MEERUT INSTITUTE OF ENGINEERING AND TECHNOLOGY

(SESSION 2021-2022)

MINI PROJECT REPORT

ON

"OBJECT DETECTION"

BACHELOR OF TECHNOLOGY IN

COMPUTER SCIENCE AND ENGINEERING

Submitted to:

MR. ABHAY JAIN

Submitted by:

Nidhi Singhal - RN:2000681530033

Shreya Singh - RN: 2000681530047

Prachi Gupta - RN:2000681530037

3rd SEMESTER

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MEERUT INSTITUTE OF ENGINEERING AND TECHNOLOGY, MEERUT

DECLARATION

We hereby declare that the project entitled - "**Object Detection**", which is being submitted as Mini Project in department of Computer Science and engineering to Meerut Institute of Engineering and Technology, Meerut (U.P.) is an authentic record of our genuine work done under the guidance of Assistant Prof. " **Mr. Abhay Jain**" of Computer Science and Engineering, Meerut Institute of Engineering and Technology, Meerut.

DATE: 17 JANUARY 2022

PLACE: MEERUT

NAME: Nidhi Singhal (2000681530033**)**

Prachi Gupta (2000681530037**)**

Shreya Singh (2000681530047**)**

CERTIFICATE

This is to certify that mini project report entitled – "**Object Detection**" submitted by "**Prachi Gupta, Shreya Singh, Nidhi Singhal**" has been carried out under the guidance of "**Mr. Abhay Jain**" of Computer Science and Engineering, Meerut Institute of Engineering and Technology, Meerut. This project report is approved for Mini Project (KCN 354) in 3rd semester in computer science and engineering department from Meerut Institute of Engineering and Technology, Meerut.

Supervisor: Mr. Abhay Jain

Date: 17 January 2022

ACKNOWLEDGEMENT

We express our sincere indebtedness towards our guide **Mr Abhay Jain** our professor of Computer Science and Engineering, Meerut Institute of Engineering and Technology, Meerut for his valuable suggestion, guidance and supervision throughout the work. Without his kind patronage and guidance the project would not have taken shape.We would also like to extend our gratitude and sincere regards for his kind approval of the project. Time to time counselling and advises.

We would also like to thank our HOD Dr. SWATI SHARMA, Department of Computer Science and engineering, Meerut Institute of Engineering and Technology, Meerut for her expert advice and counselling from time to time. We owe sincere thanks to all the faculty members in the department of Computer Science and engineering for their kind guidance and encouragement .

TABLE OF CONTENTS

.

CHAPTER 1

INTRODUCTION

Object detection :

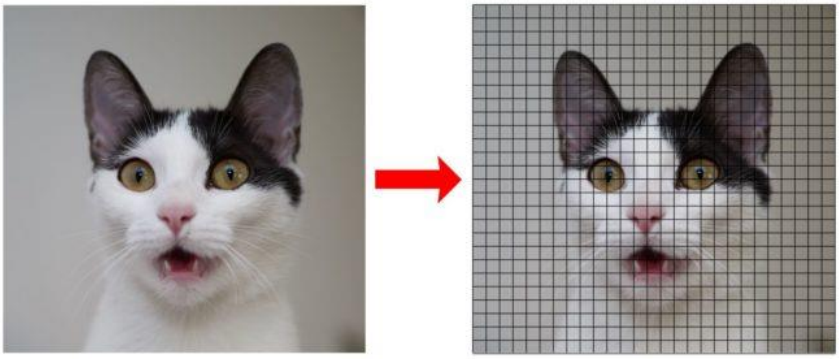
Object detection is a computer vision technique in which a software system can detect, locate, and trace the object from a given image or video. It identifies the class of object (person, table, chair, etc.) and their location-specific coordinates in the image. The location is pointed out by drawing a bounding box around the object. The bounding box may or may not accurately locate the position of the object. The ability to locate the object inside an image defines the performance of the algorithm used for detection .

Here in our project we used pre-trained models for training our system because creating from scratch would require an in-depth knowledge of CNN and DEEP neural network.

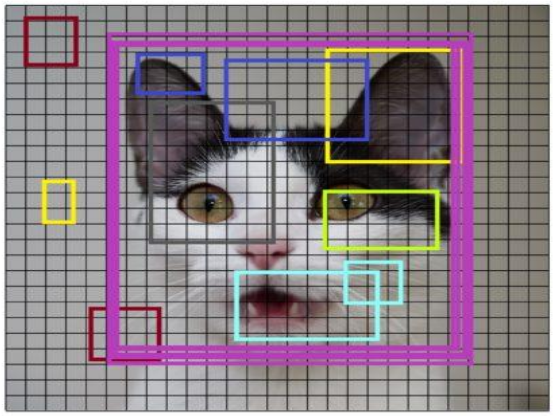
BACKGROUND

Generally, the object detection task is carried out in three steps:

- Generates the small segments in the input as shown in the image below. As you can see the large set of bounding boxes are spanning the full image.



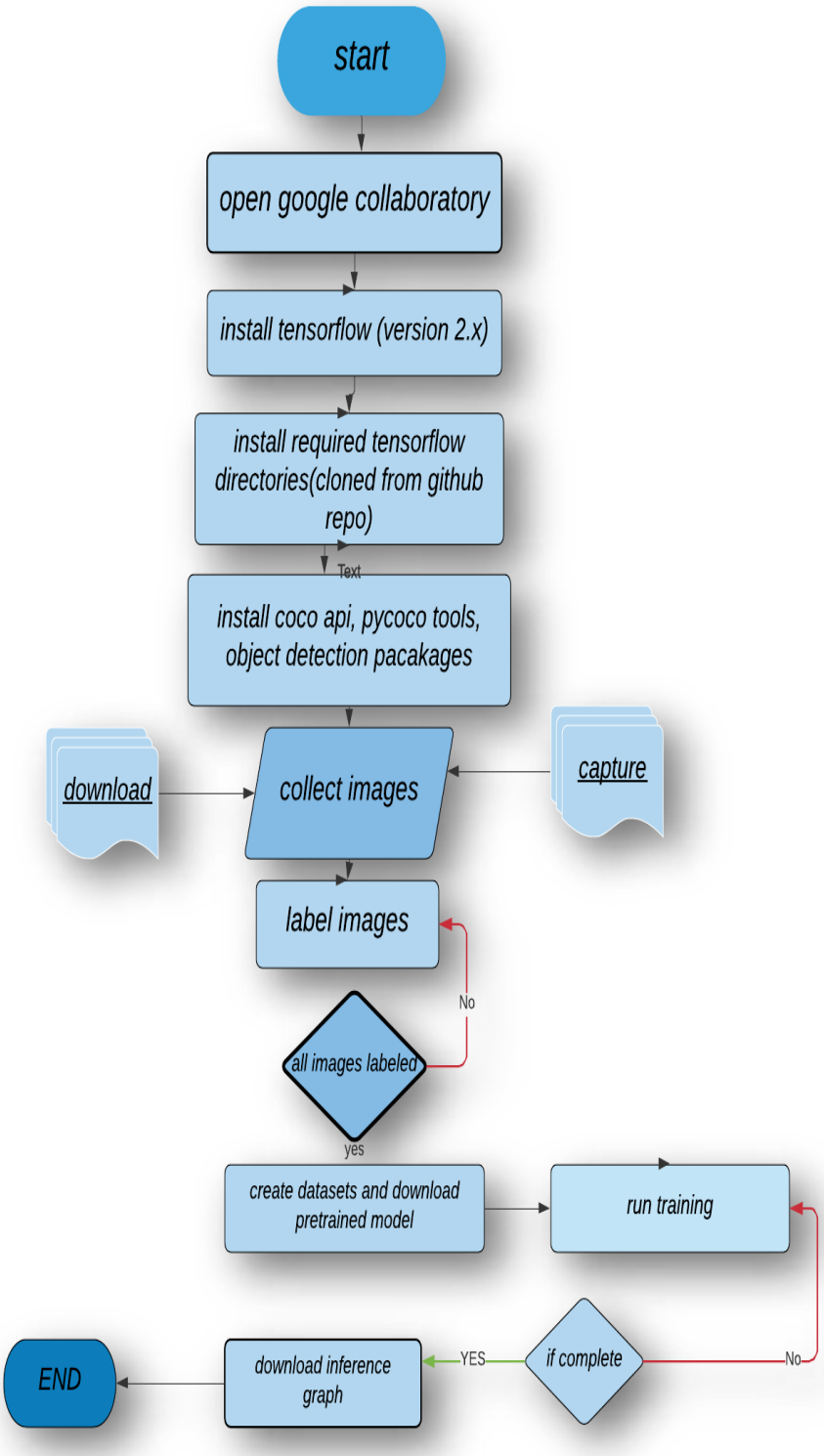
- Feature extraction is carried out for each segmented rectangular area to predict whether the rectangle contains a valid object.



