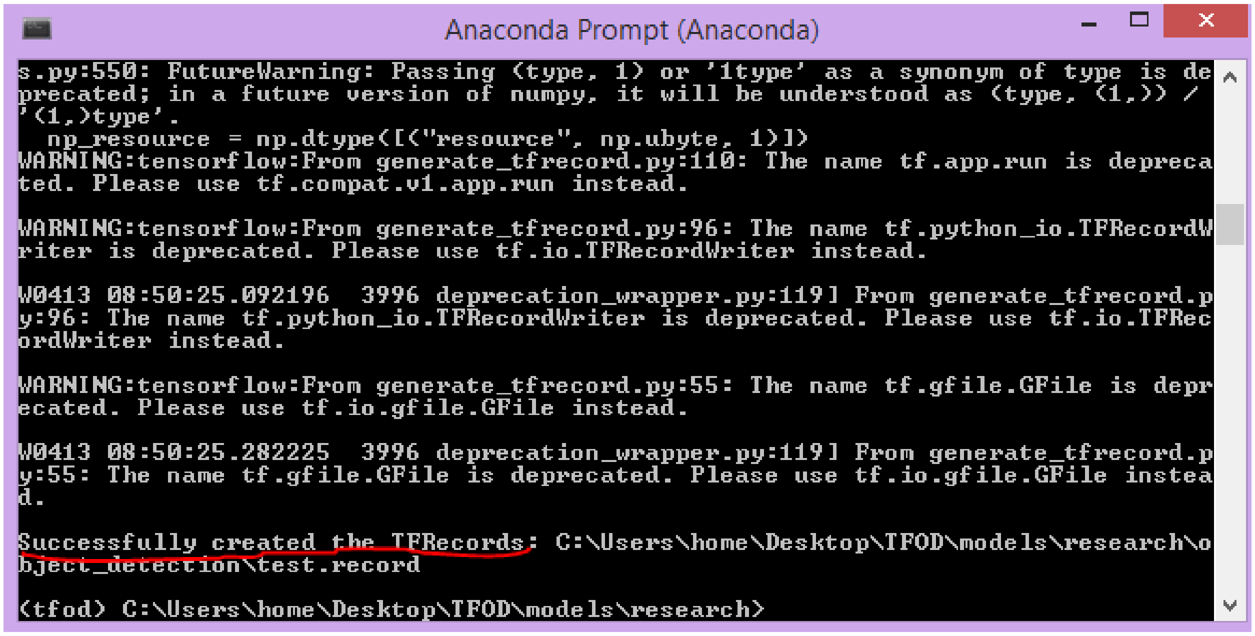
Now we are going to change csv to tfrecords. For that open command prompt

Navigate to TFOD --- > models --- > research ----- > execute the script separately for test and train

Test:



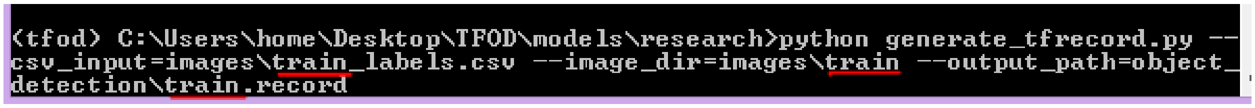
```
(tfod) C:\Users\home\Desktop\TFOD\models\research>python generate_tfrecord.py --
csv_input=images\test_labels.csv --image_dir=images\test --output_path=object_de
tection\test.record
```



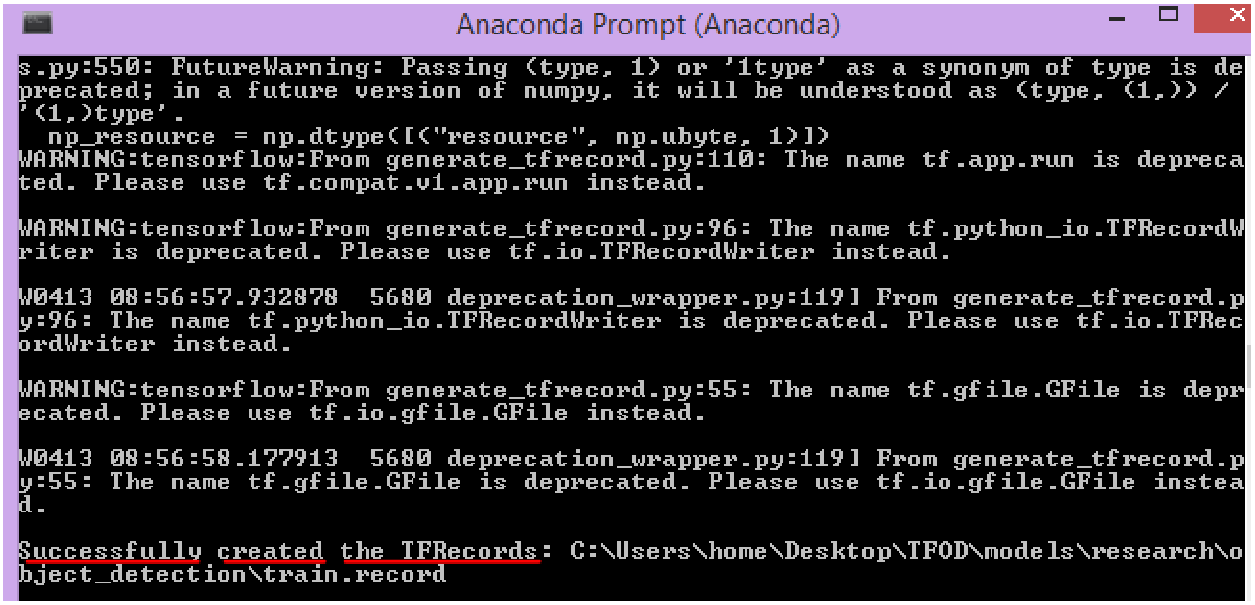
```
s.py:550: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is de
precated; in a future version of numpy, it will be understood as (type, (1,)) /
'(1,)type'.
  np_resource = np.dtype([("resource", np.ubyte, 1)])
WARNING:tensorflow:From generate_tfrecord.py:110: The name tf.app.run is depreca
ted. Please use tf.compat.v1.app.run instead.

WARNING:tensorflow:From generate_tfrecord.py:96: The name tf.python_io.TFRecordW
riter is deprecated. Please use tf.io.TFRecordWriter instead.

W0413 08:50:25.092196  3996 deprecation_wrapper.py:119] From generate_tfrecord.p
y:96: The name tf.python_io.TFRecordWriter is deprecated. Please use tf.io.TFRec
ordWriter instead.

WARNING:tensorflow:From generate_tfrecord.py:55: The name tf.gfile.GFile is depr
ecated. Please use tf.io.gfile.GFile instead.

W0413 08:50:25.282225  3996 deprecation_wrapper.py:119] From generate_tfrecord.p
y:55: The name tf.gfile.GFile is deprecated. Please use tf.io.gfile.GFile instea
d.

Successfully created the TFRecords: C:\Users\home\Desktop\TFOD\models\research\o
bject_detection\test.record

(tfod) C:\Users\home\Desktop\TFOD\models\research>
```

Now successfully Tfrecords has been created with respect to test record csv file.

Train:

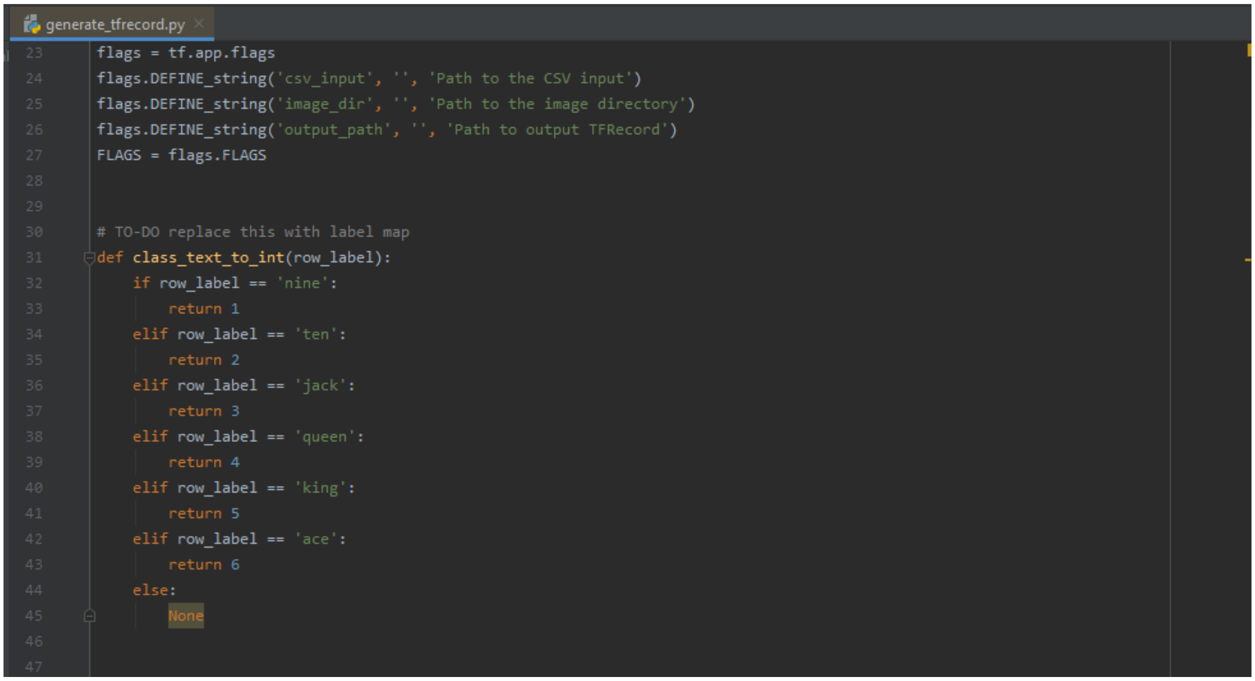


```
(tfod) C:\Users\home\Desktop\TFOD\models\research>python generate_tfrecord.py --
csv_input=images\train_labels.csv --image_dir=images\train --output_path=object_
detection\train.record
```



```
s.py:550: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is de
precated; in a future version of numpy, it will be understood as (type, (1,)) /
'(1,)type'.
  np_resource = np.dtype([("resource", np.ubyte, 1)])
WARNING:tensorflow:From generate_tfrecord.py:110: The name tf.app.run is depreca
ted. Please use tf.compat.v1.app.run instead.

WARNING:tensorflow:From generate_tfrecord.py:96: The name tf.python_io.TFRecordW
riter is deprecated. Please use tf.io.TFRecordWriter instead.

W0413 08:56:57.932878  5680 deprecation_wrapper.py:119] From generate_tfrecord.p
y:96: The name tf.python_io.TFRecordWriter is deprecated. Please use tf.io.TFRec
ordWriter instead.

WARNING:tensorflow:From generate_tfrecord.py:55: The name tf.gfile.GFile is depr
ecated. Please use tf.io.gfile.GFile instead.

W0413 08:56:58.177913  5680 deprecation_wrapper.py:119] From generate_tfrecord.p
y:55: The name tf.gfile.GFile is deprecated. Please use tf.io.gfile.GFile instea
d.

Successfully created the TFRecords: C:\Users\home\Desktop\TFOD\models\research\o
bject_detection\train.record
```

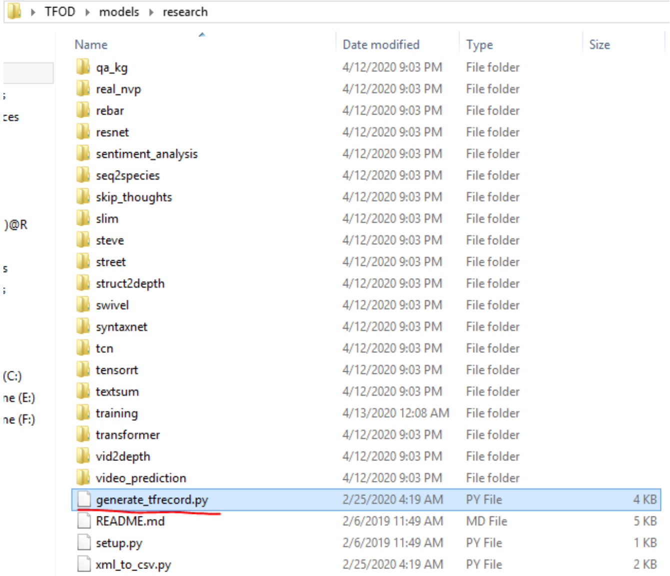
Generate_tfrecord.py:



```python
flags = tf.app.flags
flags.DEFINE_string('csv_input', '', 'Path to the CSV input')
flags.DEFINE_string('image_dir', '', 'Path to the image directory')
flags.DEFINE_string('output_path', '', 'Path to output TFRecord')
FLAGS = flags.FLAGS


# TO-DO replace this with label map
def class_text_to_int(row_label):
    if row_label == 'nine':
        return 1
    elif row_label == 'ten':
        return 2
    elif row_label == 'jack':
        return 3
    elif row_label == 'queen':
        return 4
    elif row_label == 'king':
        return 5
    elif row_label == 'ace':
        return 6
    else:
        None
```

In this file you will be able to see if and multiple elif statement. Which is used to mention the number of labels or classes here. For instance, if you are building an object detection model in that we have 10 classes inside that model, then you have to change the if and elif statements accordingly.

Sometimes you may get a none type error after the above command is executed. To overcome that issue, we need to do some changes in the generate_tfrecord.py in the research folder.



