



Real Time Monument Monitoring System

Group 2

Shubham Jha
Subhadip Jana
Saurav Kumar
Vanita Sharma

An aerial photograph of Rome, Italy, featuring the Colosseum in the center. To the right of the Colosseum is the Temple of Mars Ultor. The surrounding city is densely packed with buildings, and the sky is filled with white clouds. The image is overlaid with a dark blue gradient and decorative white dashed lines on the left and right sides, and a solid orange line at the bottom right.

+ To make a prototype that identifies monuments from the Space

+Technologies we are using :



Python



TensorFlow Object Detection API 2.0





TensorFlow Object Detection API 2.0

SSD ResNet101 V1 FPN 640X640
(RetinaNet 101)

+ 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.

A photograph of an astronaut in a white spacesuit floating in space. The astronaut's helmet is on the left, and their body extends towards the right. In the background, there are various pieces of orbital equipment, including a large white cylindrical tank, a yellow curved structure, and a grid-like structure. The text "Our Approach" is overlaid in the center in a white serif font, with a small blue plus sign below the letter 'A'.

Our Approach

Collecting Dataset

Gathering images of famous monuments



Sources :

- Google Images
- Google Earth
- iStock
- Shutterstock

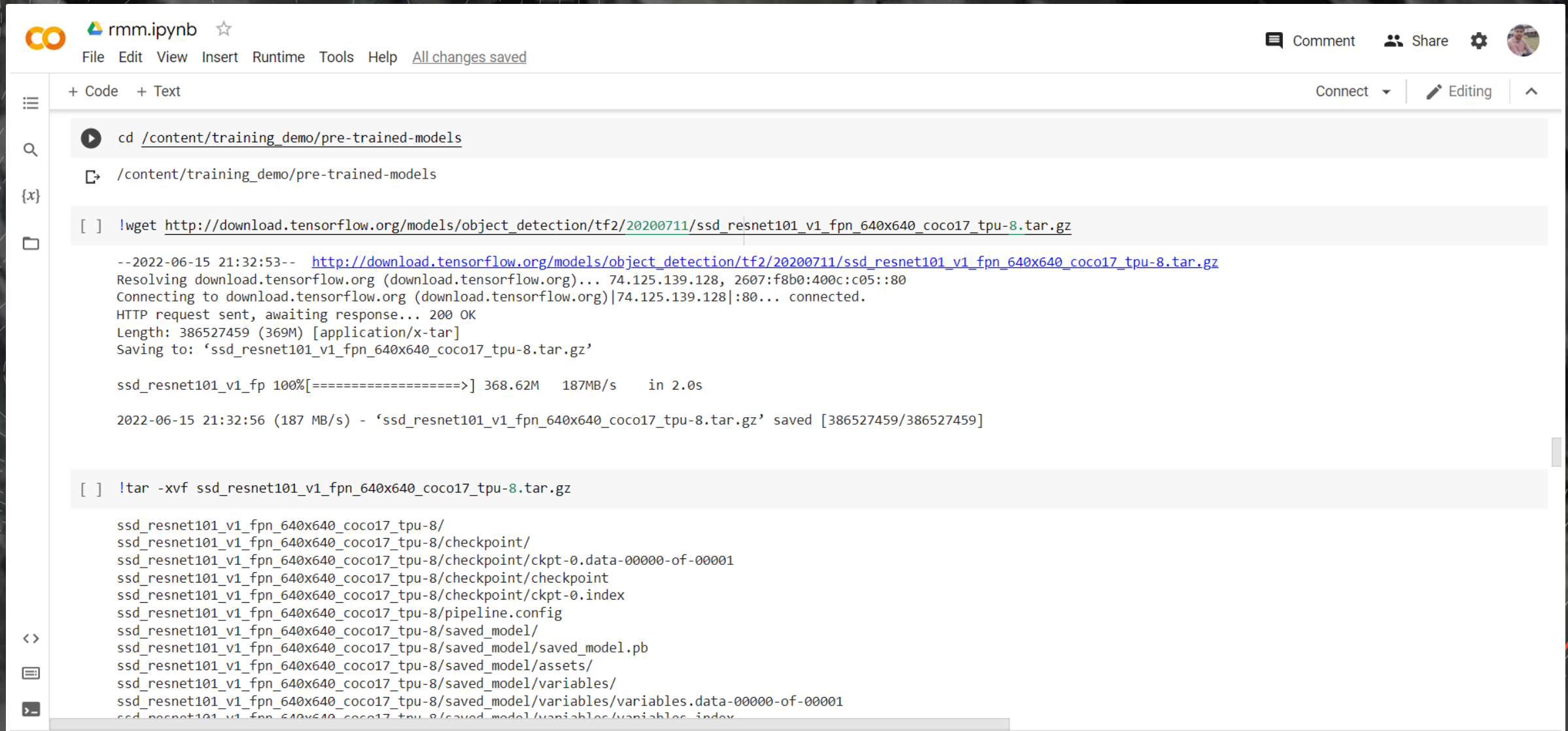
Making Annotation



Generating XML File for Annotations

```
Taj_Mahal_IMG8.xml X
> Users > acer > Downloads > labels_my-project-name_2022-06-16-05-23-27 > Taj_Mahal_IMG8.
1  <annotation>
2      <folder>my-project-name</folder>
3      <filename>Taj_Mahal_IMG8.jpg</filename>
4      <path>/my-project-name/Taj_Mahal_IMG8.jpg</path>
5      <source>
6          <database>Unspecified</database>
7      </source>
8      <size>
9          <width>750</width>
10         <height>496</height>
11         <depth>3</depth>
12     </size>
13     <object>
14         <name>Taj Mahal</name>
15         <pose>Unspecified</pose>
16         <truncated>Unspecified</truncated>
17         <difficult>Unspecified</difficult>
18         <bndbox>
19             <xmin>320</xmin>
20             <ymin>102</ymin>
21             <xmax>443</xmax>
22             <ymax>240</ymax>
23         </bndbox>
24     </object>
25 </annotation>