- Overlapping boxes are combined into a single bounding rectangle (Non-Maximum Suppression)

Chapter – 2 SYSTEM DESIGN

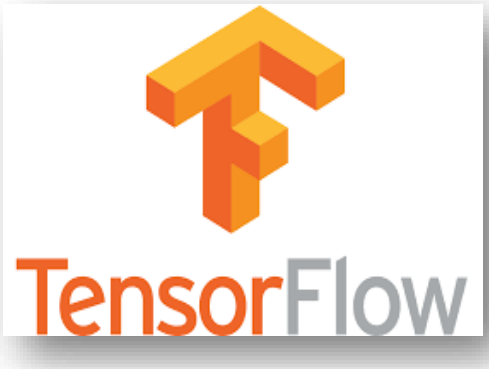
FLOWCHART



Tech stack

TENSORFLOW: **TENSORFLOW** is an open-source library for numerical computation and large-scale machine learning that ease Google Brain Tensor Flow, the process of acquiring data, training models, serving predictions, and re-fining future results.

- Tensorflow bundles through **MACHINE LEARNING** and **DEEP LEARNING** models and algorithms.

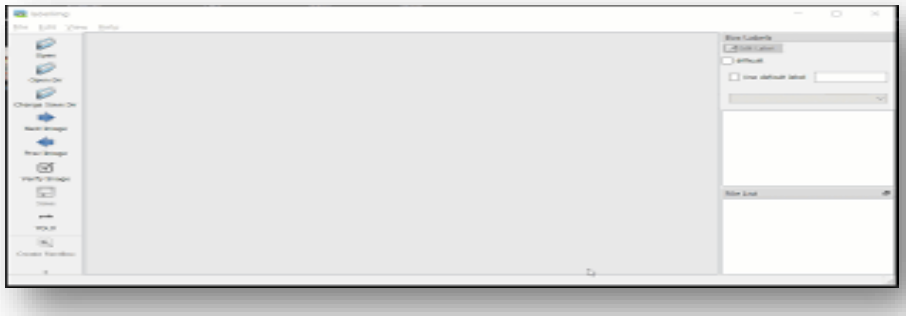


GOOGLE COLABORATORY: Colaboratory, or "Colab" for short, is **a product from Google Research**. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. It also provides us GPU or TPU support which is needed in object-detection .

GITHUB : GitHub is **a code hosting platform for version control and collaboration**. It lets you and others work together on projects from anywhere . Provider of Internet hosting for software development and version control using Git. It offers the distributed version control .

LabelImg: LabelImg is **a free, open source tool for graphically labeling images**. It's written in Python and uses QT for its graphical interface. It's an easy, free way to label a few hundred images to try out your next object detection project.

INTERFACE:



PROGRAMMING TOOLS

ANACONDA : **Anaconda** is a distribution of the PYTHON and R Programming Languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analysis etc.), that aims to simplify package management and deployment.

Provides anaconda prompt used in our project.

ATOM : **Atom** is a free and open source text and source code editor for macOS and LINUX, and Microsoft Windows with support for plug-ins written in Javascript and embedded Git Control .

OTHER TECHNOLOGIES USED:

1. Browser – Internet explorer,chrome
2. Nvidia GPU (GTX 650 or newer)
3. CUDA Toolkit v11.2
4. Flowchart – Visme Flowchart maker
5. CuDNN 8.1.0

Chapter – 3

PROCEDURE / METHAODOLOGY

What is Tensorflow object detection API?

The TensorFlow Object Detection API is an open-source framework built on top of TensorFlow that makes it easy to construct, train and deploy object detection models.

- There are already pre-trained models in their framework which are referred to as Model Zoo.
- It includes a collection of pre-trained models trained on various datasets such as the
    - COCO (Common Objects in Context) dataset,

PROCEDURE

1. INSTALL ANACONDA PYTHON 3.8 / IF WORKING ON COLLABORATORY CREATE A NEW NOTEBOOK.

Setting up a virtual environment

2. INSTALL TENSORFLOW VERSION 2.X OR ABOVE
   If you have gpu in your system install Tensorflow-gpu.

3. CLONE OR DOWNLOAD TESORFLOW GITHUB REPOSITORY.
   https://github.com/tensorflow/models.git

NOTE : DOWNLOADING TENSORFLOW API CAN BE DONE MANUALLY OR BY GITBHUB.

4. NOW CLONE AND DOWNLOAD COCO API,
   https://github.com/cocodataset/cocoapi.git

COCO API: COCO is **a large image dataset designed for object detection, segmentation, person keypoints detection, stuff segmentation, and caption generation**. This package provides Matlab, Python, and Lua APIs that assists in loading, parsing, and visualizing the annotations in COCO.

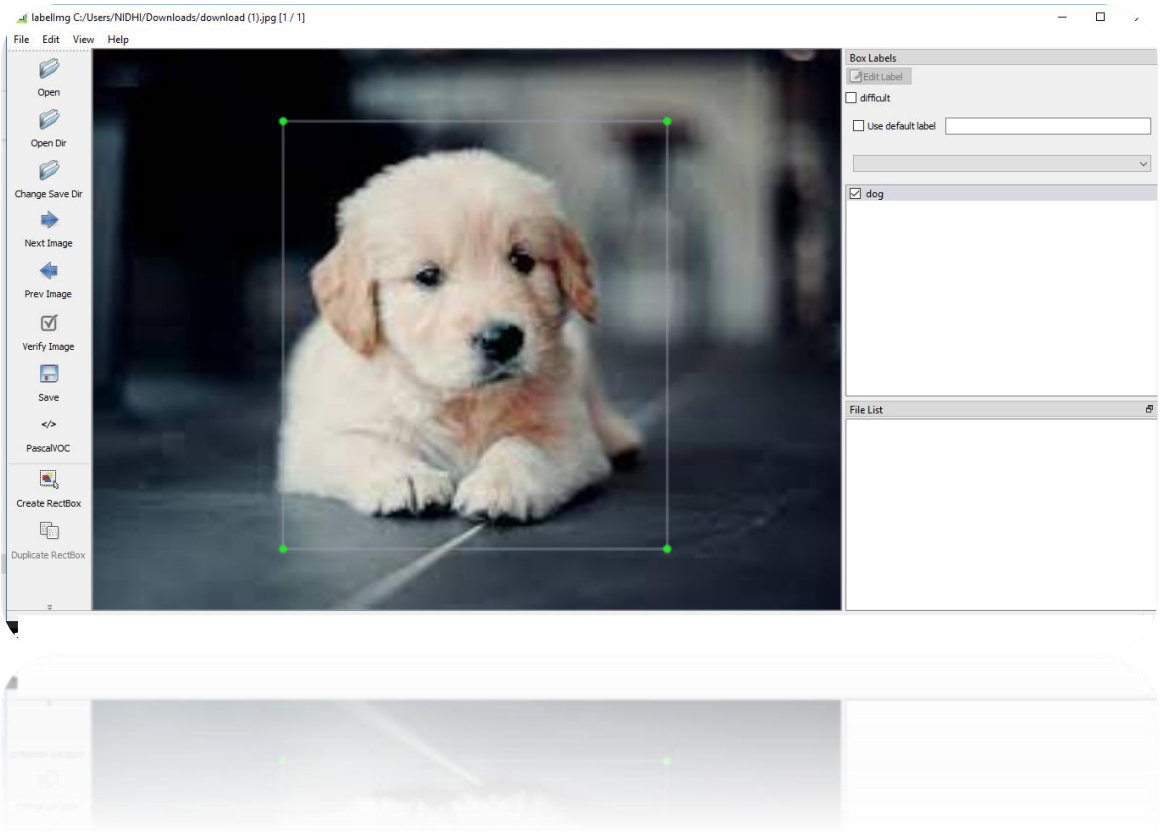
5. pip install pillow Cython lxml jupyter matplotlib contextlib2 tf_slim

6. Download object detection packages in researchfolder of model.

7. create work environment now . Here image data set is created and pre-trained model is downloaded.

Image dataset must have ration of 9:1 where 9 is for train and 1 for test.

Along with ther labeled xml files created using labelImg.



Pre-trained model is downloaded from models zoo here there is trade off between speed and accuracy if for real time we use speed over accuracy hence **faster-rcnn** or any else can be used we used **SSD Resnet** because of average accuracy.

Here **mAP (mean average precision)** is the product of precision and recall on detecting bounding boxes. It's a good combined measure for how sensitive the network is to objects of interest and how well it avoids false alarms. The higher the mAP score, the more accurate the network is but that comes at the cost of execution speed which we want to avoid here.

Model zoo:

Model name	Speed (ms)	COCO mAP[^1]	Outputs
ssd_mobilenet_v1_coco	30	21	Boxes
ssd_mobilenet_v1_0.75_depth_coco ☆	26	18	Boxes
ssd_mobilenet_v1_quantized_coco ☆	29	18	Boxes
ssd_mobilenet_v1_0.75_depth_quantized_coco ☆	29	16	Boxes
ssd_mobilenet_v1_ppn_coco ☆	26	20	Boxes
ssd_mobilenet_v1_fpn_coco ☆	56	32	Boxes
ssd_resnet_50_fpn_coco ☆	76	35	Boxes
ssd_mobilenet_v2_coco	31	22	Boxes
ssd_mobilenet_v2_quantized_coco	29	22	Boxes
ssdlite_mobilenet_v2_coco	27	22	Boxes
ssd_inception_v2_coco	42	24	Boxes
faster_rcnn_inception_v2_coco	58	28	Boxes
faster_rcnn_resnet50_coco	89	30	Boxes

8. Now tfreord file is created.

9. Configuration file is edited then changed as per our needs.

10. Then trainining process isstarted depending upon gpu presence or absence it may take time if done on cpu may take more than an our.