TFOD ▸ models ▸ research

Name	Date modified	Type	Size
qa_kg	4/12/2020 9:03 PM	File folder	
real_nvp	4/12/2020 9:03 PM	File folder	
rebar	4/12/2020 9:03 PM	File folder	
resnet	4/12/2020 9:03 PM	File folder	
sentiment_analysis	4/12/2020 9:03 PM	File folder	
seq2species	4/12/2020 9:03 PM	File folder	
skip_thoughts	4/12/2020 9:03 PM	File folder	
slim	4/12/2020 9:03 PM	File folder	
steve	4/12/2020 9:03 PM	File folder	
street	4/12/2020 9:03 PM	File folder	
struct2depth	4/12/2020 9:03 PM	File folder	
swivel	4/12/2020 9:03 PM	File folder	
syntaxnet	4/12/2020 9:03 PM	File folder	
tcn	4/12/2020 9:03 PM	File folder	
tensorrt	4/12/2020 9:03 PM	File folder	
textsum	4/12/2020 9:03 PM	File folder	
training	4/13/2020 12:08 AM	File folder	
transformer	4/12/2020 9:03 PM	File folder	
vid2depth	4/12/2020 9:03 PM	File folder	
video_prediction	4/12/2020 9:03 PM	File folder	
generate_tfrecord.py	2/25/2020 4:19 AM	PY File	4 KB
README.md	2/6/2019 11:49 AM	MD File	5 KB
setup.py	2/6/2019 11:49 AM	PY File	1 KB
xml_to_csv.py	2/25/2020 4:19 AM	PY File	2 KB

Perhaps, if you are getting a **None type error**, you will do some change in the else condition like below.

else:

return 0   (instead of none change to return 0)

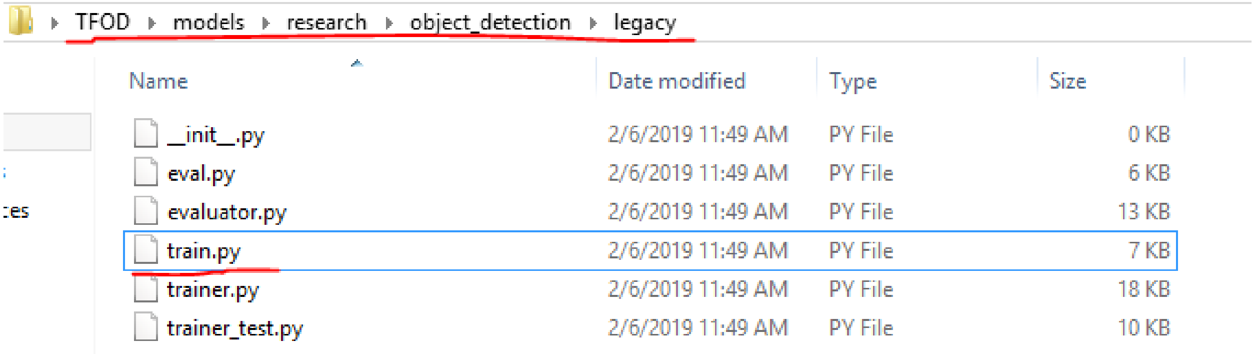
Training Part:

The next thing is we need to start the training.

Before start the training, we need some necessary files.

These two files (train, config and labelmap.pbtxt) are the inputs for our custom training's

Open TFOD -- > models -- > research -- > object_detection -- > legacy -- > **train.py** -- > copy the file -- > paste it in to the -- > research folder

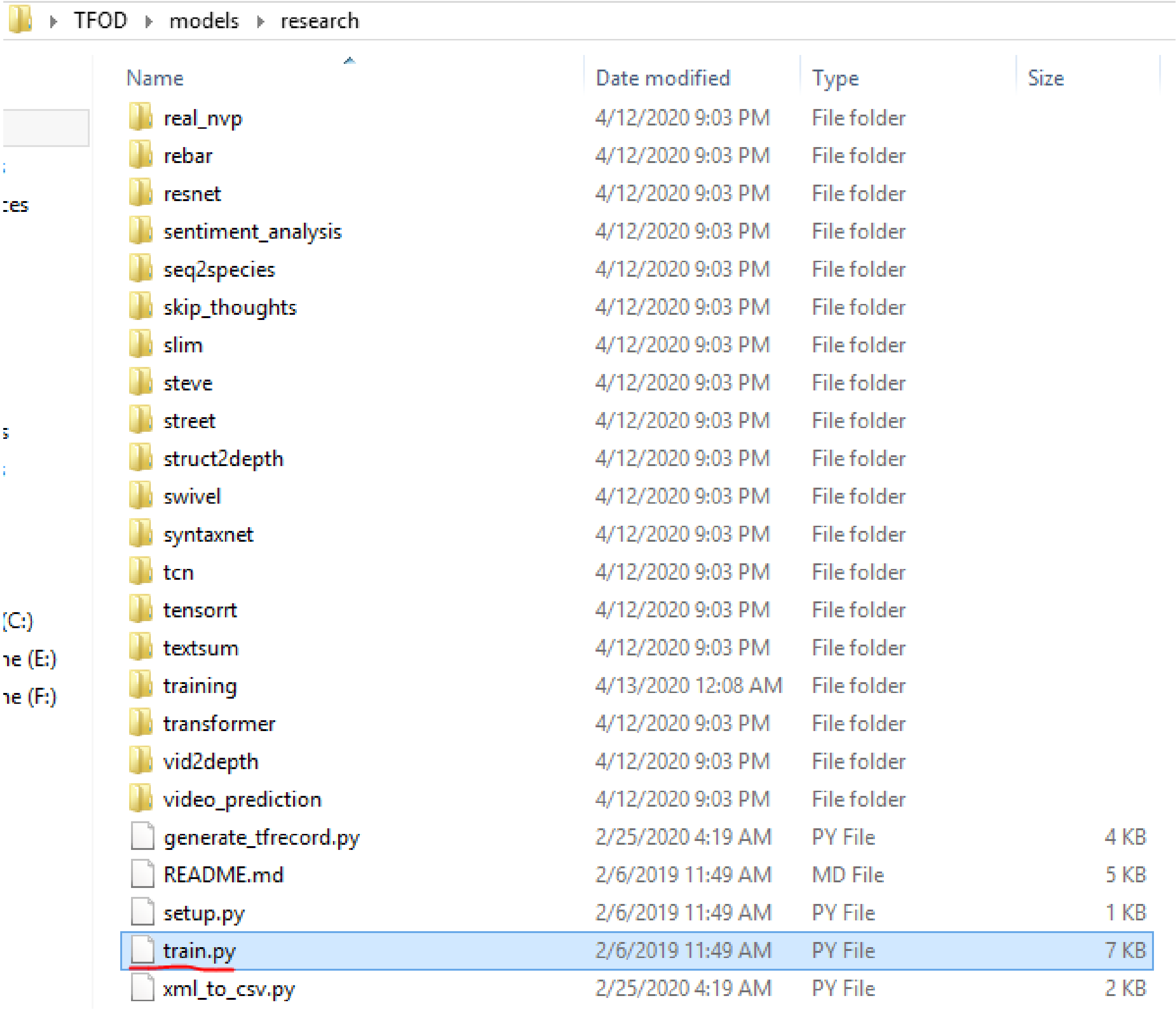


TFOD ▸ models ▸ research ▸ object_detection ▸ legacy

Name	Date modified	Type	Size
__init__.py	2/6/2019 11:49 AM	PY File	0 KB
eval.py	2/6/2019 11:49 AM	PY File	6 KB
evaluator.py	2/6/2019 11:49 AM	PY File	13 KB
train.py	2/6/2019 11:49 AM	PY File	7 KB
trainer.py	2/6/2019 11:49 AM	PY File	18 KB
trainer_test.py	2/6/2019 11:49 AM	PY File	10 KB

This file is useful to start our training process.

So, **copy the file and paste the file into the research** folder.

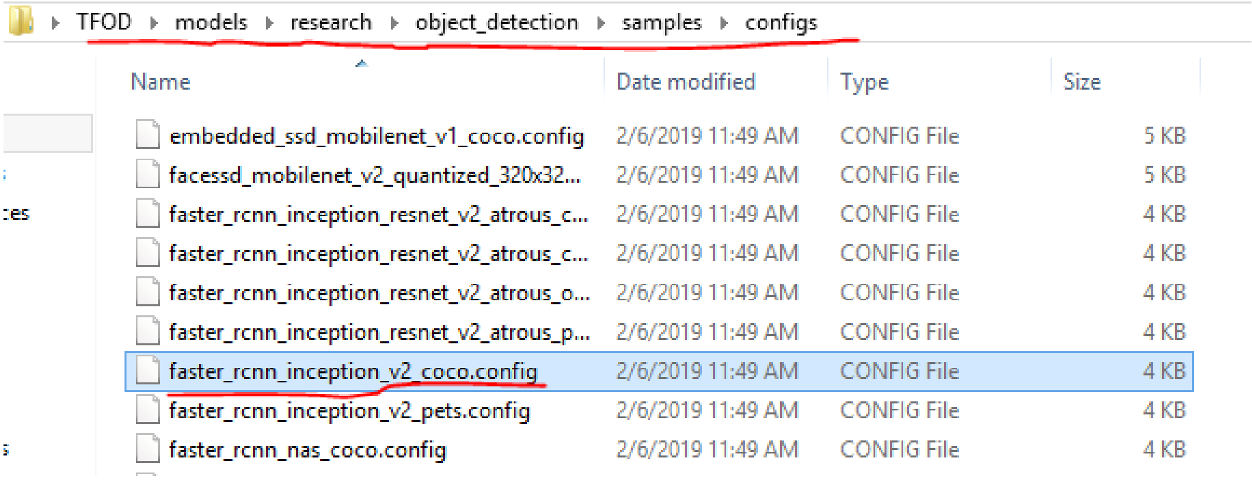


Name	Date modified	Type	Size
real_nvp	4/12/2020 9:03 PM	File folder	
rebar	4/12/2020 9:03 PM	File folder	
resnet	4/12/2020 9:03 PM	File folder	
sentiment_analysis	4/12/2020 9:03 PM	File folder	
seq2species	4/12/2020 9:03 PM	File folder	
skip_thoughts	4/12/2020 9:03 PM	File folder	
slim	4/12/2020 9:03 PM	File folder	
steve	4/12/2020 9:03 PM	File folder	
street	4/12/2020 9:03 PM	File folder	
struct2depth	4/12/2020 9:03 PM	File folder	
swivel	4/12/2020 9:03 PM	File folder	
syntaxnet	4/12/2020 9:03 PM	File folder	
tcn	4/12/2020 9:03 PM	File folder	
tensorrt	4/12/2020 9:03 PM	File folder	
textsum	4/12/2020 9:03 PM	File folder	
training	4/13/2020 12:08 AM	File folder	
transformer	4/12/2020 9:03 PM	File folder	
vid2depth	4/12/2020 9:03 PM	File folder	
video_prediction	4/12/2020 9:03 PM	File folder	
generate_tfrecord.py	2/25/2020 4:19 AM	PY File	4 KB
README.md	2/6/2019 11:49 AM	MD File	5 KB
setup.py	2/6/2019 11:49 AM	PY File	1 KB
train.py	2/6/2019 11:49 AM	PY File	7 KB
xml_to_csv.py	2/25/2020 4:19 AM	PY File	2 KB

config file, so we must move this file also into the research folder --- > training.

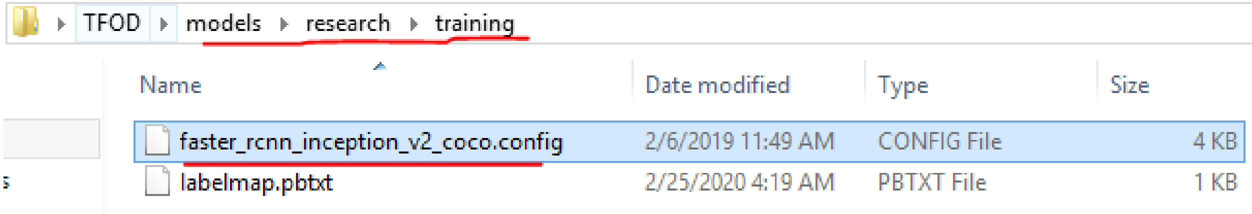
In this faster_rcnn I need one config file and in that file we should mention like for how many steps I would like to train the model and inputs that are giving to that model. So where is config file is situated?

Open TFOD -- > models -- > research -- > object_detection -- > samples -- > **config's** -- > **faster_rcnn_inception_V2_coco.config** -- > copy the file -- > paste in the research folder -- > training folder



TFOD ▸ models ▸ research ▸ object_detection ▸ samples ▸ configs

Name	Date modified	Type	Size
embedded_ssd_mobilenet_v1_coco.config	2/6/2019 11:49 AM	CONFIG File	5 KB
facessd_mobilenet_v2_quantized_320x32...	2/6/2019 11:49 AM	CONFIG File	5 KB
faster_rcnn_inception_resnet_v2_atrous_c...	2/6/2019 11:49 AM	CONFIG File	4 KB
faster_rcnn_inception_resnet_v2_atrous_c...	2/6/2019 11:49 AM	CONFIG File	4 KB
faster_rcnn_inception_resnet_v2_atrous_o...	2/6/2019 11:49 AM	CONFIG File	4 KB
faster_rcnn_inception_resnet_v2_atrous_p...	2/6/2019 11:49 AM	CONFIG File	4 KB
faster_rcnn_inception_v2_coco.config	2/6/2019 11:49 AM	CONFIG File	4 KB
faster_rcnn_inception_v2_pets.config	2/6/2019 11:49 AM	CONFIG File	4 KB
faster_rcnn_nas_coco.config	2/6/2019 11:49 AM	CONFIG File	4 KB

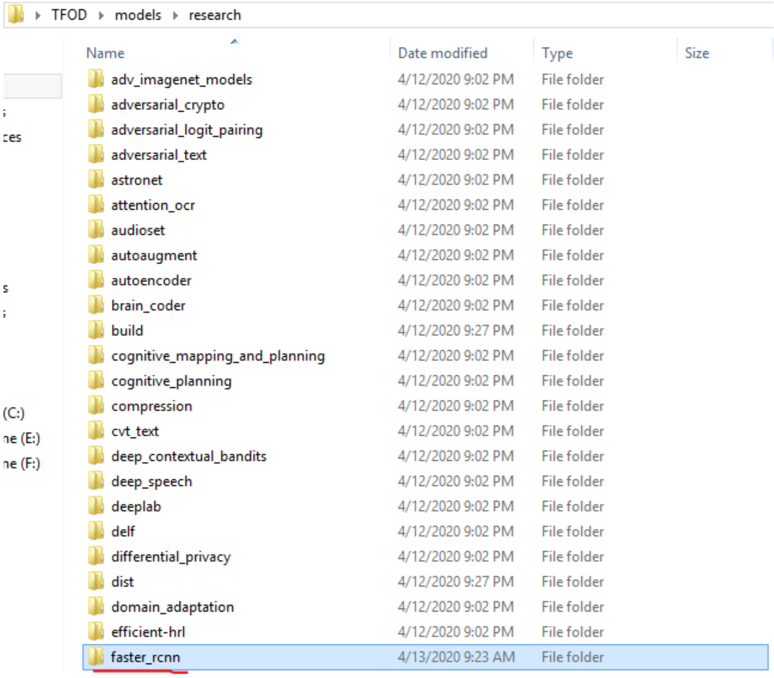


TFOD ▸ models ▸ research ▸ training

Name	Date modified	Type	Size
faster_rcnn_inception_v2_coco.config	2/6/2019 11:49 AM	CONFIG File	4 KB
labelmap.pbtxt	2/25/2020 4:19 AM	PBTXT File	1 KB

For each of the model they will provide a corresponding config file by the TensorFlow community

Now copy the **faster_rcnn** file from TFOD folder and **paste in to the research folder**.



