Data Science Lab
Tooth Detection Project 2조 최종보고

5기 김보아
5기 허유진
6기 박해균
6기 조수연

목차

1. 학습 내용

<Object Detection 개념 학습 및 프로젝트>

1. Object Detection 이해

   - Object Detection 문제영역 이해

   - Object Detection Metric: IoU, mAP

   - Pascal VOC, MS COCO, KITTI, Open Images

2. Google Colab, Tensorflow Object Detection API 학습

3. 대표적 Object Detection 모델 학습

   - R-CNN

   - Fast R-CNN

   - Faster R-CNN

   - Non-Maximum Suppression (NMS)

   - SSD (Single Shot Multibox Detector)

   - RetinaNet

   - CenterNet

4. Pre-Trained Object Detection Model 적용 프로젝트

   - Faster R-CNN을 이용한 Person Detection, Autopilot Detection, License Plate Detection Project

   - CenterNet을 이용한 Car Detection, Human Pose Estimation Project

   - SSD를 이용한 Face Detection Project

<Object Detection 논문 리뷰 스터디>

1. R-CNN 논문 리뷰
2. Fast R-CNN 논문 리뷰
3. Faster R-CNN 논문 리뷰
4. YOLO 논문 리뷰
5. SSD 논문 리뷰
6. VGGNet 논문 리뷰

2. 사용한 코드 설명

<[DSL]Tooth_Detection_Model.ipynb 파일 구성>

0. Default Setting

0. Default Setting

Executed in Colab pro environment.

- ML Framework
  - Python 3.7.11
  - Tensorflow 2.5.0
- Hardware
  - RAM: 12.7G
  - CPU: Intel(R) Xeon(R) CPU @ 2.30GHz (1core)

```
[ ] # Check GPU Availability
    gpu_info = !nvidia-smi
    gpu_info = '₩n'.join(gpu_info)
    if gpu_info.find('failed') >= 0:
      print('Select the Runtime > "Change runtime type" menu to enable a GPU accelerator, ')
      print('and then re-execute this cell.')
    else:
      print(gpu_info)

    Select the Runtime > "Change runtime type" menu to enable a GPU accelerator,
    and then re-execute this cell.
```

```
[ ] # Library import
    import os
    import json
    import cv2
    import matplotlib as mpl
    import matplotlib.pyplot as plt
    from natsort import natsorted, ns
    from object_detection.utils import config_util
    from object_detection.protos import pipeline_pb2
    from google.protobuf import text_format
    import tensorflow as tf
    import numpy as np
```

```
[ ] # Google Drive mount
    from google.colab import drive
    drive.mount('/content/drive')

    Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

딥러닝 모델을 다루기 이전에 모델, 데이터를 가져오는 과정으로 기본 환경을 설정하는 준비 단계이다. 딥러닝 모델을 빠르게 돌리기 위한 Colab pro의 RAM, GPU환경 설정에 대해 확인하고 모델, config파일, Data set(충치이미지 데이터)를 불러오기 위한 코드이다.

1. Install Tensorflow object detection API

1. Install Tensorflow object detection API

```
[ ] %%capture
    %%bash

    # Download tensorflow model garden
    mkdir /content/TensorFlow
    cd /content/TensorFlow
    git clone https://github.com/tensorflow/models.git

    # Install cocoapi
    cd /content
    git clone https://github.com/cocodataset/cocoapi.git
    cd /content/cocoapi/PythonAPI
    cp -r ./pycocotools /content/TensorFlow/models/research/

    # Protobuf installation
    cd /content/TensorFlow/models/research/
    protoc object_detection/protos/*.proto --python_out=.

