

ReTechLabs Assignment Report:

I was asked to use SOTA for object detection task for product item in the store shelf. I used YOLO V5 using pytorch and SSD MobileNet V2 FPN lite using TensorFlow api for the task.

YOLO v5:

Environment setup:

For setting up environment for the YOLO V5 algorithm I used requirements.txt file provided by Official YOLO V5 tutorial, I only had to additionally setup weights and Biases account.

Dataset Preprocessing:

For YOLO V5 task the dataset was needed to be in the COCO format, Initially to deal with the bounding box errors I used Roboflow service to automatically detect error bounding boxes and remove them from the data once This was performed, I converted the dataset into COCO format, I also scaled the images to 160*160 size since I was unable to train using larger images on my laptop. I used Roboflow service to scale the images as well.

Training:

I trained the data for 100 epochs and obtained around 0.76 map@0.50 for validation data. I also was able to generate F1 curve, Confusion Matrix, Precision Curve, Recall Curve and Precision Recall curve using the script which was provided by official tutorial. I have attached both train and validation metrics to the repository along with the weights and config files.

Test:

I also performed testing on the given Test_Set data and I was able to see most of the objects being detected from the images, However I was unable to precisely measure the effectiveness of the model since I did not have annotations for the test data.

I have also attached Commands which I used to run in the scripts folder along with codes used to convert Pascal VOC to COCO format.

SSD MobileNet V2 fpnlite 320*320:

Environment Setup:

For this model I had to setup environment on my own. I initially setup Tensorflow gpu and installed other libraries, for this task I specifically referred to the following tutorial [\(1565\) Intro - TensorFlow Object Detection API Tutorial p.1 - YouTube](#). I only installed tensorboard to visualize the results.

Data Preprocessing:

Initially I used same tool used in yolo to deal with bounding box errors and scale images to 320*320 to make it easier to run on my local machine. Then I converted the Pascal VOC to TFrecords using the Script mentioned in the scripts folder.

Training:

The training for this model was different from the previous model, I ran the model for 3 hours by monitoring the loss to avoid overfitting. This also generated log files which I have attached along with the checkpoint and saved model file.

Testing: I ran inference scripts to generate log files for validation data, I visualized the validation data using tensor board and attached results. I also generated inference graph which I later used to generate predictions for test data. MAP @ 0.50 for validation data was 0.70 which was slightly lesser than the yolo model.

The outputs for the test data have also been attached.

I also attached commands which I used to run scripts in the scripts folder along with scripts used to test on test images and the format conversion scripts.

Overall, I think from the following metrics that yolo has performed better, However I am yet to understand my results for the test data.

References:

[How to Train YOLO v5 on a Custom Dataset | Paperspace Blog](#)

[Train Object Detection model TF2.ipynb - Colaboratory \(google.com\)](#)

[\(1568\) Intro - TensorFlow Object Detection API Tutorial p.1 - YouTube](#)

[roboflow.com](#)

[ultralytics/yolov5: YOLOv5 🚀 in PyTorch > ONNX > CoreML > TFLite \(github.com\)](#)