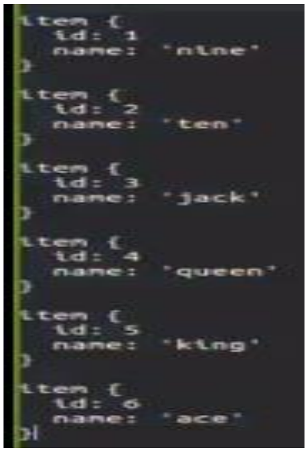
TFOD ▸ models ▸ research

Name	Date modified	Type	Size
adv_imagenet_models	4/12/2020 9:02 PM	File folder	
adversarial_crypto	4/12/2020 9:02 PM	File folder	
adversarial_logit_pairing	4/12/2020 9:02 PM	File folder	
adversarial_text	4/12/2020 9:02 PM	File folder	
astronet	4/12/2020 9:02 PM	File folder	
attention_ocr	4/12/2020 9:02 PM	File folder	
audioset	4/12/2020 9:02 PM	File folder	
autoaugment	4/12/2020 9:02 PM	File folder	
autoencoder	4/12/2020 9:02 PM	File folder	
brain_coder	4/12/2020 9:02 PM	File folder	
build	4/12/2020 9:27 PM	File folder	
cognitive_mapping_and_planning	4/12/2020 9:02 PM	File folder	
cognitive_planning	4/12/2020 9:02 PM	File folder	
compression	4/12/2020 9:02 PM	File folder	
cvt_text	4/12/2020 9:02 PM	File folder	
deep_contextual_bandits	4/12/2020 9:02 PM	File folder	
deep_speech	4/12/2020 9:02 PM	File folder	
deeplab	4/12/2020 9:02 PM	File folder	
delf	4/12/2020 9:02 PM	File folder	
differential_privacy	4/12/2020 9:02 PM	File folder	
dist	4/12/2020 9:27 PM	File folder	
domain_adaptation	4/12/2020 9:02 PM	File folder	
efficient-hrl	4/12/2020 9:02 PM	File folder	
faster_rcnn	4/13/2020 9:23 AM	File folder	

labelmap.pbxt (train and test)

The **labelmap.pbxt** is another important file here we can able to see the different classes.

Tensorflow accepts only .pbxt format



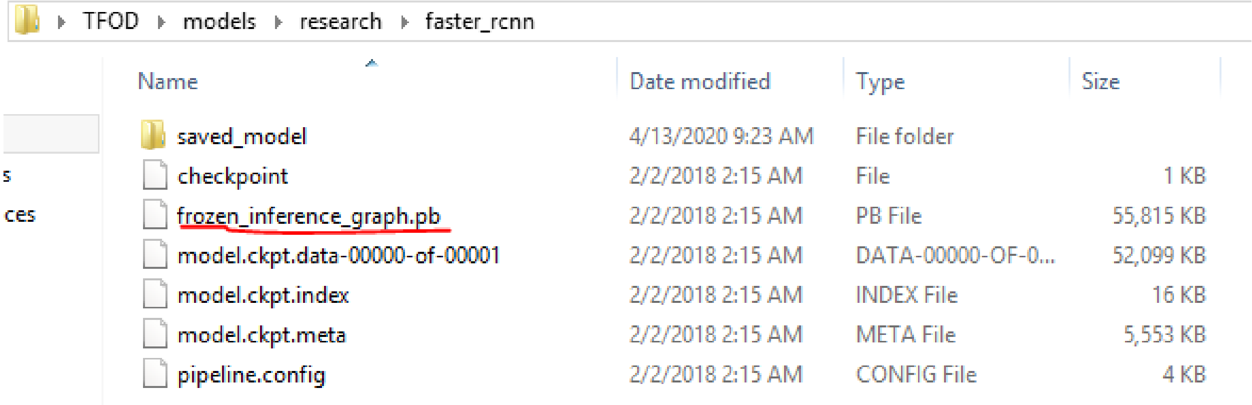
```
item {
    id: 1
    name: 'nine'
}
item {
    id: 2
    name: 'ten'
}
item {
    id: 3
    name: 'jack'
}
item {
    id: 4
    name: 'queen'
}
item {
    id: 5
    name: 'king'
}
item {
    id: 6
    name: 'ace'
}
```

If I have more classes then I need to add those extra classes in to this file (**train and test**). This file also very helpful for prediction purpose.

If u fail to include any classes inside this file then it will not show you the class name or label name.

frozen_inference_graph.pb:

Inside the faster_rcnn the most important file is **frozen_inference_graph.pb.** For tensorflow it uses the file format **.pb.** This file is used for prediction and testing in real time



TFOD ▸ models ▸ research ▸ faster_rcnn

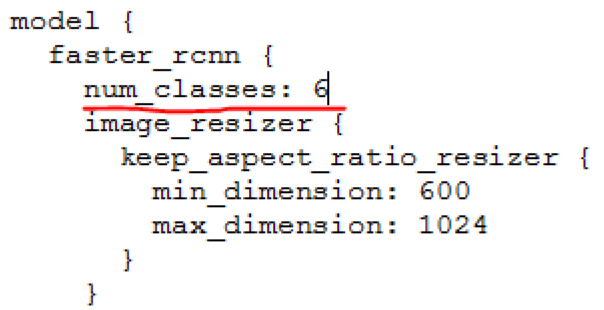
Name	Date modified	Type	Size
saved_model	4/13/2020 9:23 AM	File folder	
checkpoint	2/2/2018 2:15 AM	File	1 KB
frozen_inference_graph.pb	2/2/2018 2:15 AM	PB File	55,815 KB
model.ckpt.data-00000-of-00001	2/2/2018 2:15 AM	DATA-00000-OF-0...	52,099 KB
model.ckpt.index	2/2/2018 2:15 AM	INDEX File	16 KB
model.ckpt.meta	2/2/2018 2:15 AM	META File	5,553 KB
pipeline.config	2/2/2018 2:15 AM	CONFIG File	4 KB

Now we to do some changes on the **faster_rcnn_inception_v2_coco.config** file.

In this config file I have to mention few parameters. We need to 6 changes in that file.

1st change:

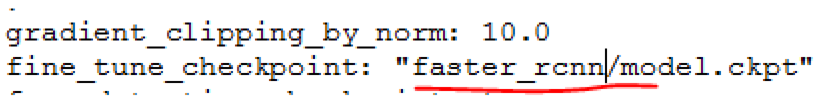
By default in the num_classes(COCO Dataset) we have 90 and change into 6 because I have 6 classes only



```
model {
  faster_rcnn {
    num_classes: 6
    image_resizer {
      keep_aspect_ratio_resizer {
        min_dimension: 600
        max_dimension: 1024
      }
    }
```

2nd change:

Models:

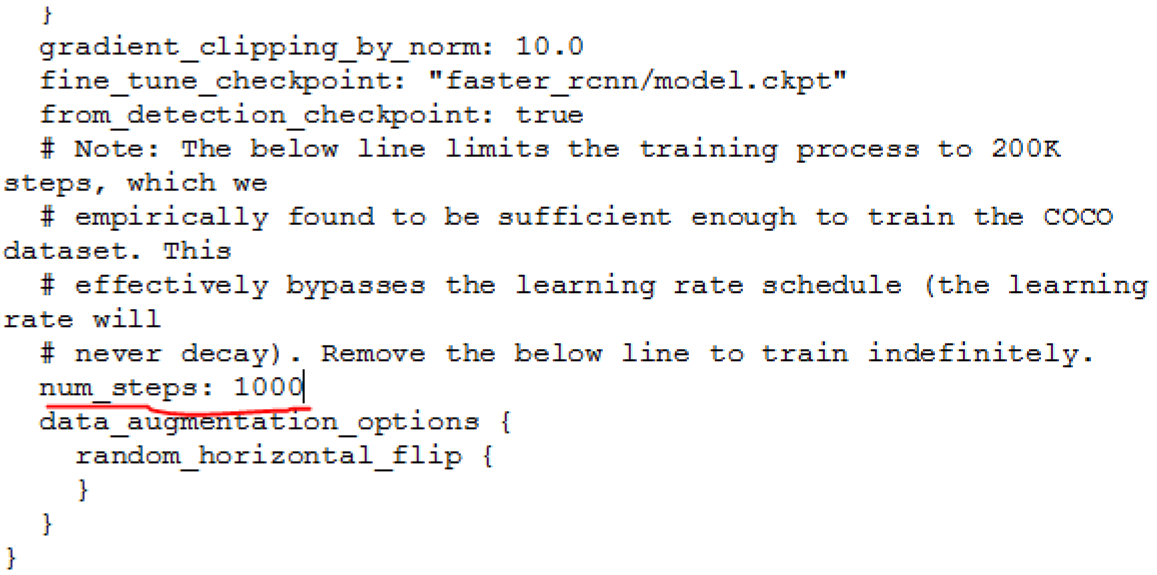


```
gradient_clipping_by_norm: 10.0
fine_tune_checkpoint: "faster_rcnn/model.ckpt"
```

“faster_rcnn” folder name

3rd change:

Epochs:

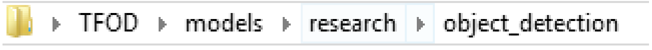


```
  }
  gradient_clipping_by_norm: 10.0
  fine_tune_checkpoint: "faster_rcnn/model.ckpt"
  from_detection_checkpoint: true
  # Note: The below line limits the training process to 200K
steps, which we
  # empirically found to be sufficient enough to train the COCO
dataset. This
  # effectively bypasses the learning rate schedule (the learning
rate will
  # never decay). Remove the below line to train indefinitely.
  num_steps: 1000
  data_augmentation_options {
    random_horizontal_flip {
    }
  }
}
```

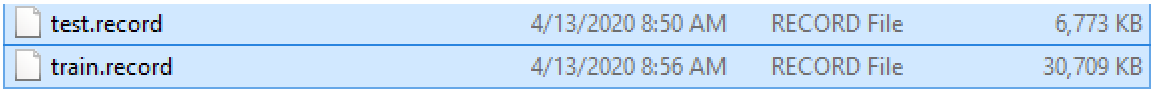
4th change:

Give the path of Tf records

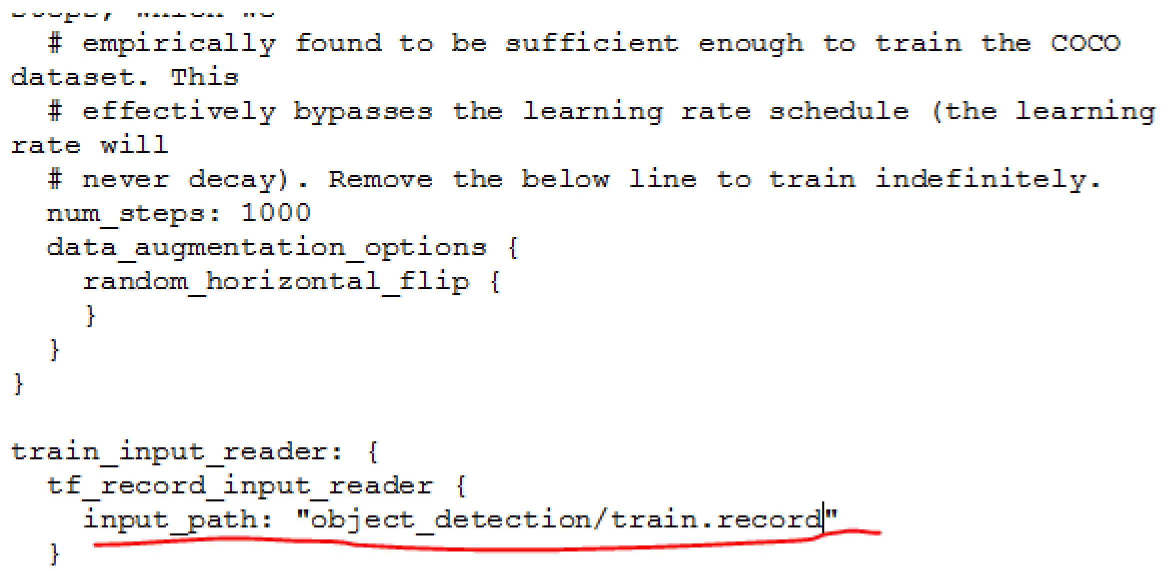
Those records are available in the object detection folder



TFOD ▸ models ▸ research ▸ object_detection



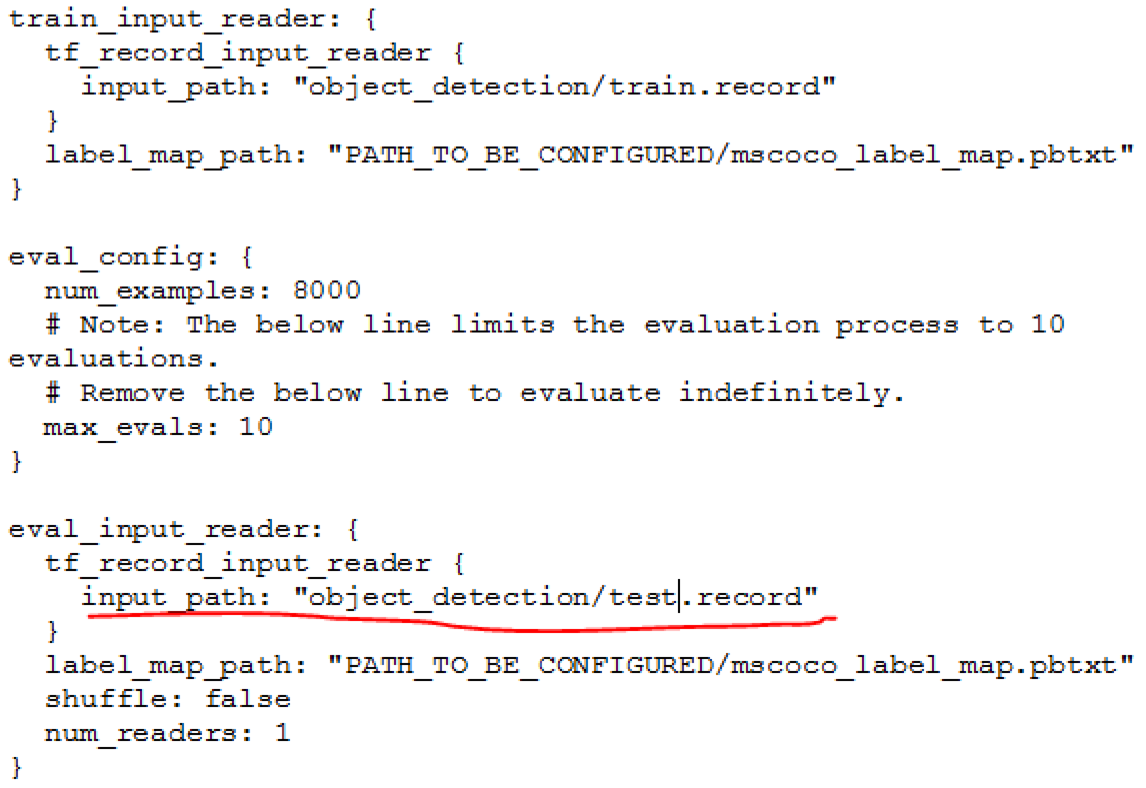
test.record	4/13/2020 8:50 AM	RECORD File	6,773 KB
train.record	4/13/2020 8:56 AM	RECORD File	30,709 KB



```
  # empirically found to be sufficient enough to train the COCO
dataset. This
  # effectively bypasses the learning rate schedule (the learning
rate will
  # never decay). Remove the below line to train indefinitely.
  num_steps: 1000
  data_augmentation_options {
    random_horizontal_flip {
    }
  }
}

train_input_reader: {
  tf_record_input_reader {
    input_path: "object_detection/train.record"
  }
```

