Documentation: Implementation of Mask-RCNN using TensorFlow API

Prepared By: Lim Jun Jie

Environment Set-Up

- Refer to my github link for the instruction to implement TensorFlow GPU (CUDA) in Windows https://github.com/JJLim99/Implementation-of-TensorFlow-GPU-CUDA-in-Windows.git
- If you tend to train the model using your own graphic card, make sure you have enough GPU, otherwise you can use Google Colab (GPU runtime).

Mask-RCNN Project Set-up

1. Download the project repository from *https://github.com/JJLim99/SDD*.
2. Download the latest tensorflow models from *https://github.com/tensorflow/models* and replace it into the project repository.
3. In your current project repository, you must have following file:

i. CP (CheckPoint for your train model)

ii. dataset

- annotations
  - masks (annotated masks images in png)
  - xmls (xmls files for the annotated images)
- JPEGImages (train images in JPEG)
- TestImages (test images in JPEG)
- label.pbtxt

iii. IG (To save inference graph of your train model)

iv. models

v. pre_trained_models (To store pre-trained models, can try more models from *https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md*)

vi. Supporting_Script (Contains resize_image.py to resize and rename your dataset)

vii. mask_rcnn_inception_v2_coco (or any other .config file that u want to use, search more .config files in *PATHTOPROJECT\models\research\object_detection\samples\configs*)

viii. mask_rcnn_eval.ipnyb (To test your train model)

ix. create_mask_rcnn_tf_record.py (Copy this file into *PATHTOPROJECT\models\research\object_detection\dataset_tools*)

Implementation of Mask-RCNN

1. Preparing datasets
   i. Obtain your dataset. You can resize and rename your dataset using resize_images.py in Supporting_Script file.
   ii. Preparing annotated masks images using PixelAnnotationTool
   -Download PixelAnnotationTool from
   https://github.com/abreheret/PixelAnnotationTool
   -Save your annotated masks images in *PATHTOPROJECT/annotations/masks*
   iii. Preparing xml files for annotated images using labelImg
   -Download labelImg from *https://github.com/tzutalin/labelImg*
   -Save your xml files in *PATHTOPROJECT/annotations/xmls*
2. Run Check_pixel_values.ipynb to check the pixel of certain object.
3. Go to label.pbtxt and create_mask_rcnn_tf_record.py, label the object with its corresponding pixel value. In label.pbtxt, the last 3 digits refer to the pixel value of the object.
4. Generating tf record
   i. Before generating tf record, in cmd, go to *PATHTOPROJECT/models/research* and run the following command to prevent error.

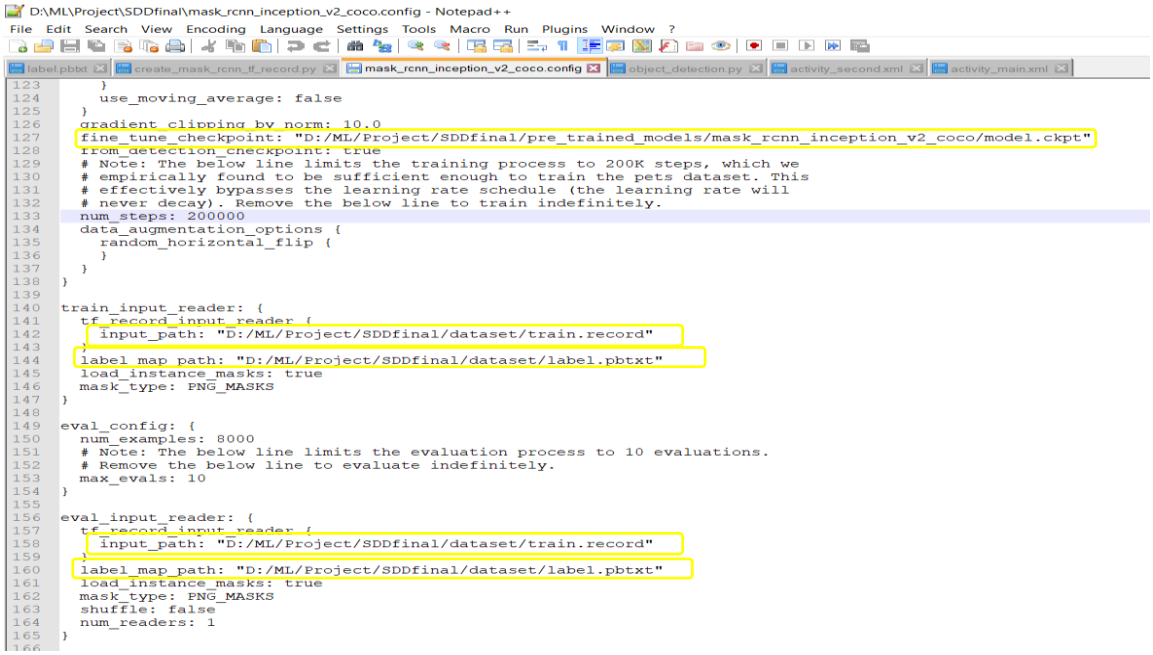
```
set
PYTHONPATH=PATHTOPROJECT\models;PATHTOPROJECT\models\research
;PATHTOPROJECT\models\research\slim
set PATH=%PATH%;%PYTHONPATH%
protoc object_detection/protos/*.proto --python_out=.
python setup.py build
python setup.py install
```

   ii. Then, at the *PATHTOPROJECT\models\research*, run the following command to generate tf record.

```
python object_detection/dataset_tools/create_mask_rcnn_tf_record.py --
data_dir=PATHTOPROJECT/dataset --annotations_dir=Annotations --
image_dir=JPEGImages --output_dir=PATHTOPROJECT/dataset/train.record --
label_map_path=PATHTOPROJECT/dataset/label.pbtxt
```

5. Training

i. Open the .config file in your project repository and modify 5 highlighted parts as shown in the image below according to your path. (I used mask_rcnn_inception_v2_coco.config)



ii. Download the Mask-RCNN pre-trained models from https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md (I used mask_rcnn_inception_v2_coco)

iii. Run the following command to start training.

python object_detection/legacy/train.py --train_dir=PATHTOPROJECT/CP --pipeline_config_path=PATHTOPROJECT/mask_rcnn_inception_v2_coco.config

Train the models until the loss is below 0.2 or lower.

6. Generating Inference_Graph

   Once the training had been done, run the following command to generate inference_graph.

```
python object_detection/export_inference_graph.py --input_type=image_tensor --
pipeline_config_path=PATHTOPROJECT/mask_rcnn_inception_v2_coco.config --
trained_checkpoint_prefix=PATHTOPROJECT/CP/model.ckpt-20000 --
output_directory=PATHTOPROJECT/IG
```

   # The highlighted test in the command above should be changed according to your steps of training

7. Test the trained models.

   Open jupyter notebook and run mask_rcnn_eval.ipnyb to test the trained models. Change the PATH as commented in the script.