    # Install object detection API
    cp object_detection/packages/tf2/setup.py .
    pip install --use-feature=2020-resolver .
```

```
[ ] %%bash

    # Check whether obj detection API is well installed
    cd /content/TensorFlow/models/research/
    python object_detection/builders/model_builder_tf2_test.py
```

```
2021-09-29 03:13:20.736976: E tensorflow/stream_executor/cuda/cuda_driver.cc:271] failed call to cuInit: CUDA_ERROR_NO_DEVICE: no CUDA-capable device is detected
2021-09-29 03:13:20.737040: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (45889216c9a2): /proc/driver/nvidia/version does not exist
Running tests under Python 3.7.12: /usr/bin/python3
[ RUN      ] ModelBuilderTF2Test.test_create_center_net_deepmac
W0929 03:13:21.054871 140634035390336 model_builder.py:1091] Building experimental DeepMAC meta-arch. Some features may be omitted.
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_create_center_net_deepmac): 0.69s
I0929 03:13:21.429259 140634035390336 test_util.py:2189] time(__main__.ModelBuilderTF2Test.test_create_center_net_deepmac): 0.69s
[       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)): 0.53s
I0929 03:13:21.955032 140634035390336 test_util.py:2189] time(__main__.ModelBuilderTF2Test.test_create_center_net_model0 (customize_head_params=True)): 0.53s
[       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)): 0.29s
I0929 03:13:22.241658 140634035390336 test_util.py:2189] time(__main__.ModelBuilderTF2Test.test_create_center_net_model1 (customize_head_params=False)): 0.29s
```

Object detection을 하기 위해 Tensorflow기반으로 만들어진 Tensorflow object detection API를 설치하는 코드이다.

2. Prepare Train data and EDA

2. Prepare Train data and EDA

```
[ ] %%capture
    %%bash
    # Prepare train data
    # Copy and unzip dataset in wd
    # Modify below directory properly based on your environment

    cp '/content/drive/MyDrive/Corporation_Project/tooth-sample-dsl(original).zip' '/content/tooth-sample-dsl.zip'
    unzip -q '/content/tooth-sample-dsl.zip' -d '/content/TensorFlow'
```

```
[ ] # Creating definition file(.pbtxt)

    labels = [{'name':'normal', 'id':1}, {'name':'caries', 'id':2}, ]

    !mkdir '/content/TensorFlow/tooth-sample-dsl/annotations'
    with open('/content/TensorFlow/tooth-sample-dsl/annotations/caries_map.pbtxt', 'w') as f:
        for label in labels:
            f.write('item { ₩n')
            f.write('₩tname:₩'{}₩'₩n'.format(label['name']))
            f.write('₩tid:{}₩n'.format(label['id']))
            f.write('}₩n')
```

```
mkdir: cannot create directory '/content/TensorFlow/tooth-sample-dsl/annotations': File exists
```

```
[ ] # Check pbtxt file

    def get_num_classes(pbtxt_fname):
        from object_detection.utils import label_map_util
        label_map = label_map_util.load_labelmap(pbtxt_fname)
        categories = label_map_util.convert_label_map_to_categories(
            label_map, max_num_classes=90, use_display_name=True)
        category_index = label_map_util.create_category_index(categories)
        return len(category_index.keys())
```

모델을 훈련시키기 위한 준비를 하는 코드로, 데이터와, 데이터에 대한 간단한 EDA를 하는 과정이다. train data(압축파일)를 불러와, 압축을 풀고, 데이터에 대한 정의서가 들어있는 pbtxt를 만들고 pbtxt 파일안에 데이터가 잘 정의되어 있는지 확인하기 위한 함수를 짠 코드이다.

```
[ ] # Get num_classes of pbtxt file (1: normal, 2: caries)

    wd = '/content/TensorFlow/tooth-sample-dsl'
    pbtxt_fname = os.path.join(wd,'annotations', 'caries_map.pbtxt')
    get_num_classes(pbtxt_fname)

    2
```

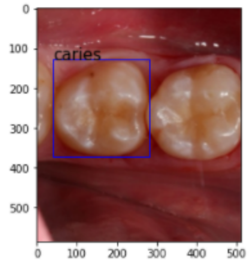
pbtxt파일의 class의 수를 출력한 결과, 2로 1: normal(정상), 2: caries(충치)로 잘 정의되어 있는 것을 확인할 수 있다.

```
[ ] # Check data
    dir_path = '/content/TensorFlow/tooth-sample-dsl'
    imgs = natsorted([i for i in os.listdir(dir_path + '/img_train') if i.startswith('Folder')])
    jsons = natsorted([i for i in os.listdir(dir_path + '/json_train') if i.startswith('Folder')])

    print(len(imgs))
    print(len(jsons))

    1734
    1734
```

1734개의 데이터도 모두 잘 불러와졌는지 개수를 통해 확인하였다.