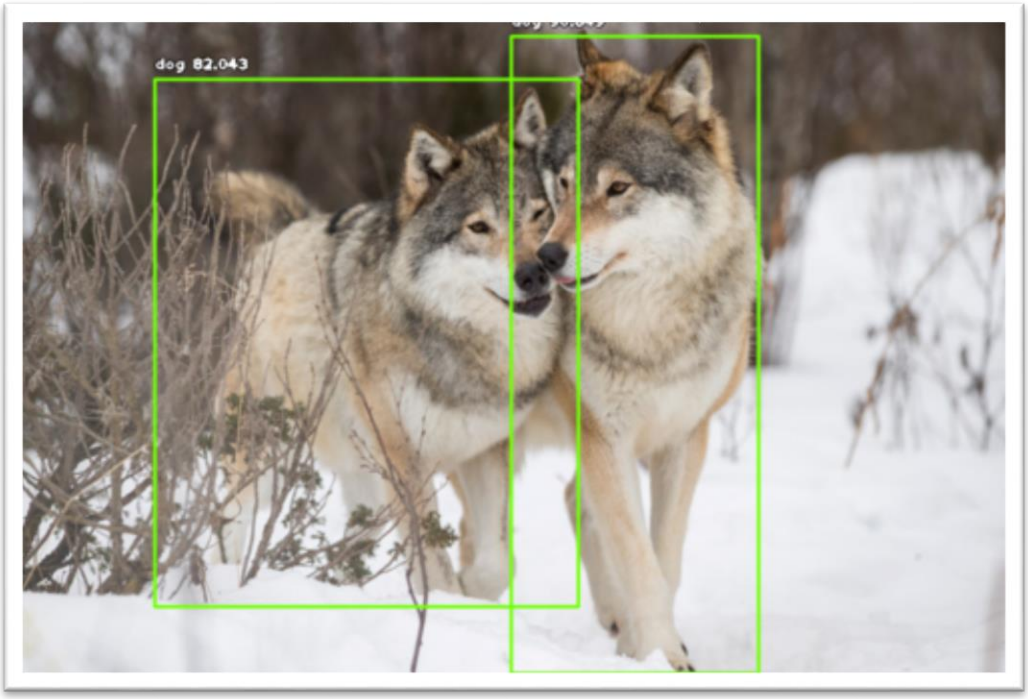
11. After training model is tested by images present.

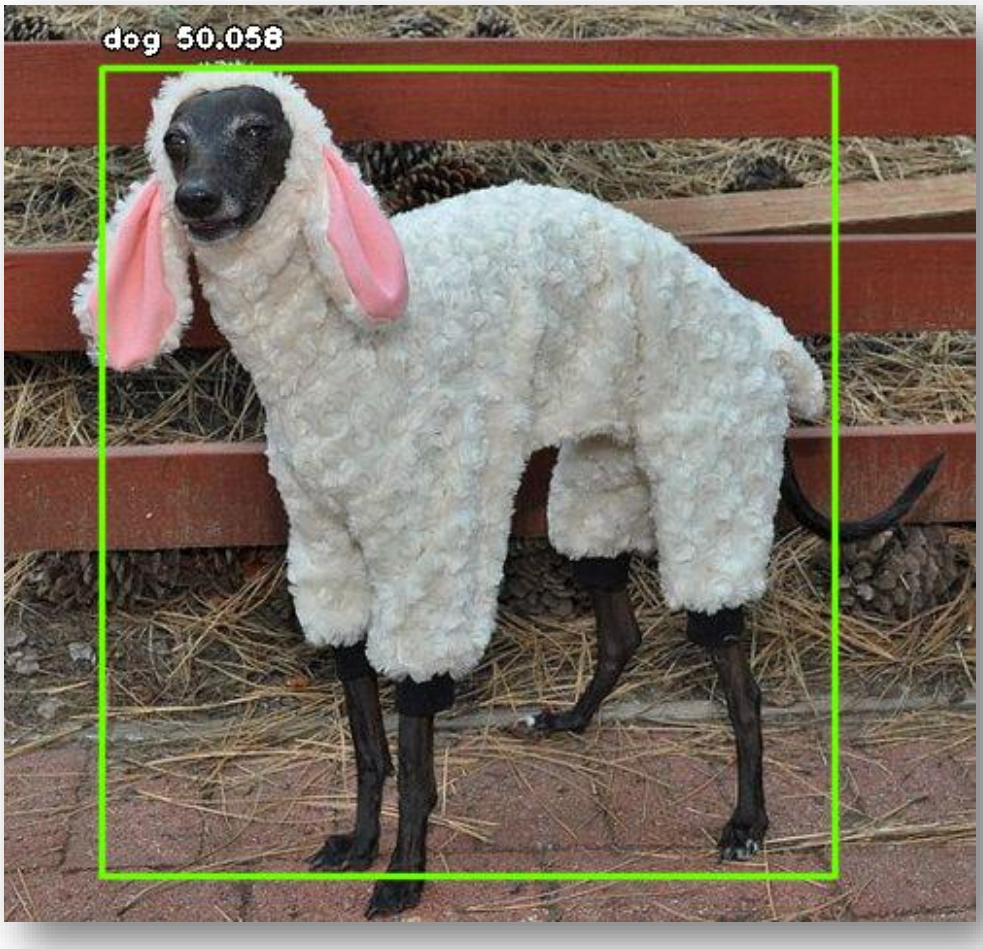
MORE WETRAIN MORE ACCURATEWILL BE THE OUTPUT.

Note: In our project we only trained it for dog class making it static.