```


Getting Pre-trained Model



The screenshot shows a Jupyter Notebook interface with a dark theme. The top bar includes the RMM logo, the name 'rmm.ipynb', and a star icon. On the right, there are links for 'Comment', 'Share', and a settings gear. Below the top bar is a menu with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', 'Help', and 'All changes saved'. The left sidebar contains icons for file explorer, search, and code editor. The main area displays the following code and output:

```
+ Code + Text
```

```
cd /content/training_demo/pre-trained-models
```

```
/content/training_demo/pre-trained-models
```

```
[ ] !wget http://download.tensorflow.org/models/object_detection/tf2/20200711/ssd_resnet101_v1_fpn_640x640_coco17_tpu-8.tar.gz
```

```
--2022-06-15 21:32:53-- http://download.tensorflow.org/models/object_detection/tf2/20200711/ssd_resnet101_v1_fpn_640x640_coco17_tpu-8.tar.gz
Resolving download.tensorflow.org (download.tensorflow.org)... 74.125.139.128, 2607:f8b0:400c:c05::80
Connecting to download.tensorflow.org (download.tensorflow.org)|74.125.139.128|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 386527459 (369M) [application/x-tar]
Saving to: 'ssd_resnet101_v1_fpn_640x640_coco17_tpu-8.tar.gz'
```

```
ssd_resnet101_v1_fp 100%[=====>] 368.62M 187MB/s in 2.0s
```

```
2022-06-15 21:32:56 (187 MB/s) - 'ssd_resnet101_v1_fpn_640x640_coco17_tpu-8.tar.gz' saved [386527459/386527459]
```

```
[ ] !tar -xvf ssd_resnet101_v1_fpn_640x640_coco17_tpu-8.tar.gz
```

```
ssd_resnet101_v1_fpn_640x640_coco17_tpu-8/
ssd_resnet101_v1_fpn_640x640_coco17_tpu-8/checkpoint/
ssd_resnet101_v1_fpn_640x640_coco17_tpu-8/checkpoint/ckpt-0.data-00000-of-00001
ssd_resnet101_v1_fpn_640x640_coco17_tpu-8/checkpoint/checkpoint
ssd_resnet101_v1_fpn_640x640_coco17_tpu-8/checkpoint/ckpt-0.index
ssd_resnet101_v1_fpn_640x640_coco17_tpu-8/pipeline.config
ssd_resnet101_v1_fpn_640x640_coco17_tpu-8/saved_model/
ssd_resnet101_v1_fpn_640x640_coco17_tpu-8/saved_model/saved_model.pb
ssd_resnet101_v1_fpn_640x640_coco17_tpu-8/saved_model/assets/
ssd_resnet101_v1_fpn_640x640_coco17_tpu-8/saved_model/variables/
ssd_resnet101_v1_fpn_640x640_coco17_tpu-8/saved_model/variables/variables.data-00000-of-00001
ssd_resnet101_v1_fpn_640x640_coco17_tpu-8/saved_model/variables/variables.index
```


