

Project Writeup Template

1. Project overview:

In this project we used Waymo dataset and download the data as a **tfrecord** extention.

In addition, we utilized a tensorflow object detection api to localize the objects (vehicles , bedestrains and cyclist) where the object detection is necessary to perception of the enlivements during the driving besides of other sensors as lidar and radar.

2. Set up:

Firstly, we download TensorFlow object detection API and download the Waymo dataset.

We used Utilize.py to read the tfrecord files and put the boxes in the objects of the photos.

In addition, we analyze the data by using bar diagrams.

3. Dataset:

Exploring the data

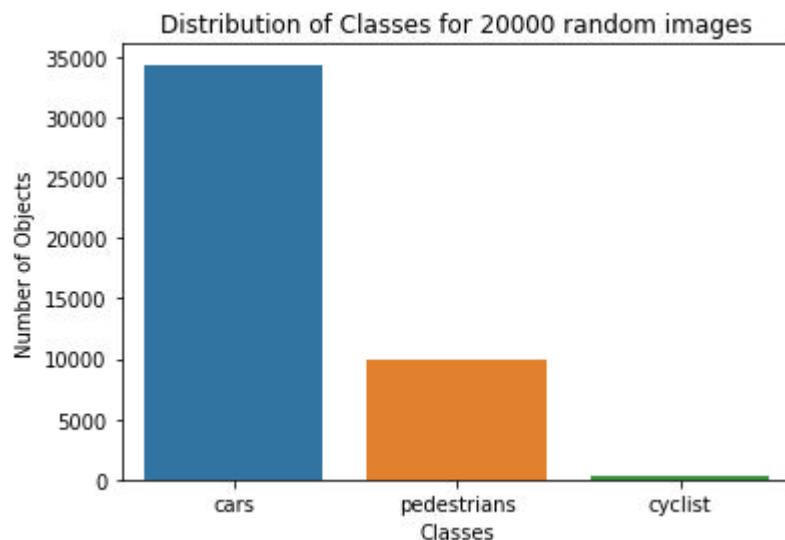
Using utils.py to open the tfrecord data with adding the bounding boxes, we find the sample of data as the following:





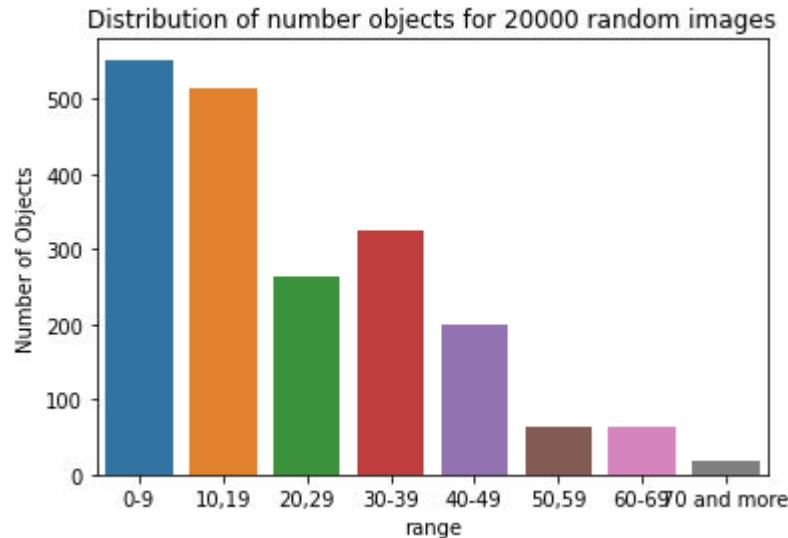
Analyze the data:

According to the diagram below:



We used 20000 images and we find out that the most of objects are cars whereas pedestrians and cyclists are fewer which means we need more data about pedestrians and cyclists to increase their accuracy.

In other sides, the diagram below:



We count the number of objects in each photo where we used 2000 images.

We see that most images have less than 10 objects whereas the images with more than 70 objects are less than 100 images, we can deduce that the images with congestion flow are a little.