Chapter – 4

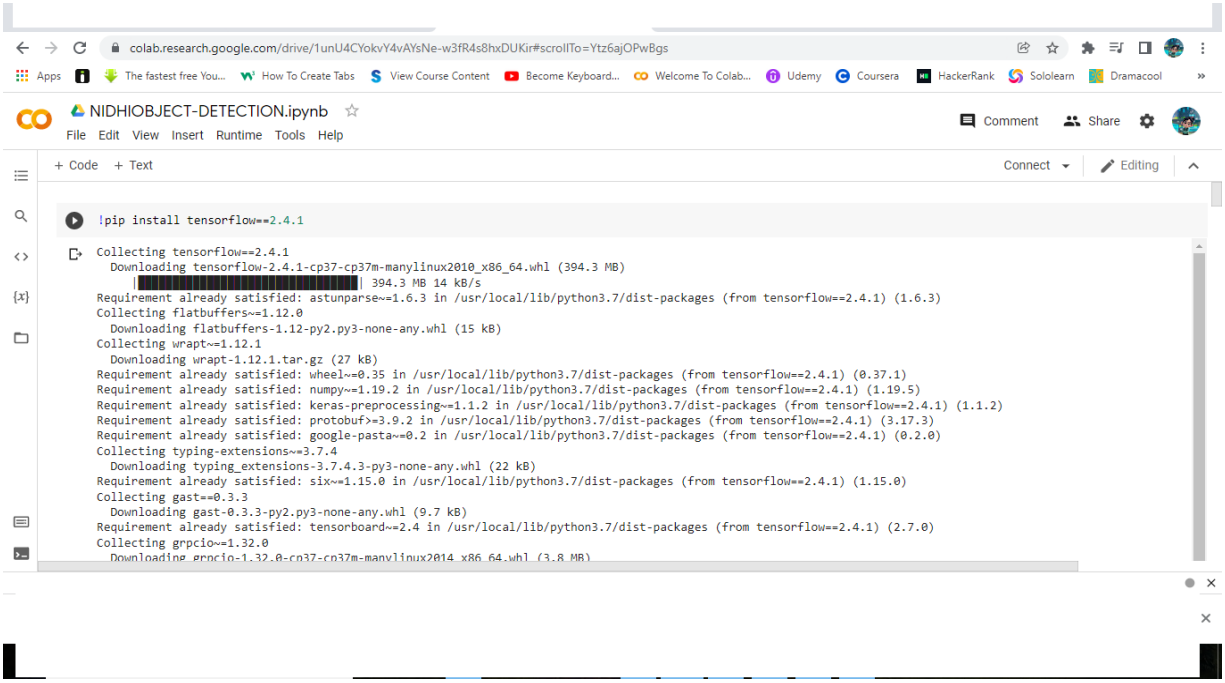
OUTPUT

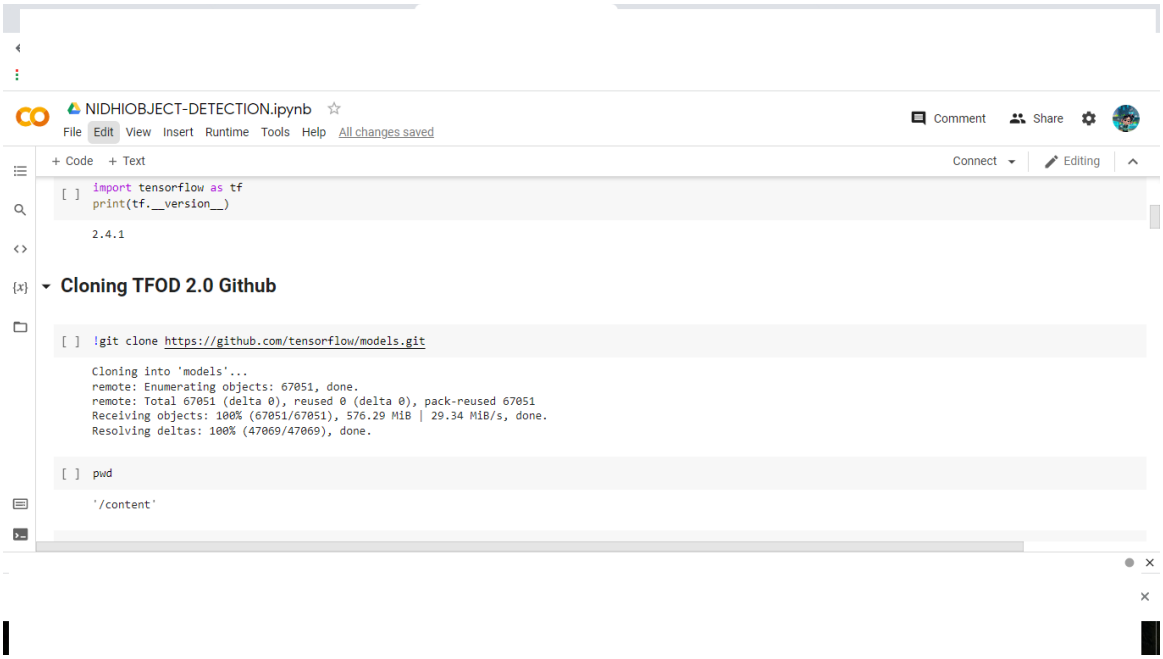

dog 82.043

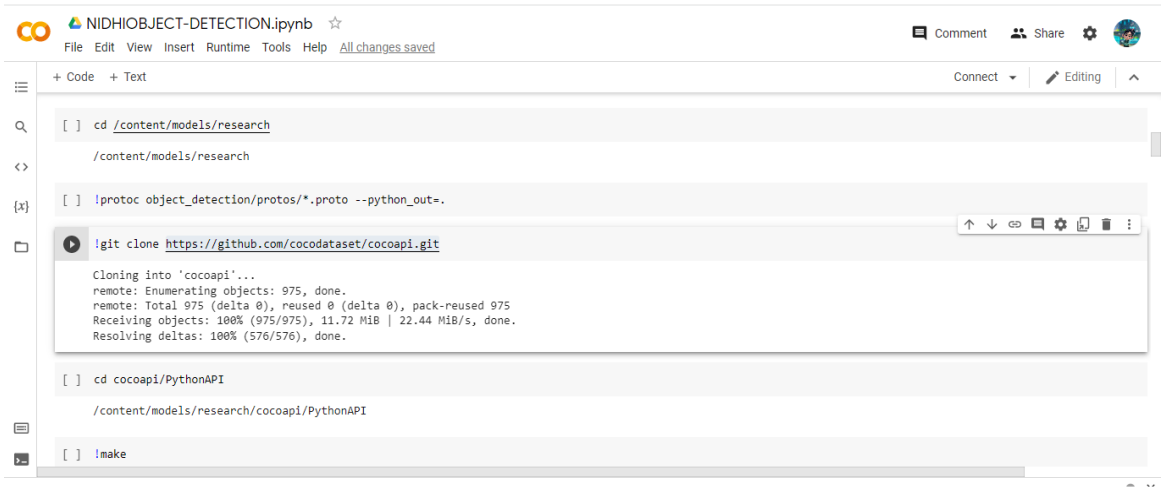

dog 50.058

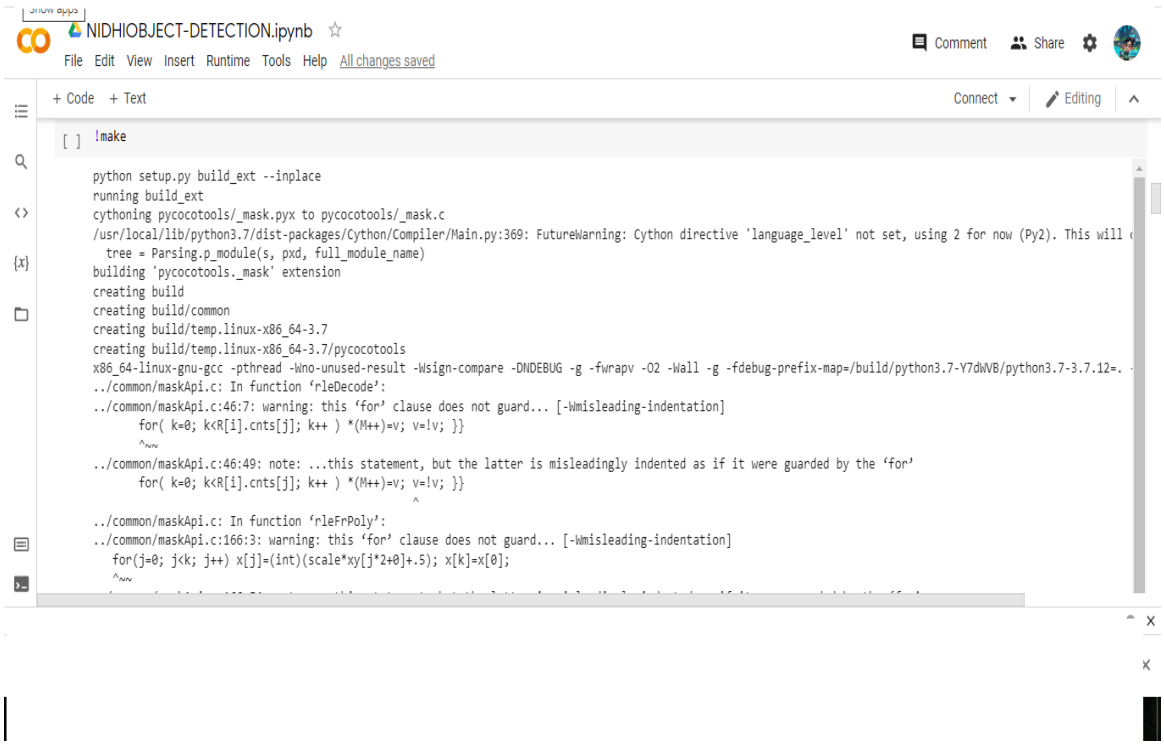
Chapter – 5 APPENDIX

IMPLEMENATATION CODE

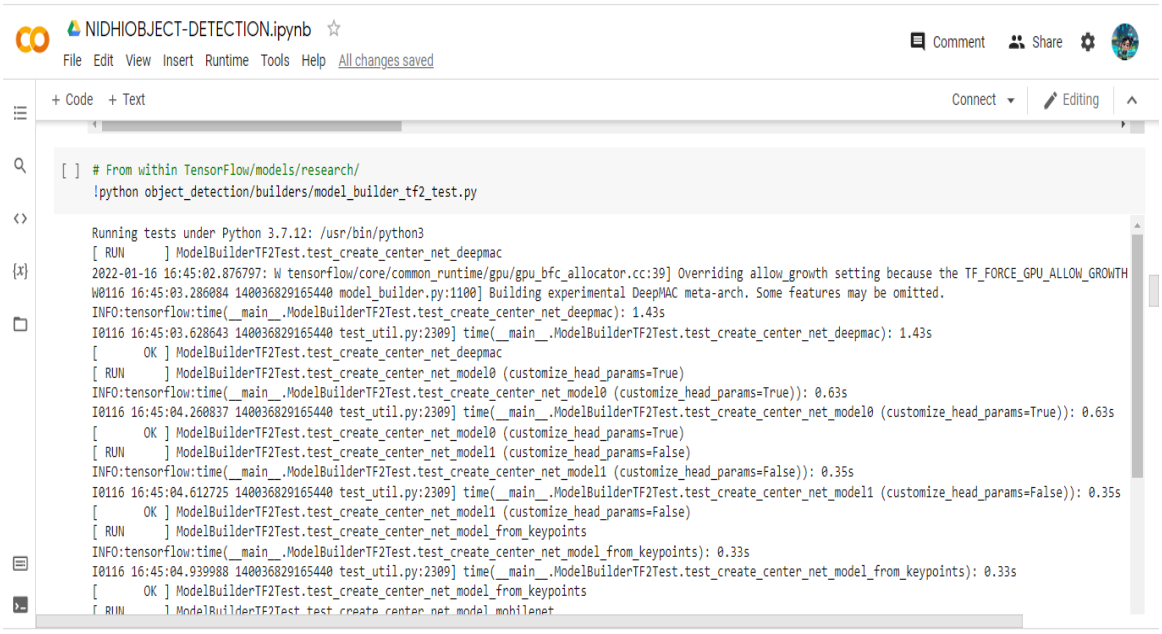




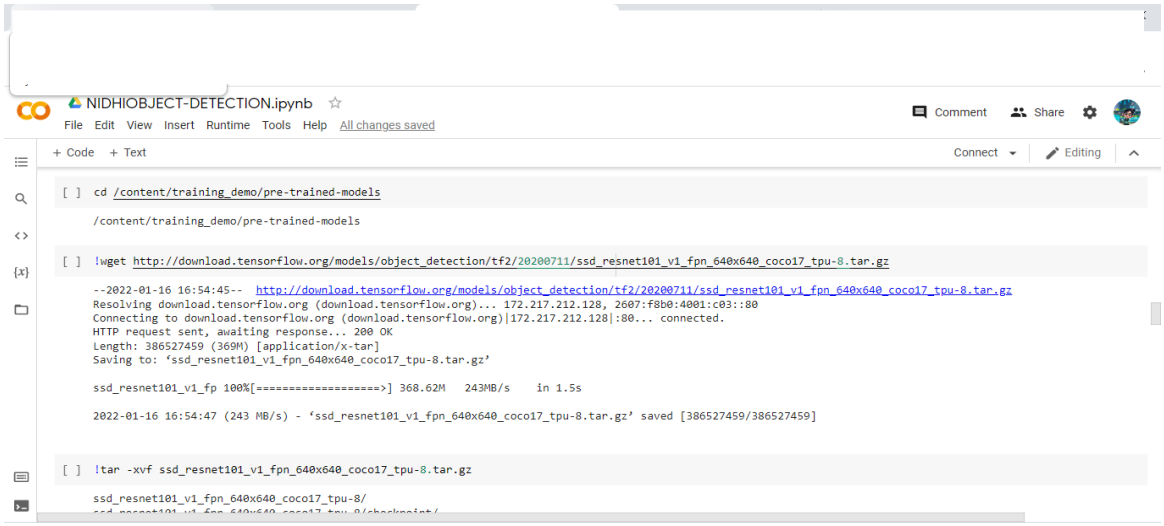

NIDHIOBJECT-DETECTION.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
Connect
Editing
[ ] cd /content/models/research
/content/models/research
[ ] !protoc object_detection/protos/*.proto --python_out=.
!git clone https://github.com/cocodataset/cocoapi.git
Cloning into 'cocoapi'...
remote: Enumerating objects: 975, done.
remote: Total 975 (delta 0), reused 0 (delta 0), pack-reused 975
Receiving objects: 100% (975/975), 11.72 MiB | 22.44 MiB/s, done.
Resolving deltas: 100% (576/576), done.
[ ] cd cocoapi/PythonAPI
/content/models/research/cocoapi/PythonAPI
[ ] !make

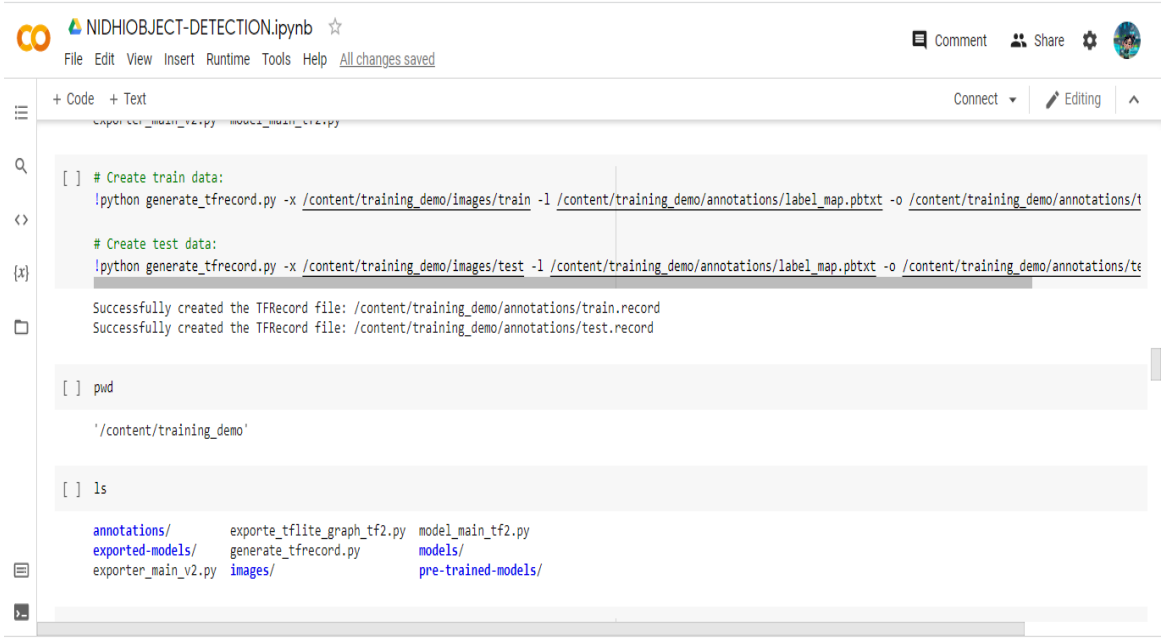

NIDHIOBJECT-DETECTION.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
Connect
Editing
[ ] !make
python setup.py build_ext --inplace
running build_ext
cythoning pycocotools/_mask.pyx to pycocotools/_mask.c
/usr/local/lib/python3.7/dist-packages/Cython/Compiler/Main.py:369: FutureWarning: Cython directive 'language_level' not set, using 2 for now (Py2). This will
tree = Parsing.p_module(s, pxd, full_module_name)
building 'pycocotools._mask' extension
creating build
creating build/common
creating build/temp.linux-x86_64-3.7
creating build/temp.linux-x86_64-3.7/pycocotools
x86_64-linux-gnu-gcc -pthread -Wno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -O2 -Wall -g -fdebug-prefix-map=/build/python3.7-Y7dWVB/python3.7-3.7.12=.
../common/maskApi.c: In function ‘rleDecode’:
../common/maskApi.c:46:7: warning: this ‘for’ clause does not guard... [-Wmisleading-indentation]
for(k=0; k<R[i].cnts[j]; k++) *(M++)=v; v=!v; }}
