# Faster-RCNN-COCO\_TF

The repo has been modified for training on [MS COCO](http://cocodataset.org/#home), in particular the 2014 dataset, as well as visualizing on a headless server. A pretrained model is also included below.

### Requirements: software

1. Requirements for Tensorflow (see: [Tensorflow](https://www.tensorflow.org/))

2. Python packages you might not have: `cython`, `python-opencv`, `easydict`

### Installation

1. Clone the Faster R-CNN repository

```bash

# Make sure to clone with --recursive

git clone --recursive https://github.com/dxyang/Faster-RCNN-COCO\_TF.git

```

2. Build pycocotools modules

```bash

cd $FRCN\_ROOT/lib

git clone https://github.com/cocodataset/cocoapi.git

cd cocoapi/PythonAPI

make

cd ../..

mv cocoapi/PythonAPI/pycocotools pycocotools

rm -rf cocoapi

```

3. Build the Cython modules

```bash

cd $FRCN\_ROOT/lib

make

```

### Training Model

1. Install gsutil if you haven't already

```bash

curl https://sdk.cloud.google.com | bash

```

2. Download the training, validation, test data for MS COCO

```bash

cd $FRCN\_ROOT/data

mkdir coco; cd coco

mkdir images; cd images

mkdir train2014

mkdir test2014

mkdir val2014

gsutil -m rsync gs://images.cocodataset.org/train2014 train2014

gsutil -m rsync gs://images.cocodataset.org/test2014 test2014

gsutil -m rsync gs://images.cocodataset.org/val2014 val2014

```

3. Download the annotations for MS COCO and unzip

```bash

cd $FRCN\_ROOT/data

gsutil -m rsync gs://images.cocodataset.org/annotations coco

cd coco

unzip annotations\_trainval2014.zip

unzip image\_info\_test2014.zip

rm \*.zip

```

4. Download the annotations for the 5000 image minival subset of COCO val2014 as mentioned [here](https://github.com/rbgirshick/py-faster-rcnn/tree/master/data)

```bash

cd $FRCN\_ROOT/data/coco/annotations

wget https://dl.dropboxusercontent.com/s/o43o90bna78omob/instances\_minival2014.json.zip

wget https://dl.dropboxusercontent.com/s/s3tw5zcg7395368/instances\_valminusminival2014.json.zip

unzip instances\_minival2014.json.zip; rm instances\_minival2014.json.zip

unzip instances\_valminusminival2014.json.zip; rm instances\_valminusminival2014.json.zip

```

5. Download the pre-trained ImageNet model [[Google Drive]](https://drive.google.com/open?id=0ByuDEGFYmWsbNVF5eExySUtMZmM) [[Dropbox]](https://www.dropbox.com/s/po2kzdhdgl4ix55/VGG\_imagenet.npy?dl=0)

```bash

cd $FRCN\_ROOT

wget https://www.dropbox.com/s/po2kzdhdgl4ix55/VGG\_imagenet.npy

mkdir data/pretrain\_model

mv VGG\_imagenet.npy data/pretrain\_model/VGG\_imagenet.npy

```

6. Create an output directory for log files

```bash

cd $FRCN\_ROOT

mkdir experiments/logs

```

7. Run script to train and test model

```bash

cd $FRCN\_ROOT

./experiments/scripts/faster\_rcnn\_end2end.sh $DEVICE $DEVICE\_ID VGG16 coco

```

- DEVICE is either cpu/gpu

### Testing Model

Run the following command.

```bash

python ./tools/test\_net.py \

--device gpu \

--device\_id 0 \

--weights output/faster\_rcnn\_end2end/coco\_2014\_train/VGGnet\_fast\_rcnn\_iter\_490000.ckpt \

--cfg experiments/cfgs/faster\_rcnn\_end2end.yml \

--imdb coco\_2014\_minival \

--network VGGnet\_test \

--vis False

```

- Changing ```vis``` to ```True``` will save images with all detections above 0.8 for every image in the testing set.

- The checkpoint files folder contains the following:

```bash

cd output/faster\_rcnn\_end2end/coco\_2014\_train

ls

# VGGnet\_fast\_rcnn\_iter\_490000.ckpt.data-00000-of-00001

# VGGnet\_fast\_rcnn\_iter\_490000.ckpt.index

# VGGnet\_fast\_rcnn\_iter\_490000.ckpt.meta

```

### Detections on your own images

Run the following command.

```bash

python ./tools/demo.py --model output/faster\_rcnn\_end2end/coco\_2014\_train/VGGnet\_fast\_rcnn\_iter\_490000.ckpt --img-path path\_to\_img\_folder

```

All your annotated images will be saved in a directory called detections\_test

The result of testing on coco\_2014\_minival

- Tensorflow model

[[Google Drive]] (https://drive.google.com/file/d/0Bw0qMqgwZcafZlRqRDYxSnBkNFE/view?usp=sharing)