```
[ ] # Check sample

    def show_sample_tooth(img_index):

        with open(os.path.join(dir_path, 'json_train', jsons[img_index]), 'r') as f:
            data = json.load(f)

        sample_img = cv2.imread(os.path.join(dir_path, 'img_train', imgs[img_index]))
        sample_img = cv2.cvtColor(sample_img, cv2.COLOR_BGR2RGB)

        plt.imshow(sample_img)
        fig = plt.gca()
        for i,a in enumerate(data['annotation']['regions']):
            attr = a['shape_attributes']
            label = a['region_attributes']['label']
            if label == 'normal':
              continue
            fig.add_patch(mpl.patches.Rectangle(xy = (attr['x'], attr['y']),
                                                width = attr['width'],
                                                height = attr['height'],
                                                alpha=1,
                                                color='blue',
                                                fill=None))
            fig.text(attr['x'], attr['y'],
                    label,
                    size=15)
        plt.show()
```

```
show_sample_tooth(777)
```



이미지 데이터가 잘 불러져 왔는지 실질적으로 확인하기 위한 코드로, 데이터가 잘 불러와졌고, patch를 통해 충치만을 체크하도록 표시함으로써, labeling이 충치에만 되고 있는 것인지 확인했다.

3. Download TF Pretrained Models

3.Download TF Pretrained Models

```
# Setting pre-traind-models folder & variables
!mkdir '/content/TensorFlow/tooth-sample-dsl/pre-trained-models'

CUSTOM_MODEL_NAME = 'my_ssd_mobnet'
PRETRAINED_MODEL_NAME = 'ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8'
PRETRAINED_MODEL_URL = 'http://download.tensorflow.org/models/object_detection/tf2/20200711/ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8.tar.gz'
TF_RECORD_SCRIPT_NAME = 'generate_tfrecord.py'
LABEL_MAP_NAME = 'caries_map.pbtxt'
```

```
# Setting paths of pre-trained model

paths = {
    'WORKSPACE_PATH': os.path.join(os.getcwd(),'TensorFlow', 'tooth-sample-dsl'),
    'SCRIPTS_PATH': os.path.join(os.getcwd(),'TensorFlow','scripts'),
    'APIMODEL_PATH': os.path.join(os.getcwd(),'TensorFlow','models'),
    'ANNOTATION_PATH': os.path.join(os.getcwd(),'TensorFlow', 'tooth-sample-dsl','annotations'),
    'IMAGE_TRAIN_PATH': os.path.join(os.getcwd(),'TensorFlow', 'tooth-sample-dsl','images_train'),
    'IMAGE_TEST_PATH': os.path.join(os.getcwd(),'TensorFlow', 'tooth-sample-dsl','images_test'),
    'MODEL_PATH': os.path.join(os.getcwd(),'TensorFlow', 'tooth-sample-dsl','models'),
    'PRETRAINED_MODEL_PATH': os.path.join(os.getcwd(),'TensorFlow', 'tooth-sample-dsl','pretrained-models'),
    'CHECKPOINT_PATH': os.path.join(os.getcwd(),'TensorFlow', 'tooth-sample-dsl','models',CUSTOM_MODEL_NAME),
    'OUTPUT_PATH': os.path.join(os.getcwd(),'TensorFlow', 'tooth-sample-dsl','models',CUSTOM_MODEL_NAME, 'export'),
    'TFJS_PATH':os.path.join(os.getcwd(),'TensorFlow', 'tooth-sample-dsl','models',CUSTOM_MODEL_NAME, 'tfjsexport'),
    'TFLITE_PATH':os.path.join(os.getcwd(),'TensorFlow', 'tooth-sample-dsl','models',CUSTOM_MODEL_NAME, 'tfliteexport'),
    'PROTOC_PATH':os.path.join(os.getcwd(),'TensorFlow','protoc')
 }