^~~
../common/maskApi.c:46:49: note: ...this statement, but the latter is misleadingly indented as if it were guarded by the ‘for’
for(k=0; k<R[i].cnts[j]; k++) *(M++)=v; v=!v; }}
^
../common/maskApi.c: In function ‘rleFrPoly’:
../common/maskApi.c:166:3: warning: this ‘for’ clause does not guard... [-Wmisleading-indentation]
for(j=0; j<k; j++) x[j]=(int)(scale*xy[j*2+0]+.5); x[k]=x[0];
^~~


NIDHIOBJECT-DETECTION.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
Connect
Editing
[ ] cp object_detection/packages/tf2/setup.py .
[ ] !python -m pip install .
Collecting apache-beam
Downloading apache_beam-2.35.0-cp37-cp37m-manylinux2010_x86_64.whl (9.9 MB)
9.9 MB 8.9 MB/s
Requirement already satisfied: pillow in /usr/local/lib/python3.7/dist-packages (from object-detection==0.1) (7.1.2)
Requirement already satisfied: lxml in /usr/local/lib/python3.7/dist-packages (from object-detection==0.1) (4.2.6)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (from object-detection==0.1) (3.2.2)
Requirement already satisfied: Cython in /usr/local/lib/python3.7/dist-packages (from object-detection==0.1) (0.29.26)
Requirement already satisfied: contextlib2 in /usr/local/lib/python3.7/dist-packages (from object-detection==0.1) (0.5.5)
Collecting tf-slim
Downloading tf_slim-1.1.0-py2.py3-none-any.whl (352 kB)
352 kB 37.5 MB/s
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from object-detection==0.1) (1.15.0)
Requirement already satisfied: pycocotools in /usr/local/lib/python3.7/dist-packages (from object-detection==0.1) (2.0.4)
Collecting lvis
Downloading lvis-0.5.3-py3-none-any.whl (14 kB)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from object-detection==0.1) (1.4.1)
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages (from object-detection==0.1) (1.1.5)
Collecting tf-models-official>=2.5.1
Downloading tf_models_official-2.7.0-py2.py3-none-any.whl (1.8 MB)