Training Custom Monument Detector



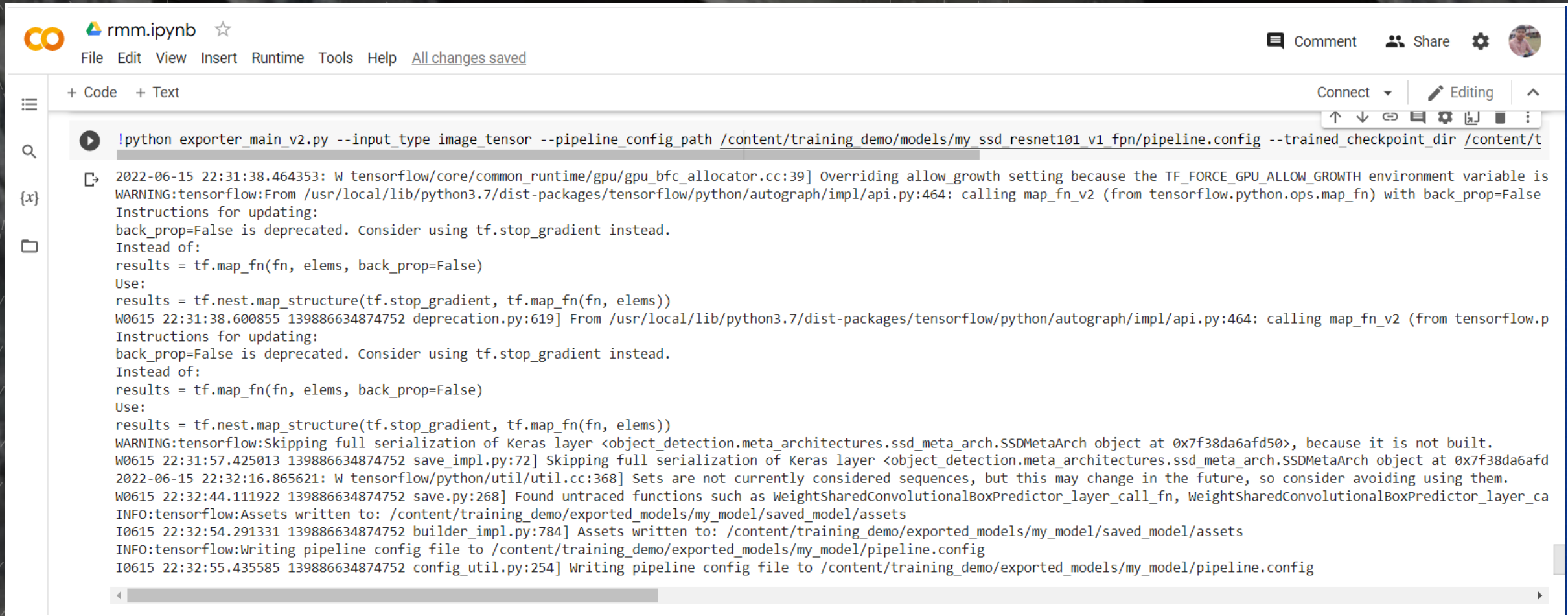
The screenshot displays a Jupyter Notebook interface with a dark-themed sidebar on the left containing icons for file explorer, search, and code execution. The main area shows a terminal window with the following content:

```
python model_main_tf2.py --model_dir=/content/training_demo/models/my_ssd_resnet101_v1_fpn --pipeline_config_path=/content/training_demo/models/my_ssd_resnet101_v1_fpn/pipeline.config
```