5th Change

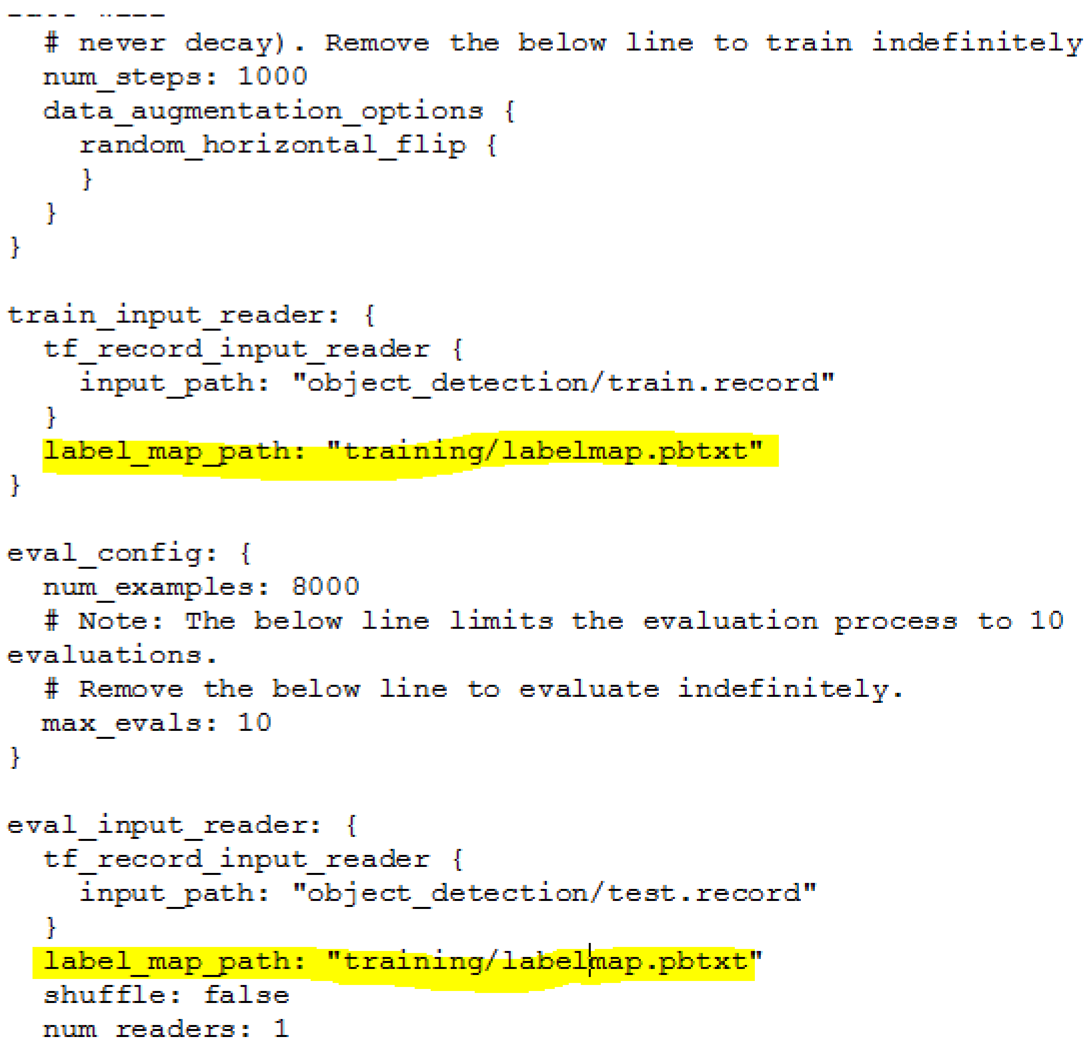
Change the path of the Test tf record:



```
train_input_reader: {
  tf_record_input_reader {
    input_path: "object_detection/train.record"
  }
  label_map_path: "PATH_TO_BE_CONFIGURED/mscoco_label_map.pbtxt"
}

eval_config: {
  num_examples: 8000
  # Note: The below line limits the evaluation process to 10
evaluations.
  # Remove the below line to evaluate indefinitely.
  max_evals: 10
}

eval_input_reader: {
  tf_record_input_reader {
    input_path: "object_detection/test.record"
  }
  label_map_path: "PATH_TO_BE_CONFIGURED/mscoco_label_map.pbtxt"
  shuffle: false
  num_readers: 1
}
```

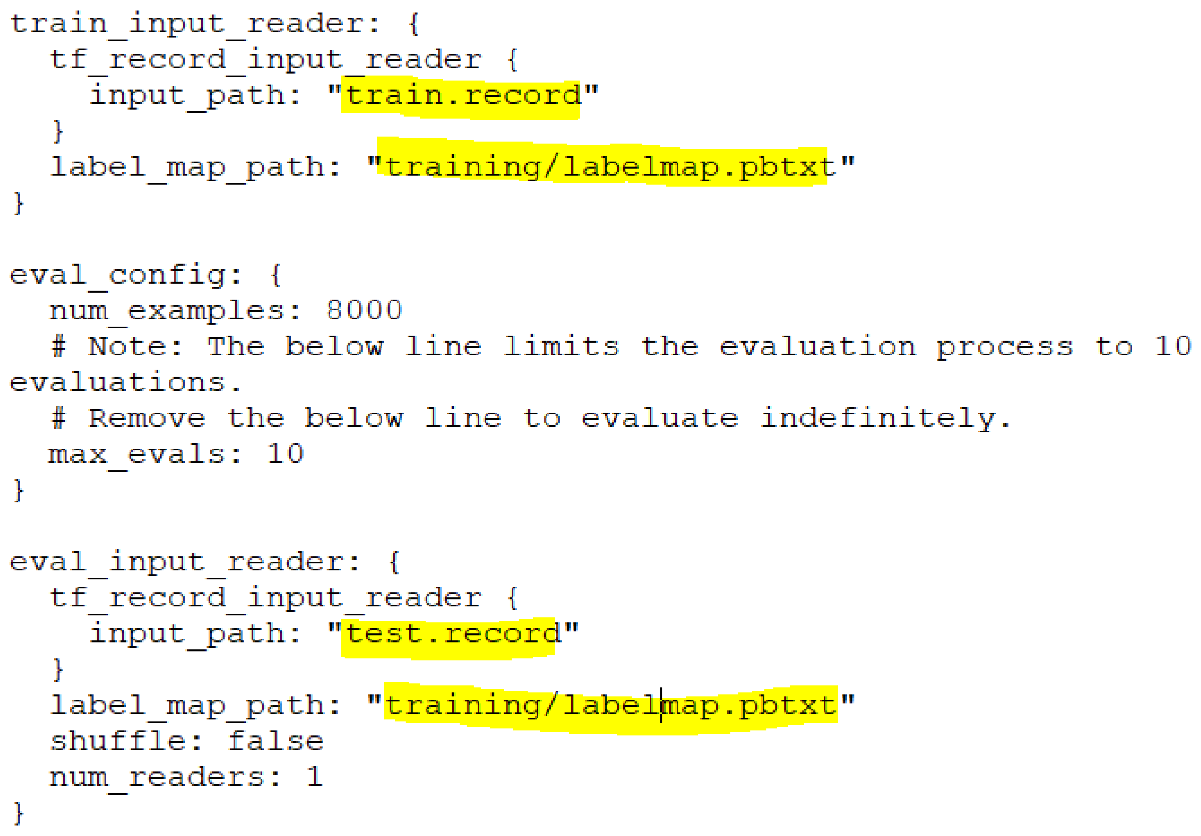
6th change

Now we need to change the **label_map_path**



```
    # never decay). Remove the below line to train indefinitely
    num_steps: 1000
    data_augmentation_options {
      random_horizontal_flip {
      }
    }
  }

  train_input_reader: {
    tf_record_input_reader {
      input_path: "object_detection/train.record"
    }
    label_map_path: "training/labelmap.pbtxt"
  }

  eval_config: {
    num_examples: 8000
    # Note: The below line limits the evaluation process to 10
  evaluations.
    # Remove the below line to evaluate indefinitely.
    max_evals: 10
  }

  eval_input_reader: {
    tf_record_input_reader {
      input_path: "object_detection/test.record"
    }
    label_map_path: "training/labelmap.pbtxt"
    shuffle: false
    num readers: 1
```

Not object_detection/train and test refer the below one



```
train_input_reader: {
  tf_record_input_reader {
    input_path: "train.record"
  }
  label_map_path: "training/labelmap.pbtxt"
}

eval_config: {
  num_examples: 8000
  # Note: The below line limits the evaluation process to 10
evaluations.
  # Remove the below line to evaluate indefinitely.
  max_evals: 10
}

eval_input_reader: {
  tf_record_input_reader {
    input_path: "test.record"
  }
  label_map_path: "training/labelmap.pbtxt"
  shuffle: false
  num_readers: 1
}
```

Finally save the file.

I have done my changes on the **faster_rcnn_inception_v2_coco.config** file.

Start the Training:

Go to the command prompt and run this command under this location TFOD -- > models -- > research -- > below command

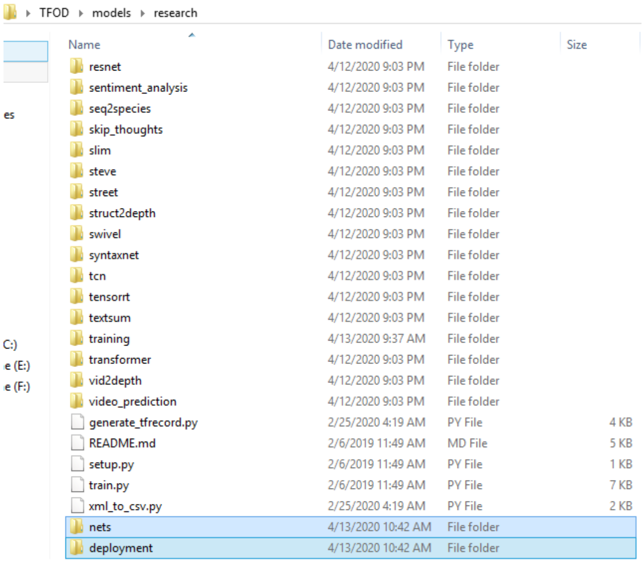
python train.py --logtostderr --train_dir=training/ --pipeline_config_path=training/faster_rcnn_inception_v2_coco.config

command : `python train.py --logtostderr --train_dir=training/ --pipeline_config_path=training/YOUR_MODEL.config`

After running this command you may get an error like **"No module named nets".**

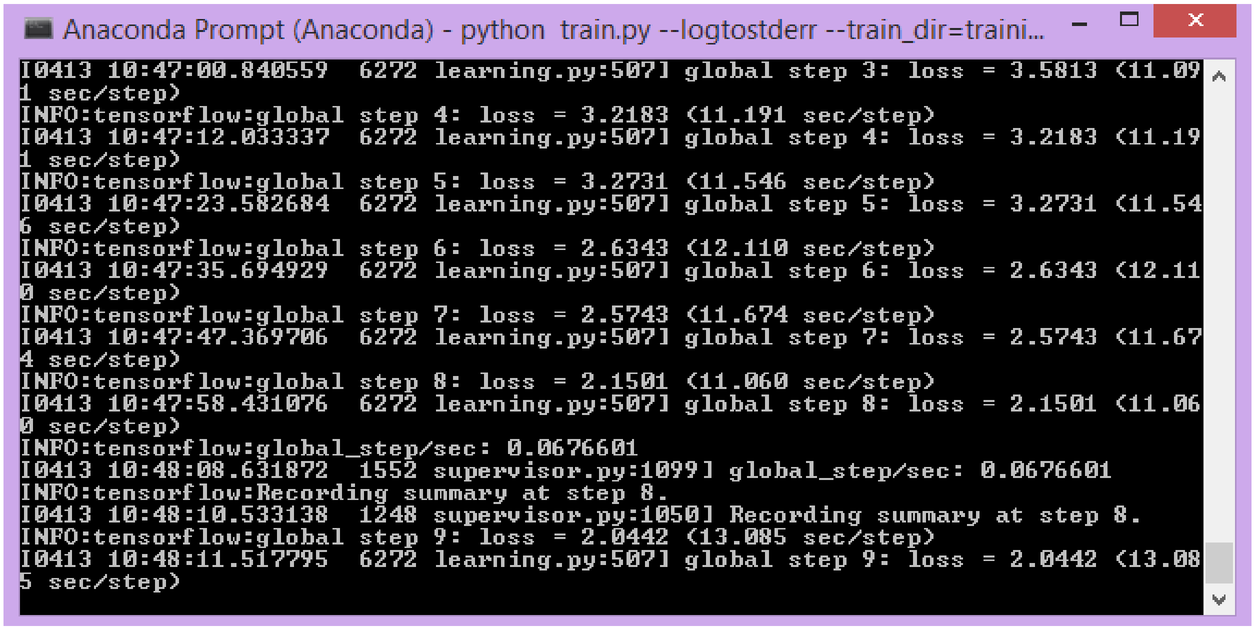
To solve this issue we have a folder called **slim** under research folder.