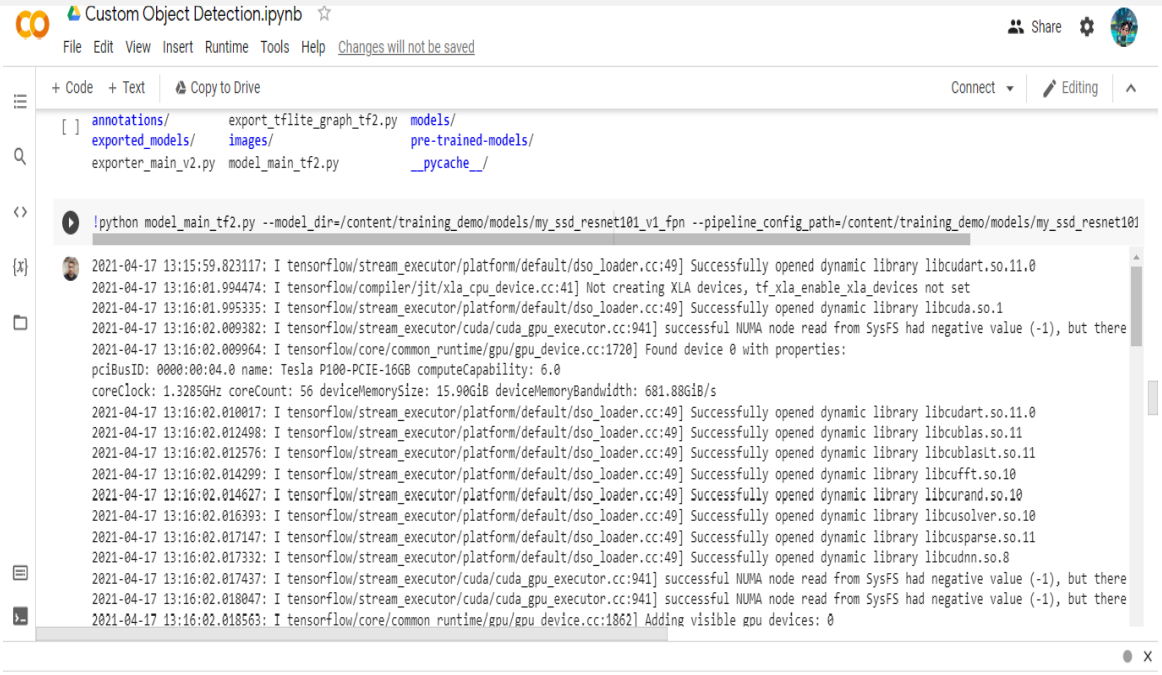

NIDHIOBJECT-DETECTION.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
Connect
Editing
[ ] # From within TensorFlow/models/research/
!python object_detection/builders/model_builder_tf2_test.py
Running tests under Python 3.7.12: /usr/bin/python3
[RUN] ModelBuilderTF2Test.test_create_center_net_deepmac
2022-01-16 16:45:02.876797: W tensorflow/core/common_runtime/gpu/gpu_bfc_allocator.cc:39] Overriding allow_growth setting because the TF_FORCE_GPU_ALLOW_GROWTH
W0116 16:45:03.286084 140036829165440 model_builder.py:1100] Building experimental DeepMAC meta-arch. Some features may be omitted.
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_create_center_net_deepmac): 1.43s
I0116 16:45:03.628643 140036829165440 test_util.py:2309] time(__main__.ModelBuilderTF2Test.test_create_center_net_deepmac): 1.43s
[OK] ModelBuilderTF2Test.test_create_center_net_deepmac
[RUN] ModelBuilderTF2Test.test_create_center_net_model0 (customize_head_params=True)
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_create_center_net_model0 (customize_head_params=True)): 0.63s
I0116 16:45:04.260837 140036829165440 test_util.py:2309] time(__main__.ModelBuilderTF2Test.test_create_center_net_model0 (customize_head_params=True)): 0.63s
[OK] ModelBuilderTF2Test.test_create_center_net_model0 (customize_head_params=True)
[RUN] ModelBuilderTF2Test.test_create_center_net_model1 (customize_head_params=False)
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_create_center_net_model1 (customize_head_params=False)): 0.35s
I0116 16:45:04.612725 140036829165440 test_util.py:2309] time(__main__.ModelBuilderTF2Test.test_create_center_net_model1 (customize_head_params=False)): 0.35s
[OK] ModelBuilderTF2Test.test_create_center_net_model1 (customize_head_params=False)
[RUN] ModelBuilderTF2Test.test_create_center_net_model_from_keypoints
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_create_center_net_model_from_keypoints): 0.33s
I0116 16:45:04.939988 140036829165440 test_util.py:2309] time(__main__.ModelBuilderTF2Test.test_create_center_net_model_from_keypoints): 0.33s
[OK] ModelBuilderTF2Test.test_create_center_net_model_from_keypoints