The terminal output shows the training progress, including loss values and learning rate for several steps:

```
'Loss/regularization_loss': 2.1549718,
'Loss/total_loss': 2.4730835,
'learning_rate': 0.025333151}
I0615 22:07:59.587969 139747739174784 model_lib_v2.py:708] {'Loss/classification_loss': 0.20783213,
'Loss/localization_loss': 0.11027953,
'Loss/regularization_loss': 2.1549718,
'Loss/total_loss': 2.4730835,
'learning_rate': 0.025333151}
INFO:tensorflow:Step 1000 per-step time 0.787s
I0615 22:09:18.327965 139747739174784 model_lib_v2.py:707] Step 1000 per-step time 0.787s
INFO:tensorflow:({'Loss/classification_loss': 0.16494603,
'Loss/localization_loss': 0.14156711,
'Loss/regularization_loss': 2.1134496,
'Loss/total_loss': 2.4199626,
'learning_rate': 0.0266665}
I0615 22:09:18.328323 139747739174784 model_lib_v2.py:708] {'Loss/classification_loss': 0.16494603,
'Loss/localization_loss': 0.14156711,
'Loss/regularization_loss': 2.1134496,
'Loss/total_loss': 2.4199626,
'learning_rate': 0.0266665}
INFO:tensorflow:Step 1100 per-step time 0.808s
I0615 22:10:39.099520 139747739174784 model_lib_v2.py:707] Step 1100 per-step time 0.808s
INFO:tensorflow:({'Loss/classification_loss': 0.2859589,
'Loss/localization_loss': 0.071680404,
'Loss/regularization_loss': 2.0793965,
'Loss/total_loss': 2.4370358,
'learning_rate': 0.02799985}
I0615 22:10:39.099871 139747739174784 model_lib_v2.py:708] {'Loss/classification_loss': 0.2859589,
'Loss/localization_loss': 0.071680404,
'Loss/regularization_loss': 2.0793965,
'Loss/total_loss': 2.4370358,
'learning_rate': 0.02799985}
```


Exporting The Model



The screenshot displays a JupyterLab environment with a terminal window showing the execution of a Python script. The terminal output includes several warnings and informational messages from TensorFlow and Keras, indicating the successful export of a model. The script being run is `python exporter_main_v2.py` with various command-line arguments.

```
!python exporter_main_v2.py --input_type image_tensor --pipeline_config_path /content/training_demo/models/my_ssd_resnet101_v1_fpn/pipeline.config --trained_checkpoint_dir /content/t
```