Where we need more images with congestion flow if we need more accuracy for congestion problems.

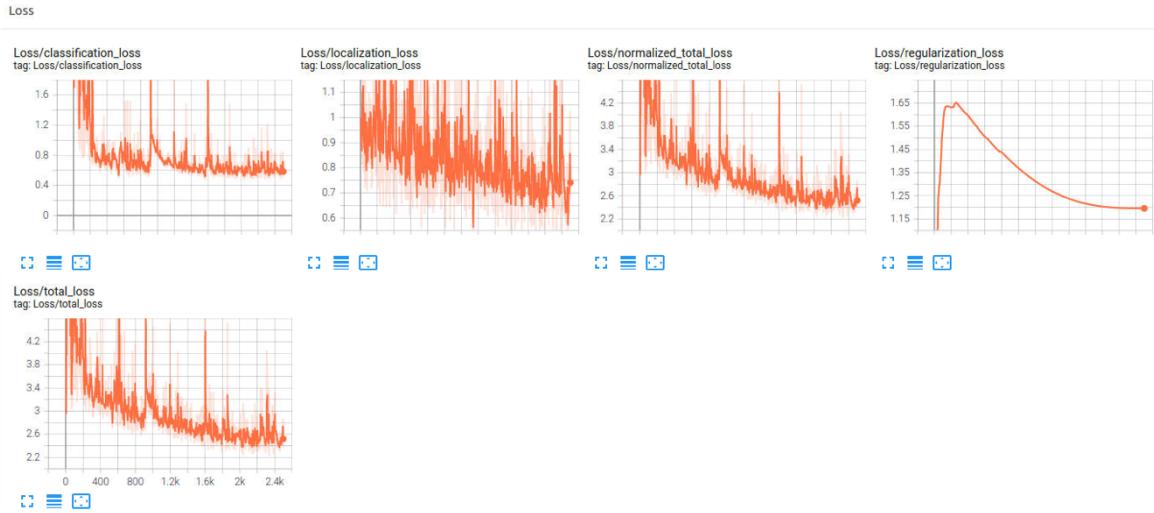
Training:

In the training we used ssd training where we detect the algorithm , matrices and threshold in the pipeline_new.config file.

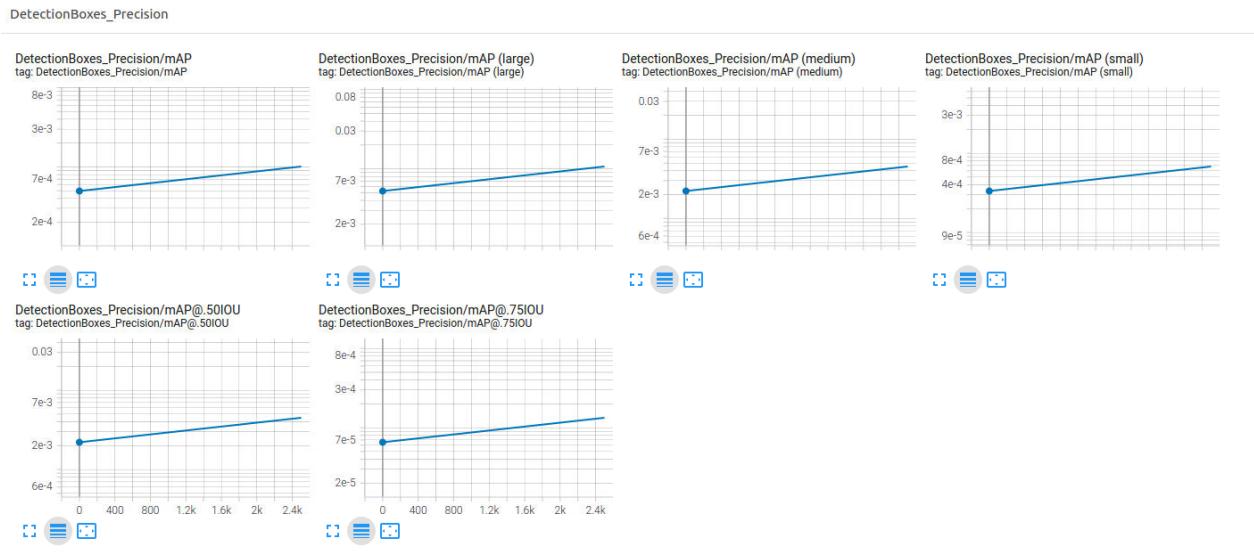
We that iou to calculate the loss function.