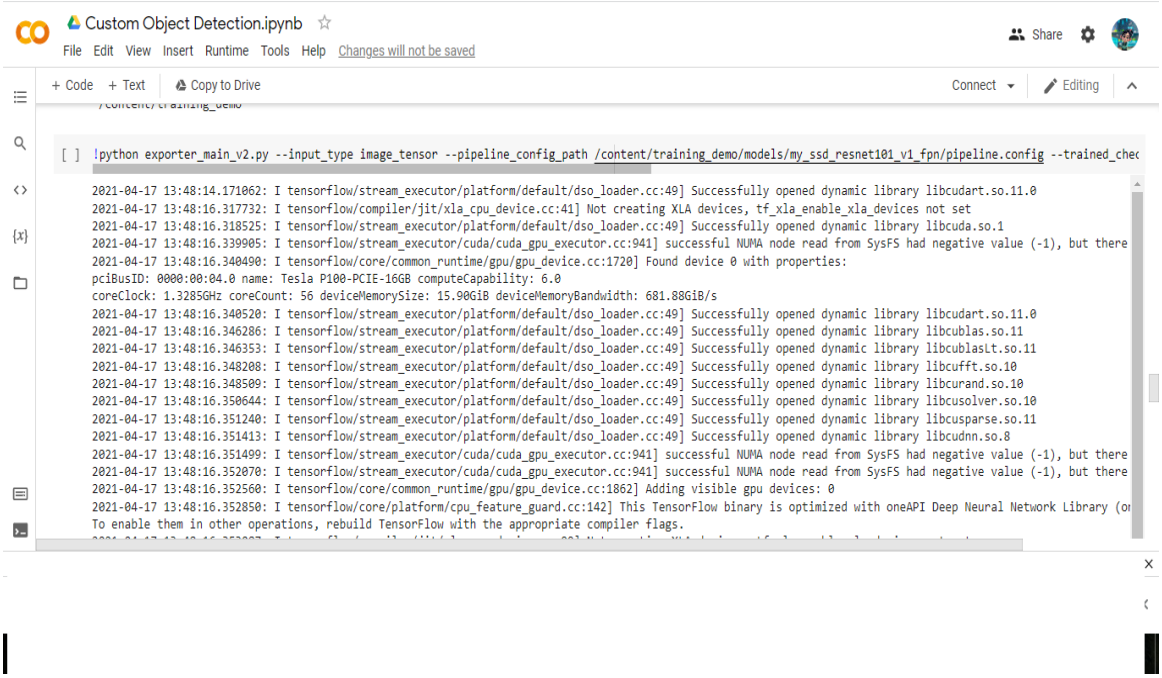
TF RECORD FILE



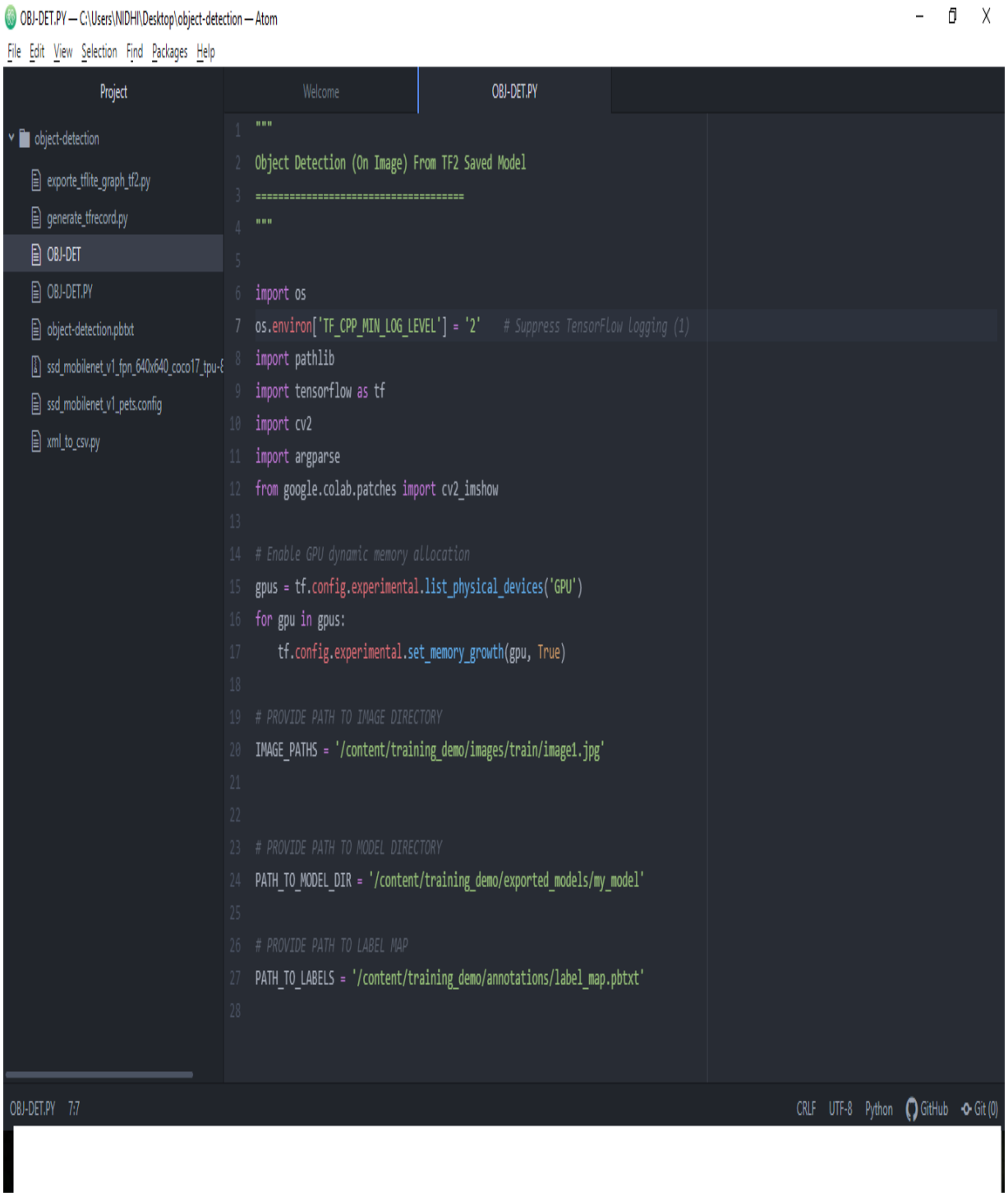


TRAINING STEP

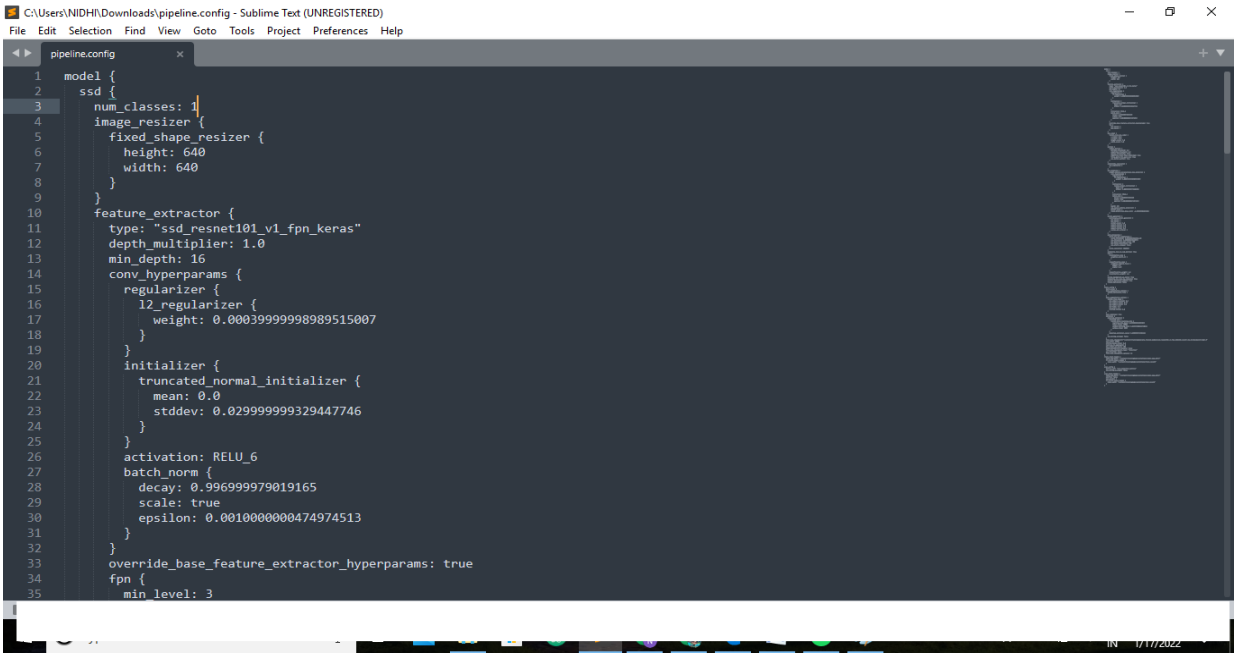


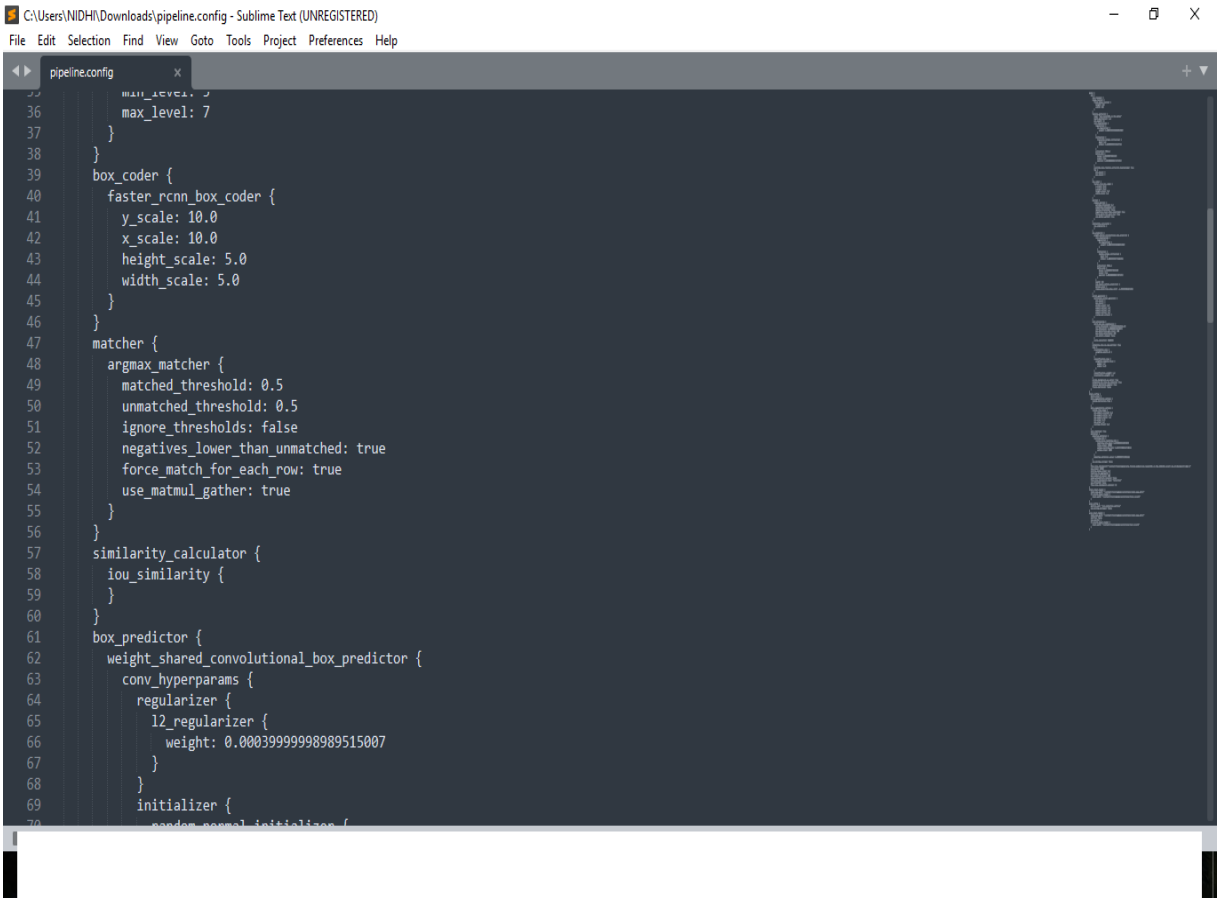


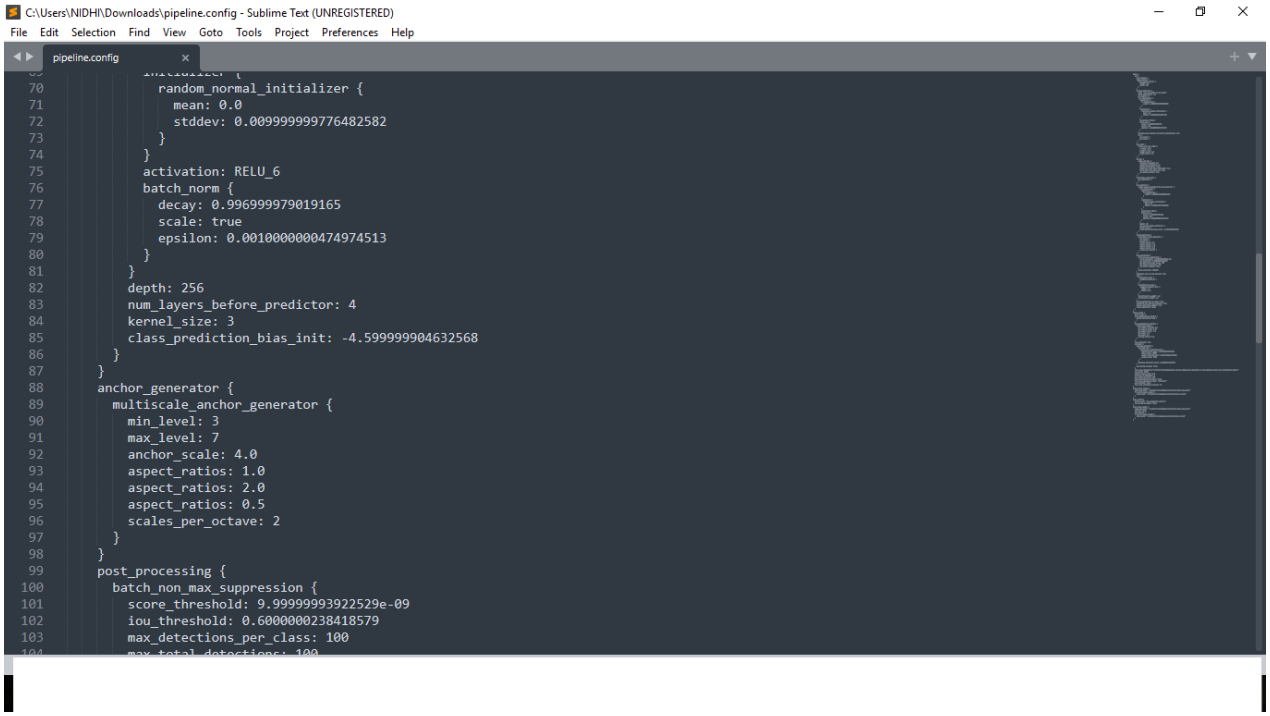
OBJECT-DETECTION CODE FOR TESTING IMAGES ON TEST LEVEL:



CONFIGURATION FILE

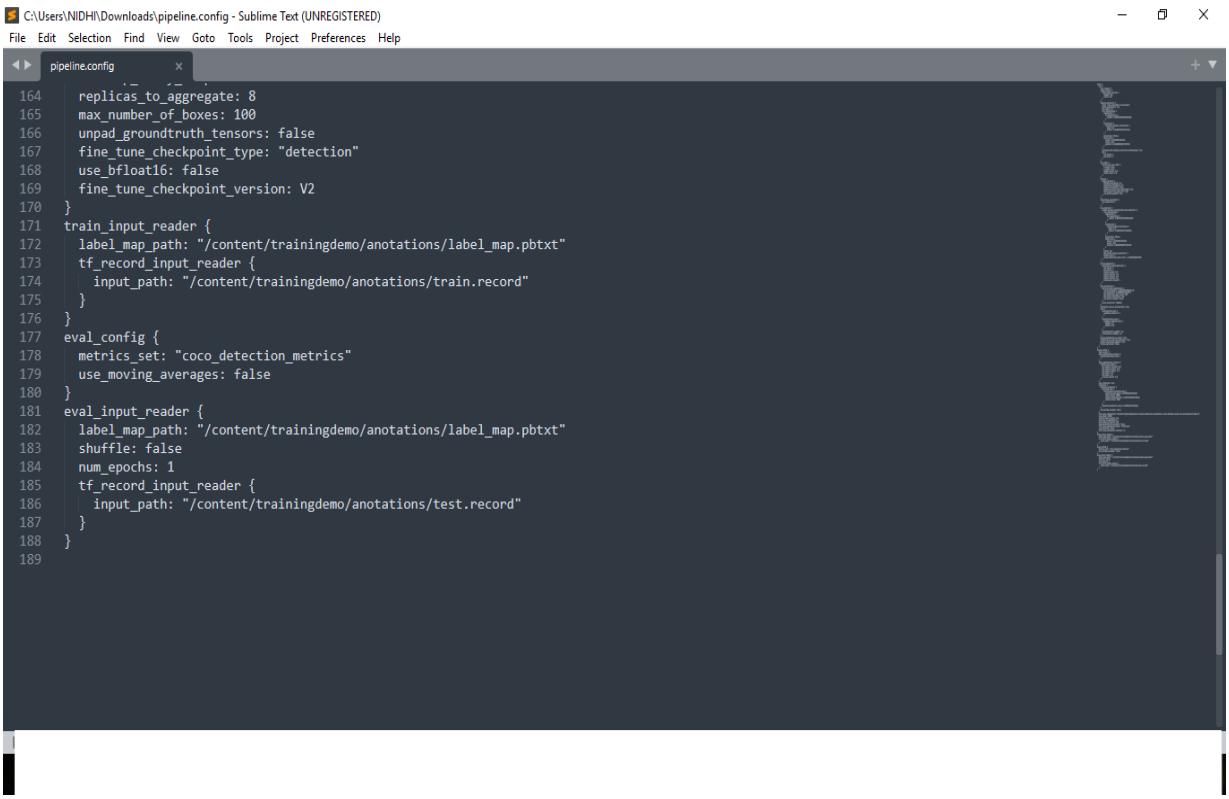





C:\Users\NIDHI\Downloads\pipeline.config - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
pipeline.config
random_normal_initializer {
mean: 0.0
stddev: 0.009999999776482582
}
}
activation: RELU_6
batch_norm {
decay: 0.996999979019165
scale: true
epsilon: 0.0010000000474974513
}
}
depth: 256
num_layers_before_predictor: 4
kernel_size: 3
class_prediction_bias_init: -4.599999904632568
}
}
anchor_generator {
multiscale_anchor_generator {
min_level: 3
max_level: 7
anchor_scale: 4.0
aspect_ratios: 1.0
aspect_ratios: 2.0
aspect_ratios: 0.5
scales_per_octave: 2
}
}
post_processing {
batch_non_max_suppression {
score_threshold: 9.99999993922529e-09
iou_threshold: 0.6000000238418579
max_detections_per_class: 100


C:\Users\NIDHI\Downloads\pipeline.config - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
pipeline.config
use_static_shapes: false
}
score_converter: SIGMOID
}
normalize_loss_by_num_matches: true
loss {
localization_loss {
weighted_smooth_l1 {
}
}
classification_loss {
weighted_sigmoid_focal {
gamma: 2.0
alpha: 0.25
}
}
classification_weight: 1.0
localization_weight: 1.0
}
encode_background_as_zeros: true
normalize_loc_loss_by_codesize: true
inplace_batchnorm_update: true
freeze_batchnorm: false
}
}
train_config {
batch_size: 8
data_augmentation_options {
random_horizontal_flip {
}
}
data_augmentation_options {
random_crop_image {
min_object_covered: 0.0


C:\Users\NIDHI\Downloads\pipeline.config - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
pipeline.config
max_aspect_ratio: 3.0
min_area: 0.75
max_area: 1.0