Now **copy the two folders like deployment and nets** then paste into the research folder.



Name	Date modified	Type	Size
resnet	4/12/2020 9:03 PM	File folder	
sentiment_analysis	4/12/2020 9:03 PM	File folder	
seq2species	4/12/2020 9:03 PM	File folder	
skip_thoughts	4/12/2020 9:03 PM	File folder	
slim	4/12/2020 9:03 PM	File folder	
steve	4/12/2020 9:03 PM	File folder	
street	4/12/2020 9:03 PM	File folder	
struct2depth	4/12/2020 9:03 PM	File folder	
swivel	4/12/2020 9:03 PM	File folder	
syntaxnet	4/12/2020 9:03 PM	File folder	
tcn	4/12/2020 9:03 PM	File folder	
tensorrt	4/12/2020 9:03 PM	File folder	
textsum	4/12/2020 9:03 PM	File folder	
training	4/13/2020 9:37 AM	File folder	
transformer	4/12/2020 9:03 PM	File folder	
vid2depth	4/12/2020 9:03 PM	File folder	
video_prediction	4/12/2020 9:03 PM	File folder	
generate_tfrecord.py	2/25/2020 4:19 AM	PY File	4 KB
README.md	2/6/2019 11:49 AM	MD File	5 KB
setup.py	2/6/2019 11:49 AM	PY File	1 KB
train.py	2/6/2019 11:49 AM	PY File	7 KB
xml_to_csv.py	2/25/2020 4:19 AM	PY File	2 KB
nets	4/13/2020 10:42 AM	File folder	
deployment	4/13/2020 10:42 AM	File folder	

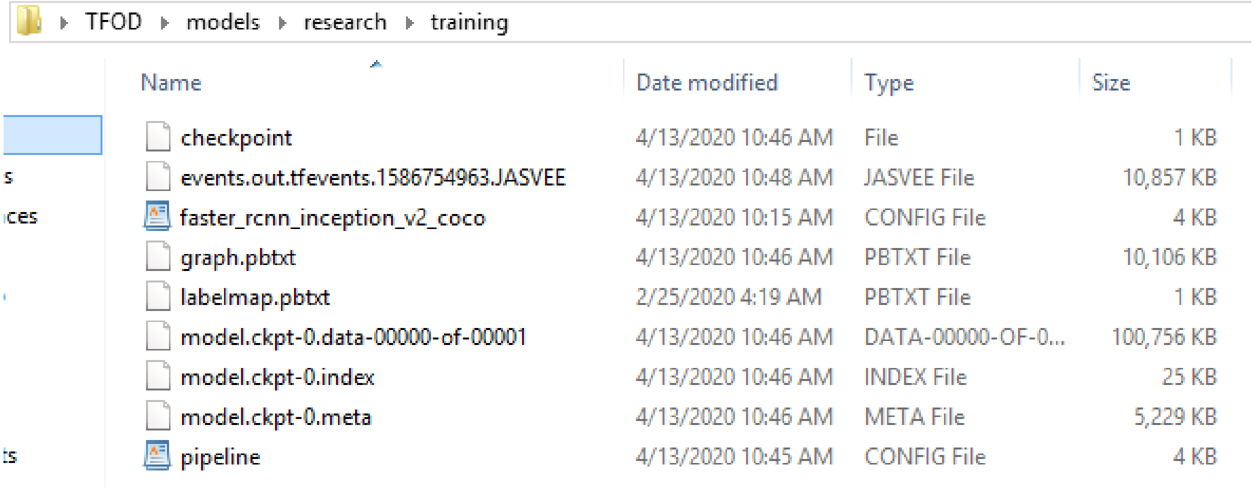
Now go the command prompt and run the same command



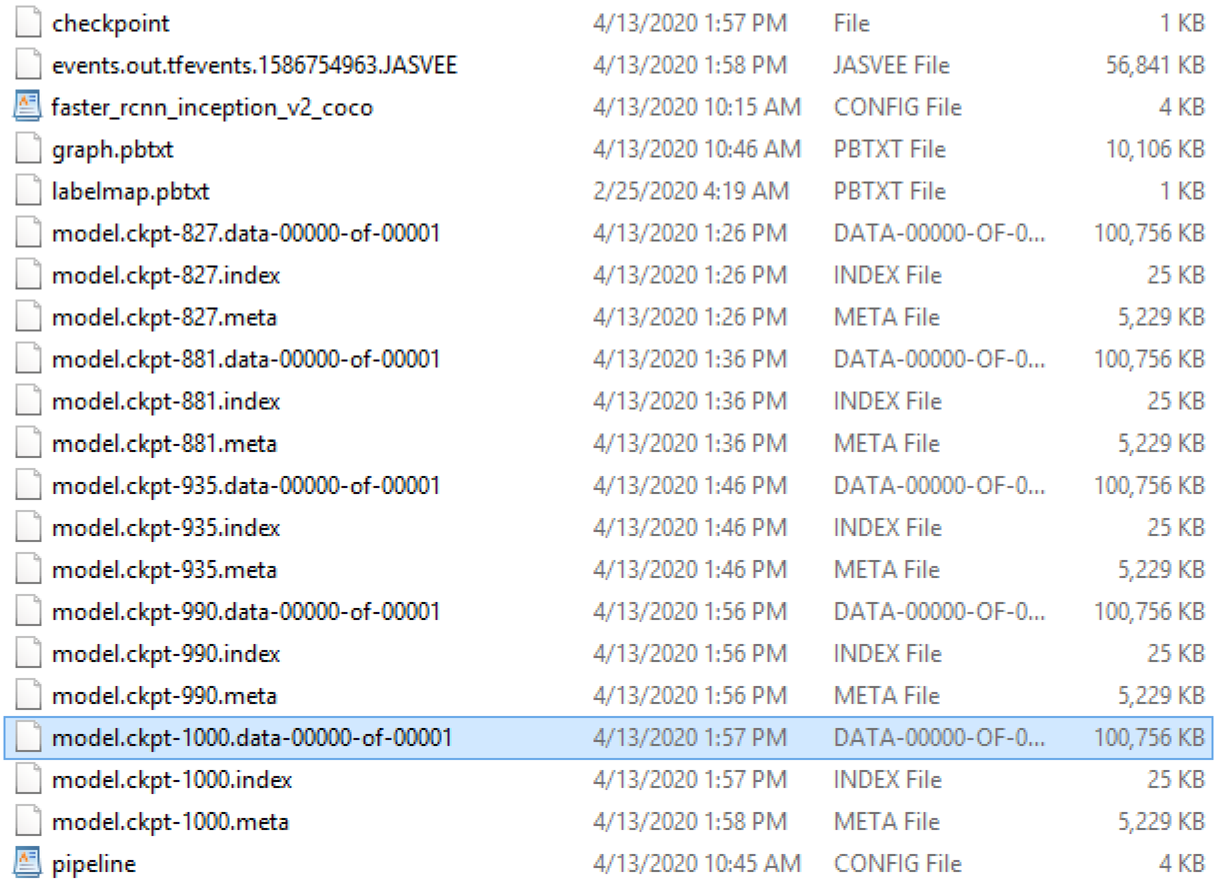
Anaconda Prompt (Anaconda) - python train.py --logtostderr --train_dir=traini...

```
I0413 10:47:00.840559  6272 learning.py:507] global step 3: loss = 3.5813 (11.09
1 sec/step)
INFO:tensorflow:global step 4: loss = 3.2183 (11.191 sec/step)
I0413 10:47:12.033337  6272 learning.py:507] global step 4: loss = 3.2183 (11.19
1 sec/step)
INFO:tensorflow:global step 5: loss = 3.2731 (11.546 sec/step)
I0413 10:47:23.582684  6272 learning.py:507] global step 5: loss = 3.2731 (11.54
6 sec/step)
INFO:tensorflow:global step 6: loss = 2.6343 (12.110 sec/step)
I0413 10:47:35.694929  6272 learning.py:507] global step 6: loss = 2.6343 (12.11
0 sec/step)
INFO:tensorflow:global step 7: loss = 2.5743 (11.674 sec/step)
I0413 10:47:47.369706  6272 learning.py:507] global step 7: loss = 2.5743 (11.67
4 sec/step)
INFO:tensorflow:global step 8: loss = 2.1501 (11.060 sec/step)
I0413 10:47:58.431076  6272 learning.py:507] global step 8: loss = 2.1501 (11.06
0 sec/step)
INFO:tensorflow:global_step/sec: 0.0676601
I0413 10:48:08.631872  1552 supervisor.py:1099] global_step/sec: 0.0676601
INFO:tensorflow:Recording summary at step 8.
I0413 10:48:10.533138  1248 supervisor.py:1050] Recording summary at step 8.
INFO:tensorflow:global step 9: loss = 2.0442 (13.085 sec/step)
I0413 10:48:11.517795  6272 learning.py:507] global step 9: loss = 2.0442 (13.08
5 sec/step)
```

Now if you go to the training folder you will be able to see some extra files which means the training has started.



Name	Date modified	Type	Size
checkpoint	4/13/2020 10:46 AM	File	1 KB
events.out.tfevents.1586754963.JASVEE	4/13/2020 10:48 AM	JASVEE File	10,857 KB
faster_rcnn_inception_v2_coco	4/13/2020 10:15 AM	CONFIG File	4 KB
graph.pbtxt	4/13/2020 10:46 AM	PBTXT File	10,106 KB
labelmap.pbtxt	2/25/2020 4:19 AM	PBTXT File	1 KB
model.ckpt-0.data-00000-of-00001	4/13/2020 10:46 AM	DATA-00000-OF-0...	100,756 KB
model.ckpt-0.index	4/13/2020 10:46 AM	INDEX File	25 KB
model.ckpt-0.meta	4/13/2020 10:46 AM	META File	5,229 KB
pipeline	4/13/2020 10:45 AM	CONFIG File	4 KB

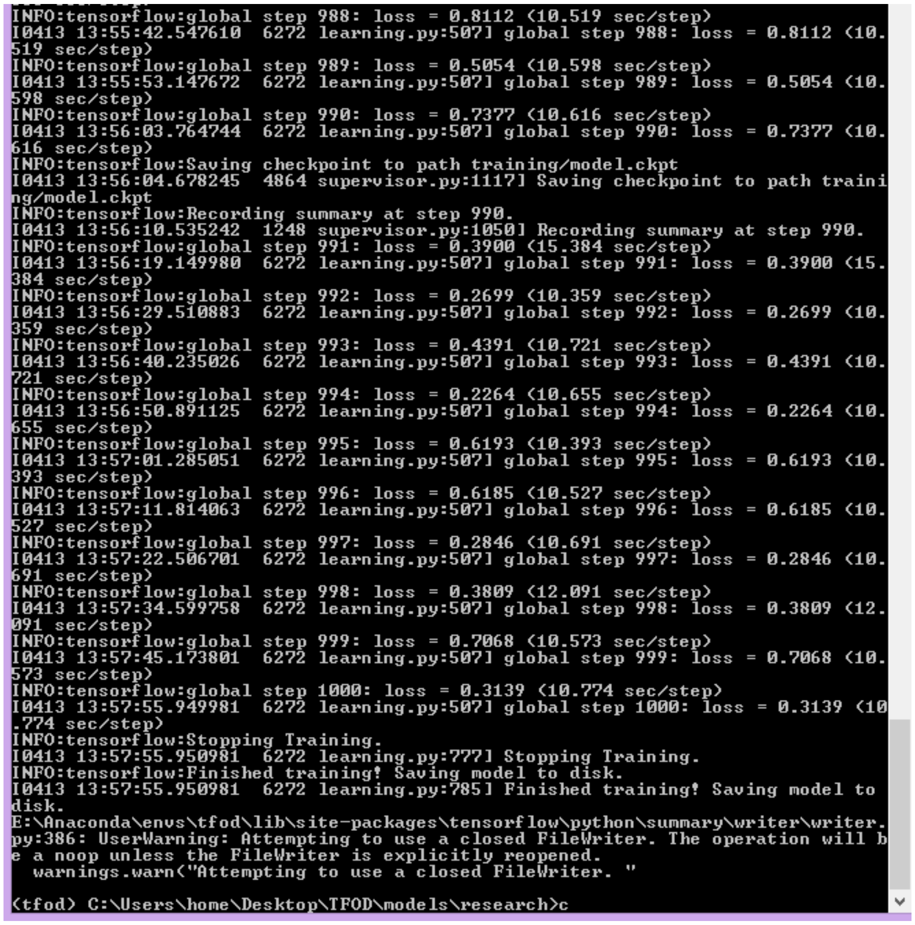


checkpoint	4/13/2020 1:57 PM	File	1 KB
events.out.tfevents.1586754963.JASVEE	4/13/2020 1:58 PM	JASVEE File	56,841 KB
faster_rcnn_inception_v2_coco	4/13/2020 10:15 AM	CONFIG File	4 KB
graph.pbtxt	4/13/2020 10:46 AM	PBTXT File	10,106 KB
labelmap.pbtxt	2/25/2020 4:19 AM	PBTXT File	1 KB
model.ckpt-827.data-00000-of-00001	4/13/2020 1:26 PM	DATA-00000-OF-0...	100,756 KB
model.ckpt-827.index	4/13/2020 1:26 PM	INDEX File	25 KB
model.ckpt-827.meta	4/13/2020 1:26 PM	META File	5,229 KB
model.ckpt-881.data-00000-of-00001	4/13/2020 1:36 PM	DATA-00000-OF-0...	100,756 KB
model.ckpt-881.index	4/13/2020 1:36 PM	INDEX File	25 KB
model.ckpt-881.meta	4/13/2020 1:36 PM	META File	5,229 KB
model.ckpt-935.data-00000-of-00001	4/13/2020 1:46 PM	DATA-00000-OF-0...	100,756 KB
model.ckpt-935.index	4/13/2020 1:46 PM	INDEX File	25 KB
model.ckpt-935.meta	4/13/2020 1:46 PM	META File	5,229 KB
model.ckpt-990.data-00000-of-00001	4/13/2020 1:56 PM	DATA-00000-OF-0...	100,756 KB
model.ckpt-990.index	4/13/2020 1:56 PM	INDEX File	25 KB
model.ckpt-990.meta	4/13/2020 1:56 PM	META File	5,229 KB
model.ckpt-1000.data-00000-of-00001	4/13/2020 1:57 PM	DATA-00000-OF-0...	100,756 KB
model.ckpt-1000.index	4/13/2020 1:57 PM	INDEX File	25 KB
model.ckpt-1000.meta	4/13/2020 1:58 PM	META File	5,229 KB
pipeline	4/13/2020 10:45 AM	CONFIG File	4 KB

Ckpt – checkpoint file

In the command prompt suppose if the loss is stable and not changing after some epochs, in that case you can stop your training process by pressing interrupt ctrl+c.