We could see in the photo at the below which it describes the loss function that the loss is decreasing and we got more accuracy.

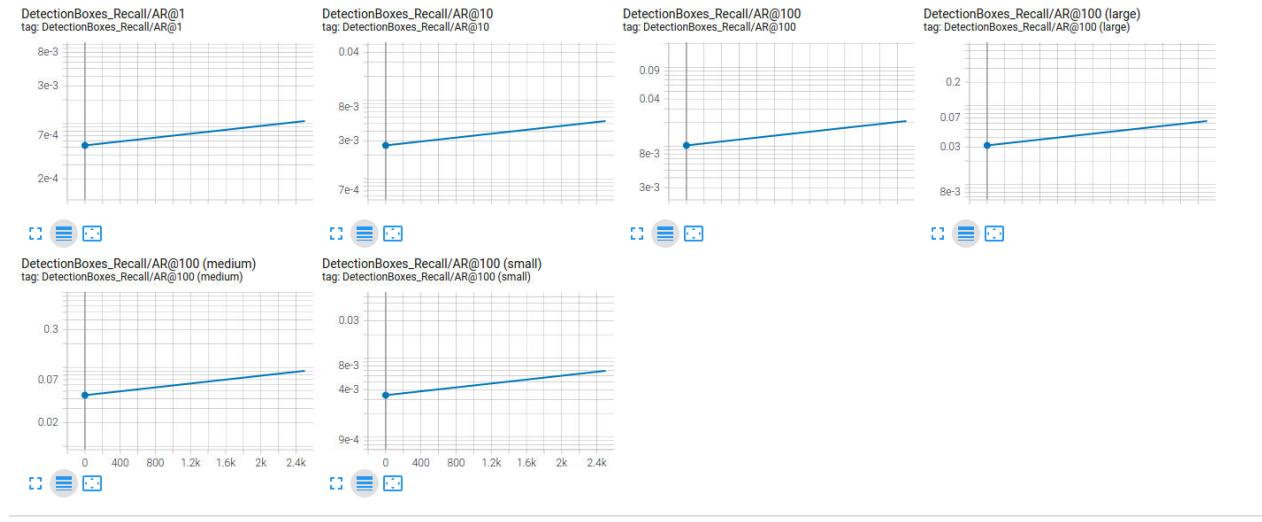
We could test other algorithms to get more accuracy as yolo and use yolo loss function instead of IOU.



For the mAP we could find the result as the following:



And recall :



Config file:

In the project there is a pipeline.config file where it use SSD Resnet algorithem.

We need to editethe config file which we need to change its location and change the batch size. Therefore, we run the following code:

```
python edit_config.py --train_dir /home/workspace/data/train/ --eval_dir /home/workspace/data/val/ --batch_size 2 --checkpoint /home/workspace/experiments/pretrained_model/ssd_resnet50_v1_fpn_640x640_coco17_tpu-8/checkpoint/ckpt-0 --label_map /home/workspace/experiments/label_map.pbtxt
```

where we got a new config file (pipeline_new.config) and we moved it to the refrence file.

Augmentation:

We used many augmentation for the images as:

- 1- Random crop image
- 2- Random rgb to gray
- 3- Random adjust contrast
- 4- Random adjust brightness



Improve the data:

Some image we see that there is some water drops in the image or raining.

We could solve the severe weather problems by many algorithms, one of them so called

Transformer-based Restoration of Images Degraded by Adverse Weather Conditions

Where we could find a paper speak about it at this link

<https://arxiv.org/abs/2111.14813>