overlap_thresh: 0.0
}
}
sync_replicas: true
optimizer {
momentum_optimizer {
learning_rate {
cosine_decay_learning_rate {
learning_rate_base: 0.03999999910593033
total_steps: 25000
warmup_learning_rate: 0.013333000242710114
warmup_steps: 2000
}
}
momentum_optimizer_value: 0.8999999761581421
}
use_moving_average: false
}
fine_tune_checkpoint:"/content/trainingdemo/pre-trained-models/ssd_resnet101_v1_fpn_640x640_coco17_tpu-8/checkpoint/ckpt-0"
num_steps: 25000
startup_delay_steps: 0.0
replicas_to_aggregate: 8
max_number_of_boxes: 100
unpad_groundtruth_tensors: false
fine_tune_checkpoint_type: "detection"
use_bfloat16: false
fine_tune_checkpoint_version: V2
}
train_input_reader {
label_map_path: "/content/trainingdemo/anotations/label_map.pbtxt"
tf_record_input_reader {


C:\Users\NIDHI\Downloads\pipeline.config - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
pipeline.config
replicas_to_aggregate: 8
max_number_of_boxes: 100
unpad_groundtruth_tensors: false
fine_tune_checkpoint_type: "detection"
use_bfloat16: false
fine_tune_checkpoint_version: V2
}
train_input_reader {
label_map_path: "/content/trainingdemo/anotations/label_map.pbtxt"
tf_record_input_reader {
input_path: "/content/trainingdemo/anotations/train.record"
}
}
eval_config {
metrics_set: "coco_detection_metrics"
use_moving_averages: false
}
eval_input_reader {
label_map_path: "/content/trainingdemo/anotations/label_map.pbtxt"
shuffle: false
num_epochs: 1
tf_record_input_reader {
input_path: "/content/trainingdemo/anotations/test.record"
}
}

REFERENCES

1. TFOD custom object detection step by step

https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/

2. GOOGLE COLABORATORY

https://colab.research.google.com/?utm_source=scs-index

3.Tensorflow Github

https://github.com/tensorflow/models/tree/master/research/object_detection

3. Images (GOOGLE): https://www.google.co.in/
4. Stack Overflow (for issues)

https://stackoverflow.com/