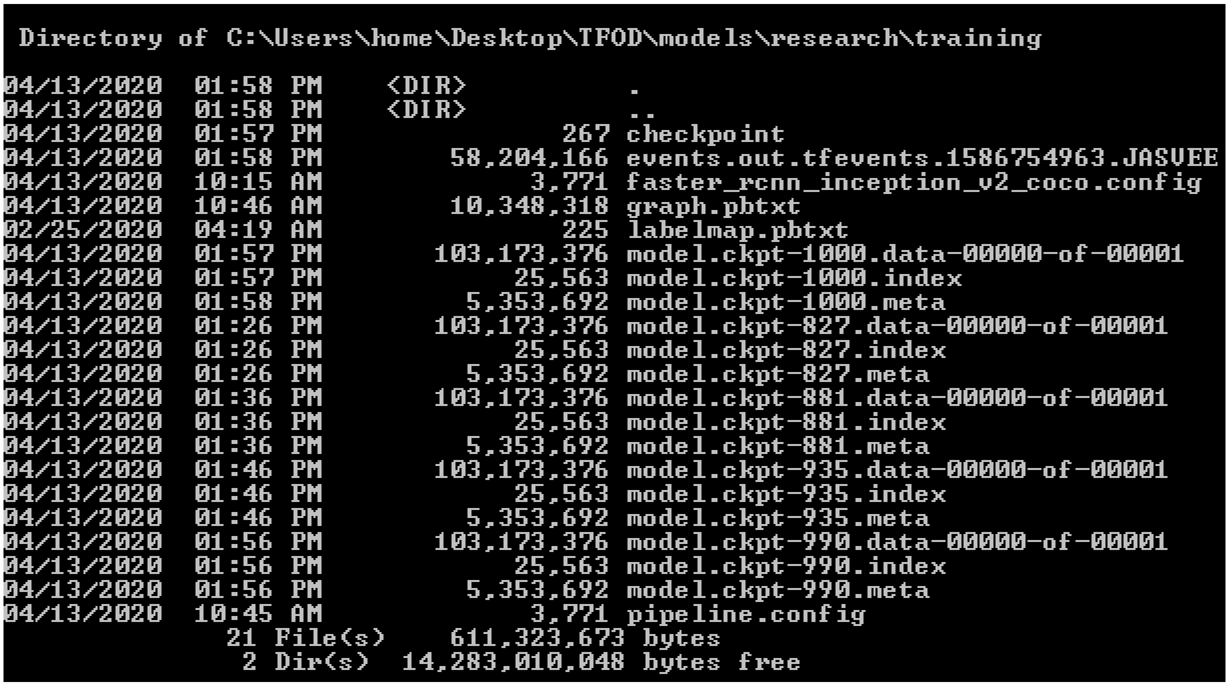
Then execute the same command then it will start from last saved checkpoint.



```
INFO:tensorflow:global step 988: loss = 0.8112 (10.519 sec/step)
I0413 13:55:42.547610  6272 learning.py:507] global step 988: loss = 0.8112 (10.
519 sec/step)
INFO:tensorflow:global step 989: loss = 0.5054 (10.598 sec/step)
I0413 13:55:53.147672  6272 learning.py:507] global step 989: loss = 0.5054 (10.
598 sec/step)
INFO:tensorflow:global step 990: loss = 0.7377 (10.616 sec/step)
I0413 13:56:03.764744  6272 learning.py:507] global step 990: loss = 0.7377 (10.
616 sec/step)
INFO:tensorflow:Saving checkpoint to path training/model.ckpt
I0413 13:56:04.678245  4864 supervisor.py:1117] Saving checkpoint to path traini
ng/model.ckpt
INFO:tensorflow:Recording summary at step 990.
I0413 13:56:10.535242  1248 supervisor.py:1050] Recording summary at step 990.
INFO:tensorflow:global step 991: loss = 0.3900 (15.384 sec/step)
I0413 13:56:19.149980  6272 learning.py:507] global step 991: loss = 0.3900 (15.
384 sec/step)
INFO:tensorflow:global step 992: loss = 0.2699 (10.359 sec/step)
I0413 13:56:29.510883  6272 learning.py:507] global step 992: loss = 0.2699 (10.
359 sec/step)
INFO:tensorflow:global step 993: loss = 0.4391 (10.721 sec/step)
I0413 13:56:40.235026  6272 learning.py:507] global step 993: loss = 0.4391 (10.
721 sec/step)
INFO:tensorflow:global step 994: loss = 0.2264 (10.655 sec/step)
I0413 13:56:50.891125  6272 learning.py:507] global step 994: loss = 0.2264 (10.
655 sec/step)
INFO:tensorflow:global step 995: loss = 0.6193 (10.393 sec/step)
I0413 13:57:01.285051  6272 learning.py:507] global step 995: loss = 0.6193 (10.
393 sec/step)
INFO:tensorflow:global step 996: loss = 0.6185 (10.527 sec/step)
I0413 13:57:11.814063  6272 learning.py:507] global step 996: loss = 0.6185 (10.
527 sec/step)
INFO:tensorflow:global step 997: loss = 0.2846 (10.691 sec/step)
I0413 13:57:22.506701  6272 learning.py:507] global step 997: loss = 0.2846 (10.
691 sec/step)
INFO:tensorflow:global step 998: loss = 0.3809 (12.091 sec/step)
I0413 13:57:34.599758  6272 learning.py:507] global step 998: loss = 0.3809 (12.
091 sec/step)
INFO:tensorflow:global step 999: loss = 0.7068 (10.573 sec/step)
I0413 13:57:45.173801  6272 learning.py:507] global step 999: loss = 0.7068 (10.
573 sec/step)
INFO:tensorflow:global step 1000: loss = 0.3139 (10.774 sec/step)
I0413 13:57:55.949981  6272 learning.py:507] global step 1000: loss = 0.3139 (10
.774 sec/step)
INFO:tensorflow:Stopping Training.
I0413 13:57:55.950981  6272 learning.py:777] Stopping Training.
INFO:tensorflow:Finished training! Saving model to disk.
I0413 13:57:55.950981  6272 learning.py:785] Finished training! Saving model to
disk.
E:\Anaconda\envs\tfod\lib\site-packages\tensorflow\python\summary\writer\writer.
py:386: UserWarning: Attempting to use a closed FileWriter. The operation will b
e a noop unless the FileWriter is explicitly reopened.
  warnings.warn("Attempting to use a closed FileWriter. "

(tfod) C:\Users\home\Desktop\TFOD\models\research>c
```

Once 1000 epochs done or you interrupt in between the epochs.

Now go to the training folder



```
 Directory of C:\Users\home\Desktop\TFOD\models\research\training

04/13/2020  01:58 PM    <DIR>          .
04/13/2020  01:58 PM    <DIR>          ..
04/13/2020  01:57 PM               267 checkpoint
04/13/2020  01:58 PM        58,204,166 events.out.tfevents.1586754963.JASVEE
04/13/2020  10:15 AM             3,771 faster_rcnn_inception_v2_coco.config
04/13/2020  10:46 AM        10,348,318 graph.pbtxt
02/25/2020  04:19 AM               225 labelmap.pbtxt
04/13/2020  01:57 PM       103,173,376 model.ckpt-1000.data-00000-of-00001
04/13/2020  01:57 PM            25,563 model.ckpt-1000.index
04/13/2020  01:58 PM         5,353,692 model.ckpt-1000.meta
04/13/2020  01:26 PM       103,173,376 model.ckpt-827.data-00000-of-00001
04/13/2020  01:26 PM            25,563 model.ckpt-827.index
04/13/2020  01:26 PM         5,353,692 model.ckpt-827.meta
04/13/2020  01:36 PM       103,173,376 model.ckpt-881.data-00000-of-00001
04/13/2020  01:36 PM            25,563 model.ckpt-881.index
04/13/2020  01:36 PM         5,353,692 model.ckpt-881.meta
04/13/2020  01:46 PM       103,173,376 model.ckpt-935.data-00000-of-00001
04/13/2020  01:46 PM            25,563 model.ckpt-935.index
04/13/2020  01:46 PM         5,353,692 model.ckpt-935.meta
04/13/2020  01:56 PM       103,173,376 model.ckpt-990.data-00000-of-00001
04/13/2020  01:56 PM            25,563 model.ckpt-990.index
04/13/2020  01:56 PM         5,353,692 model.ckpt-990.meta
04/13/2020  10:45 AM             3,771 pipeline.config
              21 File(s)    611,323,673 bytes
               2 Dir(s)  14,283,010,048 bytes free
```

Here the latest one is generated here is model.ckpt-1000, perhaps if you interrupt in between any epochs then you need to take the latest one.

So I can take the latest file and I used to create a Pb model, because for prediction we can't use the ckpt file we have to use .pb file

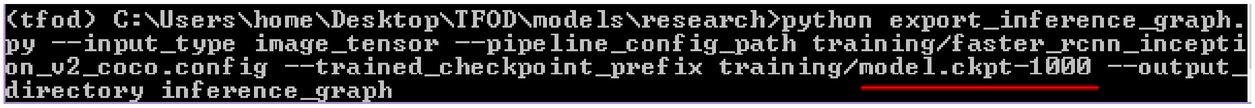
To do the conversion of .pb file one more file is needed and which is available in the object detection folder -- > **export_inference_graph.py** copy the file and paste into the research folder.

Now open the command prompt and run this command to convert .ckpt to .pb file

python export_inference_graph.py --input_type image_tensor --pipeline_config_path training/faster_rcnn_inception_v2_coco.config --trained_checkpoint_prefix training/model.ckpt-438 --output_directory inference_graph

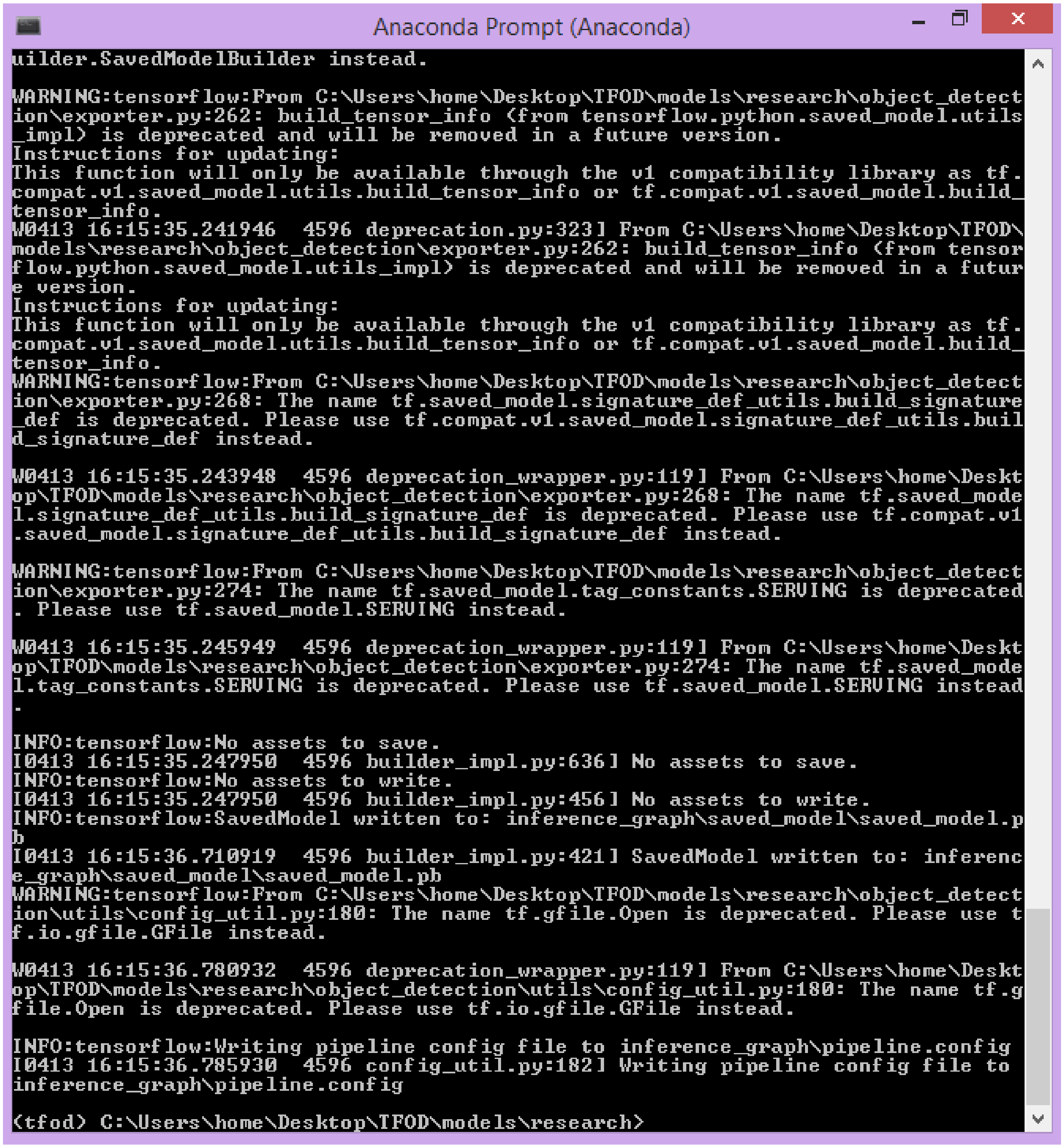
instead of inference_graph you can give any name like my_model etc.