2022-06-15 22:31:38.464353: W tensorflow/core/common_runtime/gpu/gpu_bfc_allocator.cc:39] Overriding allow_growth setting because the TF_FORCE_GPU_ALLOW_GROWTH environment variable is
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/python/autograph/impl/api.py:464: calling map_fn_v2 (from tensorflow.python.ops.map_fn) with back_prop=False
Instructions for updating:
back_prop=False is deprecated. Consider using tf.stop_gradient instead.
Instead of:
results = tf.map_fn(fn, elems, back_prop=False)
Use:
results = tf.nest.map_structure(tf.stop_gradient, tf.map_fn(fn, elems))
W0615 22:31:38.600855 139886634874752 deprecation.py:619] From /usr/local/lib/python3.7/dist-packages/tensorflow/python/autograph/impl/api.py:464: calling map_fn_v2 (from tensorflow.p
Instructions for updating:
back_prop=False is deprecated. Consider using tf.stop_gradient instead.
Instead of:
results = tf.map_fn(fn, elems, back_prop=False)
Use:
results = tf.nest.map_structure(tf.stop_gradient, tf.map_fn(fn, elems))
WARNING:tensorflow:Skipping full serialization of Keras layer <object_detection.meta_architectures.ssd_meta_arch.SSDMetaArch object at 0x7f38da6afd50>, because it is not built.
W0615 22:31:57.425013 139886634874752 save_impl.py:72] Skipping full serialization of Keras layer <object_detection.meta_architectures.ssd_meta_arch.SSDMetaArch object at 0x7f38da6afd
2022-06-15 22:32:16.865621: W tensorflow/python/util/util.cc:368] Sets are not currently considered sequences, but this may change in the future, so consider avoiding using them.
W0615 22:32:44.111922 139886634874752 save.py:268] Found untraced functions such as WeightSharedConvolutionalBoxPredictor_layer_call_fn, WeightSharedConvolutionalBoxPredictor_layer_ca
INFO:tensorflow:Assets written to: /content/training_demo/exported_models/my_model/saved_model/assets
I0615 22:32:54.291331 139886634874752 builder_impl.py:784] Assets written to: /content/training_demo/exported_models/my_model/saved_model/assets
INFO:tensorflow:Writing pipeline config file to /content/training_demo/exported_models/my_model/pipeline.config
I0615 22:32:55.435585 139886634874752 config_util.py:254] Writing pipeline config file to /content/training_demo/exported_models/my_model/pipeline.config

Testing The Model

+ Code + Text

Connect Editing

```
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2' # Suppress TensorFlow logging (1)
import pathlib
import tensorflow as tf
import cv2
import argparse
from google.colab.patches import cv2_imshow

# Enable GPU dynamic memory allocation
gpus = tf.config.experimental.list_physical_devices('GPU')
for gpu in gpus:
    tf.config.experimental.set_memory_growth(gpu, True)

# PROVIDE PATH TO IMAGE DIRECTORY
IMAGE_PATHS = '/content/training_demo/images/test/Screenshot 2022-06-15 223223.png'

# PROVIDE PATH TO MODEL DIRECTORY
PATH_TO_MODEL_DIR = '/content/training_demo/exported_models/my_model'

# PROVIDE PATH TO LABEL MAP
PATH_TO_LABELS = '/content/training_demo/annotations/label_map.pbtxt'

# PROVIDE THE MINIMUM CONFIDENCE THRESHOLD
MIN_CONF_THRESH = float(0.60)

# LOAD THE MODEL

import time
from object_detection.utils import label_map_util
from object_detection.utils import visualization_utils as viz_utils
```



Input Image

Final Output

+ Code + Text


✓ RAM
Disk

Editing

✓ 29s


```
print('Done')  
# DISPLAYS OUTPUT IMAGE  
cv2_imshow(image_with_detections)  
# CLOSSES WINDOW ONCE KEY IS PRESSED
```

↳ Loading model...Done! Took 24.462425708770752 seconds
Running inference for /content/training_demo/images/test/Screenshot 2022-06-15 223223.png... Done



✓ 29s completed at 4:25 AM

✕



30°C Haze

04:27
16-06-2022

ENG

2

An astronaut in a white spacesuit is floating in space, positioned on the left side of the frame. The astronaut's helmet is dark, and their arms are slightly outstretched. In the background, the planet Saturn is visible on the right, showing its characteristic rings. The surrounding space is filled with a vibrant nebula, displaying shades of blue, purple, and pink. A small, bright yellow star is located in the lower right quadrant. The entire image is overlaid with a pattern of white dashed lines that form concentric circles and wavy, organic shapes. A solid orange line curves along the bottom right corner.

Thank You