

# **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](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.

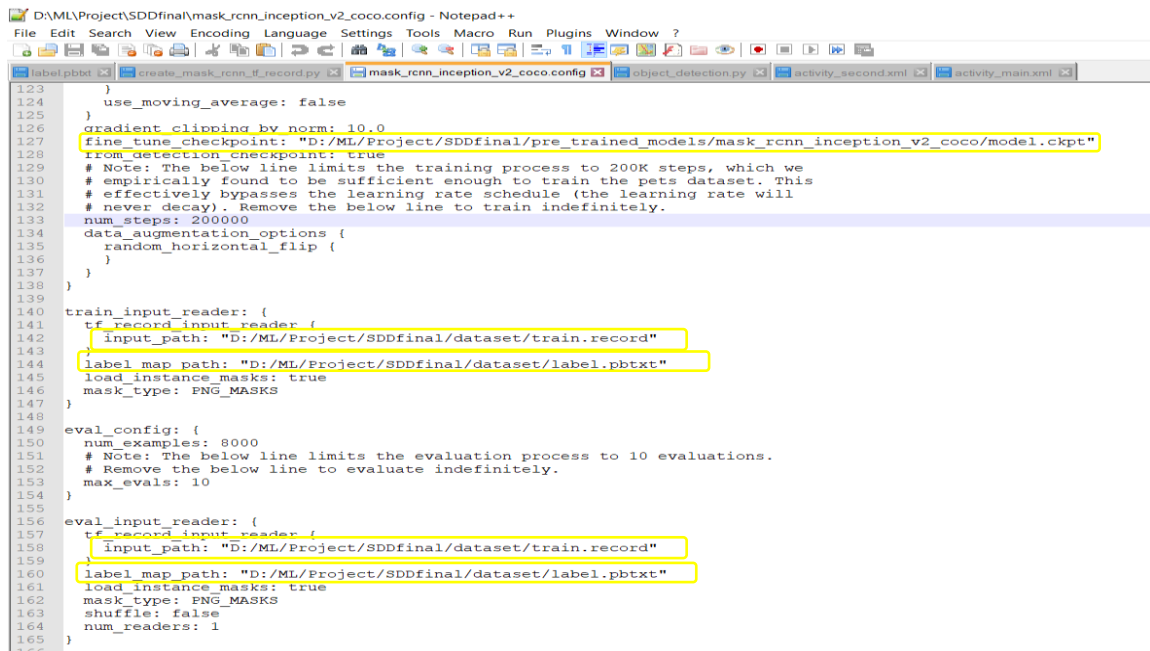
```
set
PYTHONPATH=PATHTOPROJECT\models;PATHTOPROJECT\models\research
;PATHTOPROJECT\models\research\slim
set PATH=%PATH%;%PYTHONPATH%
protoc object_detection/protos/*.proto --python_out=.
python setup.py build
python setup.py install
```

- 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)



```
123 }
124 use_moving_average: false
125 }
126 gradient_clipping_by_norm: 10.0
127 fine_tune_checkpoint: "D:/ML/Project/SDDfinal/pre_trained_models/mask_rcnn_inception_v2_coco/model.ckpt"
128 from_detection_checkpoint: true
129 # Note: The below line limits the training process to 200K steps, which we
130 # empirically found to be sufficient enough to train the pets dataset. This
131 # effectively bypasses the learning rate schedule (the learning rate will
132 # never decay). Remove the below line to train indefinitely.
133 num_steps: 200000
134 data_augmentation_options {
135   random_horizontal_flip {
136   }
137 }
138 }
139
140 train_input_reader: {
141   tf_record_input_reader {
142     input_path: "D:/ML/Project/SDDfinal/dataset/train.record"
143   }
144   label_map path: "D:/ML/Project/SDDfinal/dataset/label.pbtxt"
145   load_instance_masks: true
146   mask_type: PNG_MASKS
147 }
148
149 eval_config: {
150   num_examples: 8000
151   # Note: The below line limits the evaluation process to 10 evaluations.
152   # Remove the below line to evaluate indefinitely.
153   max_evals: 10
154 }
155
156 eval_input_reader: {
157   tf_record_input_reader {
158     input_path: "D:/ML/Project/SDDfinal/dataset/train.record"
159   }
160   label_map path: "D:/ML/Project/SDDfinal/dataset/label.pbtxt"
161   load_instance_masks: true
162   mask_type: PNG_MASKS
163   shuffle: false
164   num_readers: 1
165 }
166 }
```

- ii. Download the Mask-RCNN pre-trained models from [https://github.com/tensorflow/models/blob/master/research/object\\_detection/g3doc/detection\\_model\\_zoo.md](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.