Suppose if you use the latest cpkt means then just edit the model.ckpt.your latest ckpt no then run the command.



```
(tfod) C:\Users\home\Desktop\TFOD\models\research>python export_inference_graph.
py --input_type image_tensor --pipeline_config_path training/faster_rcnn_incepti
on_v2_coco.config --trained_checkpoint_prefix training/model.ckpt-1000 --output_
directory inference_graph
```

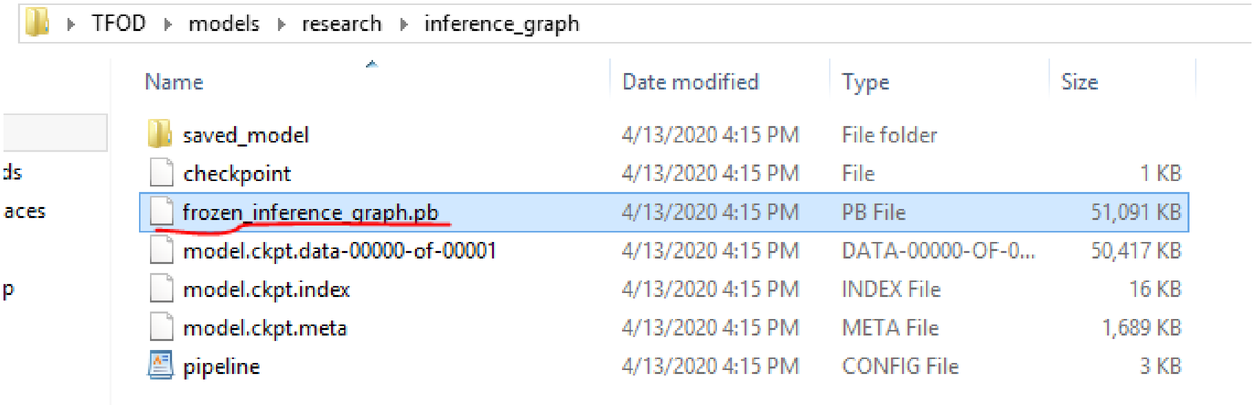
Here I'm using my latest ckpt file (model.ckpt-1000)



```
uilder.SavedModelBuilder instead.

WARNING:tensorflow:From C:\Users\home\Desktop\TFOD\models\research\object_detect
ion\exporter.py:262: build_tensor_info (from tensorflow.python.saved_model.utils
_impl) is deprecated and will be removed in a future version.
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.build_
tensor_info.
W0413 16:15:35.241946  4596 deprecation.py:323] From C:\Users\home\Desktop\TFOD\
models\research\object_detection\exporter.py:262: build_tensor_info (from tensor
flow.python.saved_model.utils_impl) is deprecated and will be removed in a futur
e version.
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.build_
tensor_info.
WARNING:tensorflow:From C:\Users\home\Desktop\TFOD\models\research\object_detect
ion\exporter.py:268: The name tf.saved_model.signature_def_utils.build_signature
_def is deprecated. Please use tf.compat.v1.saved_model.signature_def_utils.buil
d_signature_def instead.

W0413 16:15:35.243948  4596 deprecation_wrapper.py:119] From C:\Users\home\Deskt
op\TFOD\models\research\object_detection\exporter.py:268: The name tf.saved_mode
l.signature_def_utils.build_signature_def is deprecated. Please use tf.compat.v1
.saved_model.signature_def_utils.build_signature_def instead.

WARNING:tensorflow:From C:\Users\home\Desktop\TFOD\models\research\object_detect
ion\exporter.py:274: The name tf.saved_model.tag_constants.SERVING is deprecated
. Please use tf.saved_model.SERVING instead.

W0413 16:15:35.245949  4596 deprecation_wrapper.py:119] From C:\Users\home\Deskt
op\TFOD\models\research\object_detection\exporter.py:274: The name tf.saved_mode
l.tag_constants.SERVING is deprecated. Please use tf.saved_model.SERVING instead
.

INFO:tensorflow:No assets to save.
I0413 16:15:35.247950  4596 builder_impl.py:636] No assets to save.
INFO:tensorflow:No assets to write.
I0413 16:15:35.247950  4596 builder_impl.py:456] No assets to write.
INFO:tensorflow:SavedModel written to: inference_graph\saved_model\saved_model.p
b
I0413 16:15:36.710919  4596 builder_impl.py:421] SavedModel written to: inferenc
e_graph\saved_model\saved_model.pb
WARNING:tensorflow:From C:\Users\home\Desktop\TFOD\models\research\object_detect
ion\utils\config_util.py:180: The name tf.gfile.Open is deprecated. Please use t
f.io.gfile.GFile instead.

W0413 16:15:36.780932  4596 deprecation_wrapper.py:119] From C:\Users\home\Deskt
op\TFOD\models\research\object_detection\utils\config_util.py:180: The name tf.g
file.Open is deprecated. Please use tf.io.gfile.GFile instead.

INFO:tensorflow:Writing pipeline config file to inference_graph\pipeline.config
I0413 16:15:36.785930  4596 config_util.py:182] Writing pipeline config file to
inference_graph\pipeline.config

(tfod) C:\Users\home\Desktop\TFOD\models\research>
```

The conversion is done successfully.

In research folder --- > inference_graph folder is created



TFOD ▸ models ▸ research ▸ inference_graph

Name	Date modified	Type	Size
saved_model	4/13/2020 4:15 PM	File folder	
checkpoint	4/13/2020 4:15 PM	File	1 KB
frozen_inference_graph.pb	4/13/2020 4:15 PM	PB File	51,091 KB
model.ckpt.data-00000-of-00001	4/13/2020 4:15 PM	DATA-00000-OF-0...	50,417 KB
model.ckpt.index	4/13/2020 4:15 PM	INDEX File	16 KB
model.ckpt.meta	4/13/2020 4:15 PM	META File	1,689 KB
pipeline	4/13/2020 4:15 PM	CONFIG File	3 KB

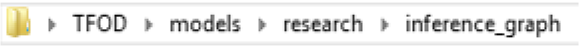
This is model we need for the prediction.

Testing the Model for prediction:

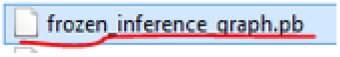
Testing the model in our local machine

Research is the root directory

Actually our .pb file for prediction is available in the inference_graph folder.



TFOD ▸ models ▸ research ▸ inference_graph

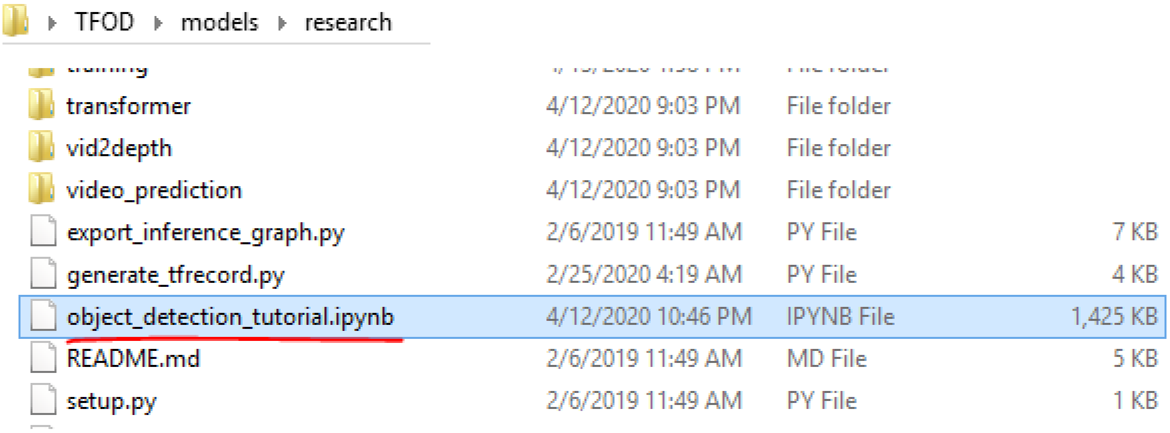


frozen_inference_graph.pb

We need to use this model for the prediction

How to do our testing in our local system.

Inside the object detection folder we have one .ipynb file, just copy the file and paste in to the research folder.

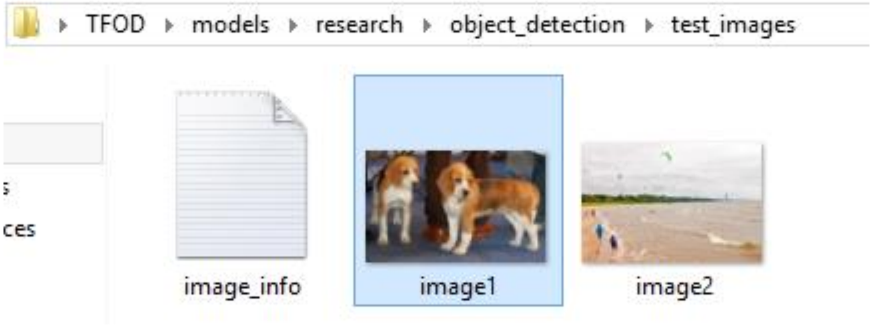


TFOD ▸ models ▸ research

training	[illegible]	File folder	
transformer	4/12/2020 9:03 PM	File folder	
vid2depth	4/12/2020 9:03 PM	File folder	
video_prediction	4/12/2020 9:03 PM	File folder	
export_inference_graph.py	2/6/2019 11:49 AM	PY File	7 KB
generate_tfrecord.py	2/25/2020 4:19 AM	PY File	4 KB
object_detection_tutorial.ipynb	4/12/2020 10:46 PM	IPYNB File	1,425 KB
README.md	2/6/2019 11:49 AM	MD File	5 KB
setup.py	2/6/2019 11:49 AM	PY File	1 KB

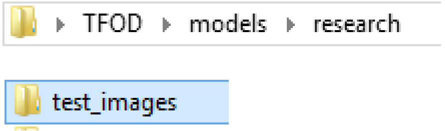
Now we need the folder called as test images for the prediction

That folder was available in below location



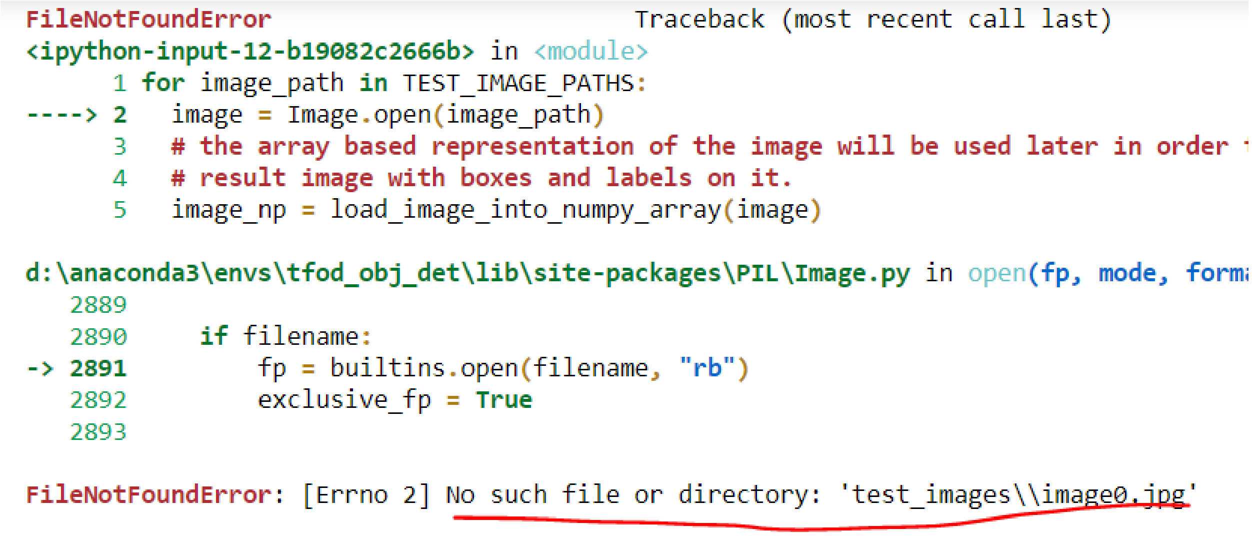
By default this images are provided by tensorflow community.

But we need to create a newfolder inside the research folder



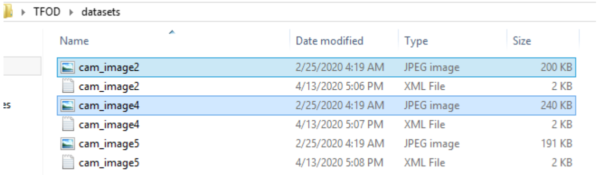
New folder was created with the name test_images inside the research folder. We need some trained pictures for prediction.

For that you have some images inside the dataset folder under TFOD copy some images there and paste in to test_images folder in the research folder.



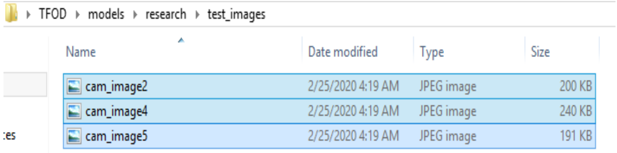
```
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-12-b19082c2666b> in <module>
      1 for image_path in TEST_IMAGE_PATHS:
----> 2   image = Image.open(image_path)
      3   # the array based representation of the image will be used later in order
      4   # result image with boxes and labels on it.
      5   image_np = load_image_into_numpy_array(image)

d:\anaconda3\envs\tfod_obj_det\lib\site-packages\PIL\Image.py in open(fp, mode, form
   2889
   2890     if filename:
-> 2891         fp = builtins.open(filename, "rb")
   2892         exclusive_fp = True
   2893

FileNotFoundError: [Errno 2] No such file or directory: 'test_images\\image0.jpg'
```

If we got the error then we need to rename all the files inside test_image folder



TFOD ▸ datasets

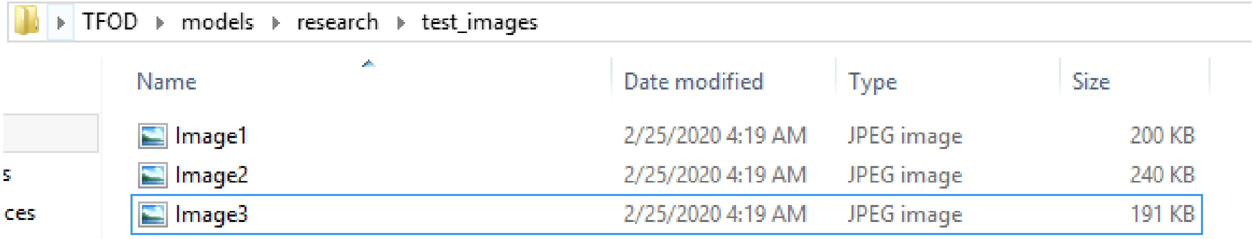
Name	Date modified	Type	Size
cam_image2	2/25/2020 4:19 AM	JPEG image	200 KB
cam_image2	4/13/2020 5:06 PM	XML File	2 KB
cam_image4	2/25/2020 4:19 AM	JPEG image	240 KB
cam_image4	4/13/2020 5:07 PM	XML File	2 KB
cam_image5	2/25/2020 4:19 AM	JPEG image	191 KB
cam_image5	4/13/2020 5:08 PM	XML File	2 KB



TFOD ▸ models ▸ research ▸ test_images

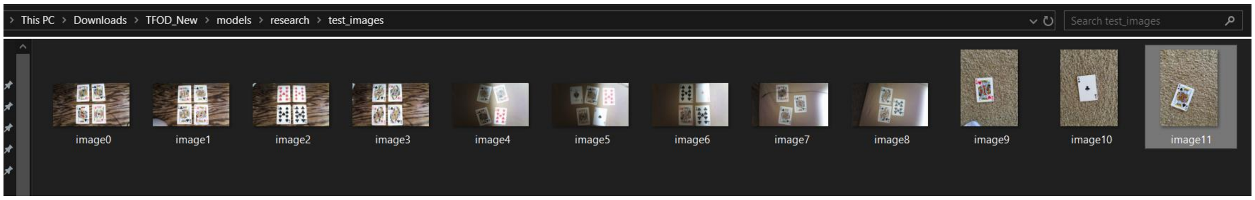
Name	Date modified	Type	Size
cam_image2	2/25/2020 4:19 AM	JPEG image	200 KB
cam_image4	2/25/2020 4:19 AM	JPEG image	240 KB
cam_image5	2/25/2020 4:19 AM	JPEG image	191 KB

Then I vl rename the file as image1, image2, image3



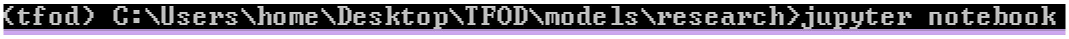
TFOD ▸ models ▸ research ▸ test_images

Name	Date modified	Type	Size
Image1	2/25/2020 4:19 AM	JPEG image	200 KB
Image2	2/25/2020 4:19 AM	JPEG image	240 KB
Image3	2/25/2020 4:19 AM	JPEG image	191 KB



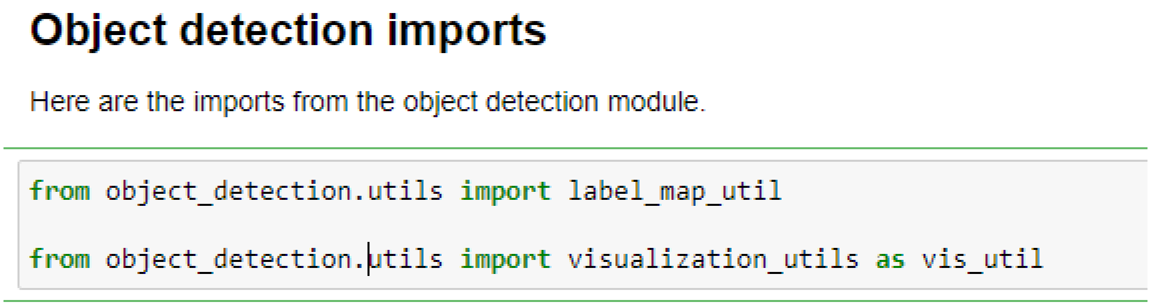
This PC › Downloads › TFOD_New › models › research › test_images

Now open the Jupyter notebook



```
(tfod) C:\Users\home\Desktop\TFOD\models\research>jupyter notebook
```

After open the object_detection_tutorial.ipynb file then we need to do some changes.



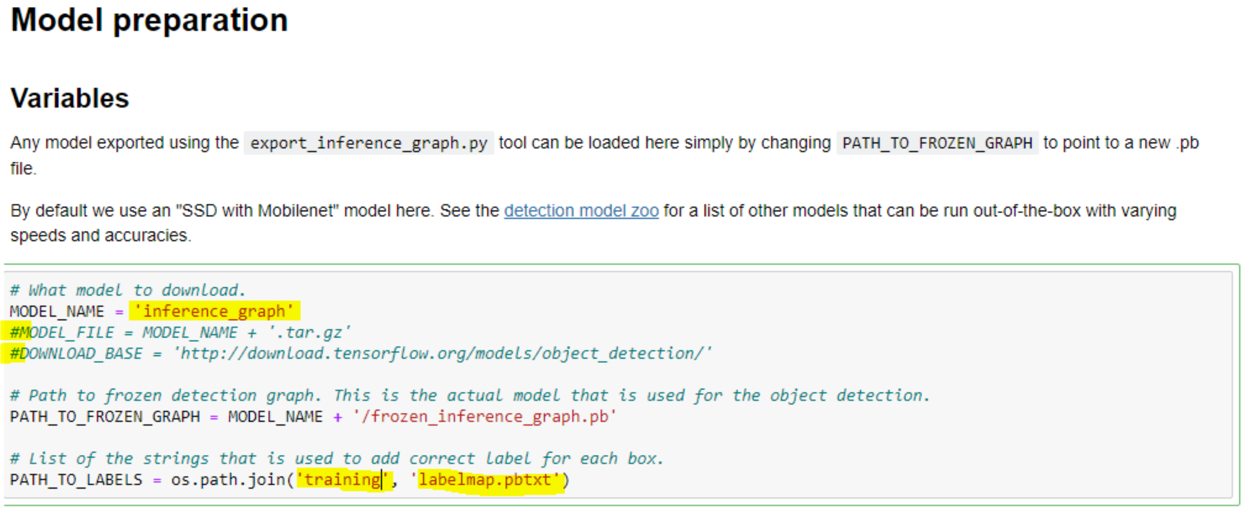
Object detection imports

Here are the imports from the object detection module.

```
from object_detection.utils import label_map_util

from object_detection.utils import visualization_utils as vis_util
```

Here instead of utils we need to add like object_detection.utils in both lines.

This are the changes I have done in the variables



Model preparation

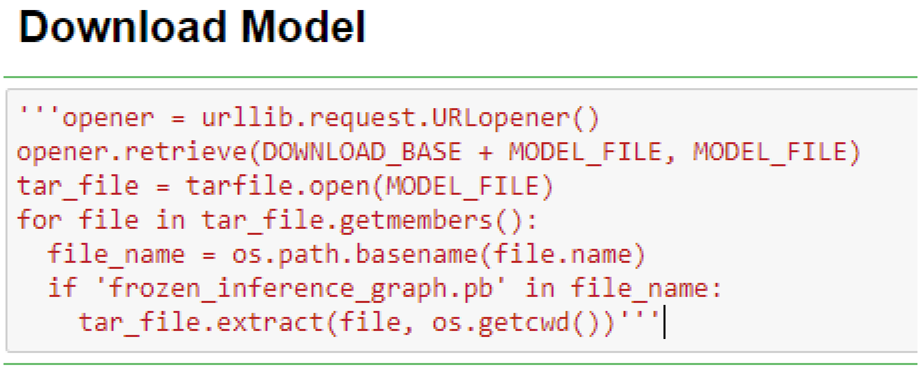
Variables

Any model exported using the `export_inference_graph.py` tool can be loaded here simply by changing `PATH_TO_FROZEN_GRAPH` to point to a new .pb file.

By default we use an "SSD with Mobilenet" model here. See the detection model zoo for a list of other models that can be run out-of-the-box with varying speeds and accuracies.

```
# What model to download.
MODEL_NAME = 'inference_graph'
#MODEL_FILE = MODEL_NAME + '.tar.gz'
#DOWNLOAD_BASE = 'http://download.tensorflow.org/models/object_detection/'

# Path to frozen detection graph. This is the actual model that is used for the object detection.
PATH_TO_FROZEN_GRAPH = MODEL_NAME + '/frozen_inference_graph.pb'

# List of the strings that is used to add correct label for each box.
PATH_TO_LABELS = os.path.join('training', 'labelmap.pbtxt')
```

For download we need so I just commented that code

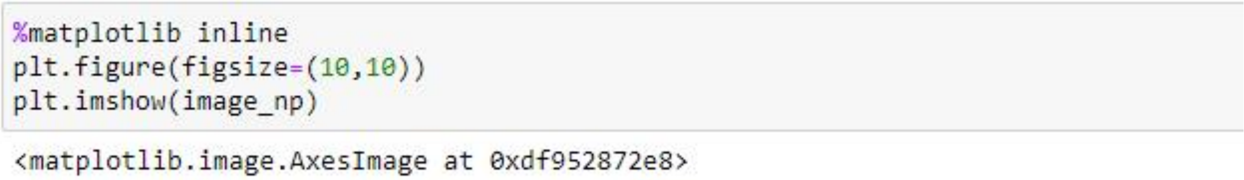


Download Model

```
'''opener = urllib.request.URLopener()
opener.retrieve(DOWNLOAD_BASE + MODEL_FILE, MODEL_FILE)
tar_file = tarfile.open(MODEL_FILE)
for file in tar_file.getmembers():
  file_name = os.path.basename(file.name)
  if 'frozen_inference_graph.pb' in file_name:
    tar_file.extract(file, os.getcwd())'''
```

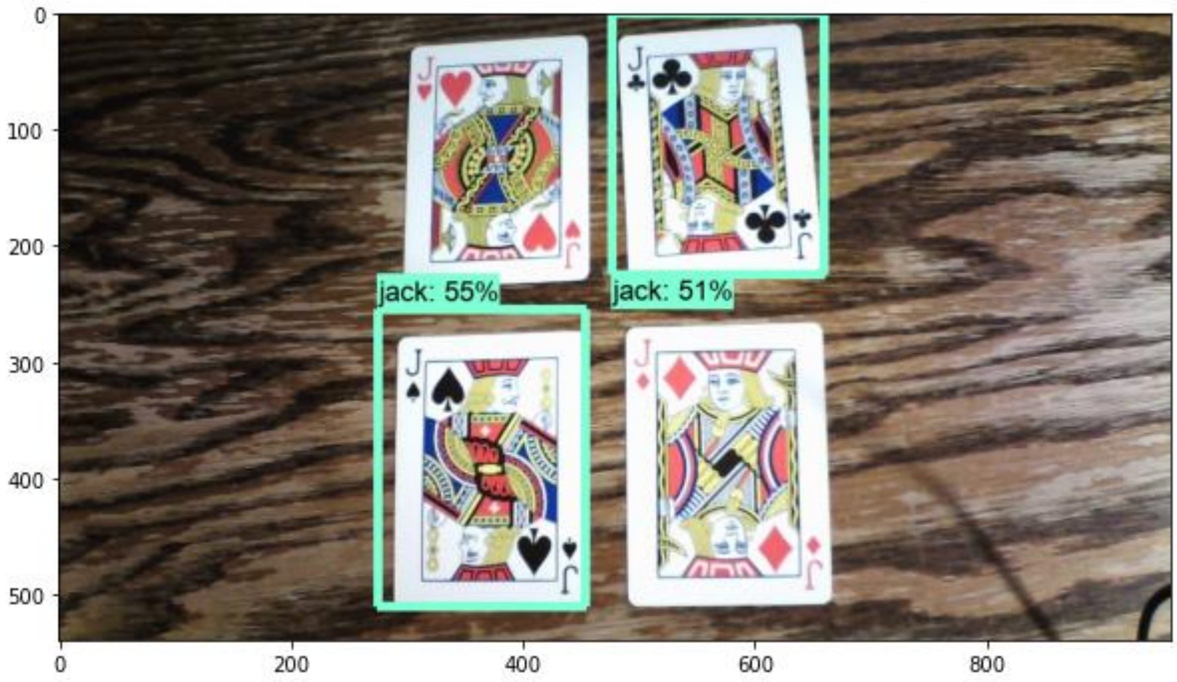
Run all the cells after done some necessary changes.

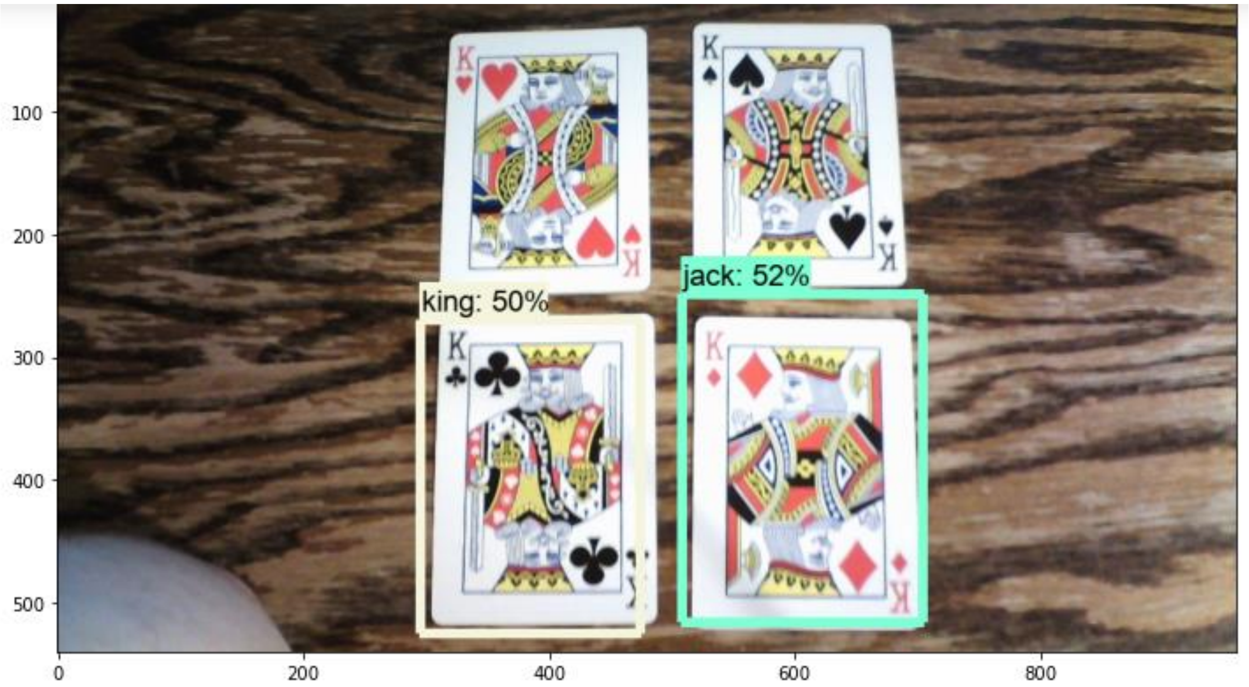
Finally you will get output like this.



```
%matplotlib inline
plt.figure(figsize=(10,10))
plt.imshow(image_np)
```

```
<matplotlib.image.AxesImage at 0xdf952872e8>
```





king: 50%
jack: 52%
100
200
300
400
500
0
200
400
600
800