files = {
    'PIPELINE_CONFIG':os.path.join(os.getcwd(),'TensorFlow', 'tooth-sample-dsl','models', CUSTOM_MODEL_NAME, 'pipeline.config'),
    'TF_RECORD_SCRIPT': os.path.join(paths['SCRIPTS_PATH'], TF_RECORD_SCRIPT_NAME),
    'LABELMAP': os.path.join(paths['ANNOTATION_PATH'], LABEL_MAP_NAME)
}
```

이미 훈련되어있는 ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8를 기반으로 충치구별 모델을 만들기 위해 모델을 다운로드하고 'tooth-sample-dsl'이라는 경로를 설정함으로써 모델을 복사하기 위한 준비를 했다.

4. Copy Model Config to Training Folder

4. Copy Model Config to Training Folder

```
%%capture
%%bash


mkdir '/content/TensorFlow/tooth-sample-dsl/models'
mkdir '/content/TensorFlow/tooth-sample-dsl/models/my_ssd_mobnet'

# Copy model config from pretrained model to 'models' folder.
cp '/content/TensorFlow/tooth-sample-dsl/pre-trained-models/ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8/pipeline.config' '/content/TensorFlow/tooth-sample-dsl/models/my_ssd_mobnet'
```

모델을 훈련시키기 위해 우리의 training 폴더에 모델 config를 복사했다. (ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8에서 hyperparameter를 직접 수정하기 위해 우리파일에 복사해온 것이다.)

5. Update Config for Transfer Learning

```
pipeline_config = pipeline_pb2.TrainEvalPipelineConfig()

# Writing config file directory in custom model folders "model"
with tf.io.gfile.GFile('/content/TensorFlow/tooth-sample-dsl/models/my_ssd_mobnet/pipeline.config' , "r") as f:
    proto_str = f.read()
    text_format.Merge(proto_str, pipeline_config)
```

```
# Adjusting config file
pipeline_config.model.ssd.num_classes = get_num_classes(pbtxt_fname) # Number of classes in labelmap
pipeline_config.train_config.fine_tune_checkpoint = '/content/TensorFlow/tooth-sample-dsl/pre-trained-models/ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8/checkpoint/ckpt-0'
pipeline_config.train_config.fine_tune_checkpoint_type = "detection"
pipeline_config.train_input_reader.label_map_path= files['LABELMAP']
pipeline_config.train_input_reader.tf_record_input_reader.input_path[:] = [os.path.join(paths['ANNOTATION_PATH'], 'train.record')]
pipeline_config.eval_input_reader[0].label_map_path = files['LABELMAP']
pipeline_config.eval_input_reader[0].tf_record_input_reader.input_path[:] = [os.path.join(paths['ANNOTATION_PATH'], 'test.record')]

# Use this cells to change hyperparmeter for improving model performance
# We can adjust any hyperparameter in config file just adding command
pipeline_config.train_config.batch_size = 50 # Batch_size
pipeline_config.model.ssd.loss.classification_weight = 1.25 # Classification_weight
pipeline_config.model.ssd.loss.localization_weight = 1.25 # Localization_weight
```

Hyperparameter 수정

```
config_text = text_format.MessageToString(pipeline_config)
with tf.io.gfile.GFile(files['PIPELINE_CONFIG'], "wb") as f:
    f.write(config_text)
```

우리 training 폴더에 복사되어 있는 conifg파일에서 모델의 hyperparameter를 직접 수정하는 코드이다. 이를 훈련 모델의 config로 사용하기 위해 수정이 끝난 config파일인 pipeline_config를 PIPELINE_CONFG 파일로 보냄으로써 수정을 반영한다.

6. Train the model

6. Train the model

```
[ ] # Setting Training_script
    TRAINING_SCRIPT = os.path.join(paths['APIMODEL_PATH'], 'research', 'object_detection', 'model_main_tf2.py')
```

```
[ ] command = "python {} --model_dir={} --pipeline_config_path={} --num_train_steps=2000".format(TRAINING_SCRIPT, paths['CHECKPOINT_PATH'],files['PIPELINE_CONFIG'])
```

```
[ ] !{command}

2021-09-29 03:36:06.921043: E tensorflow/stream_executor/cuda/cuda_driver.cc:271] failed call to cuInit: CUDA_ERROR_NO_DEVICE: no CUDA-capable device is detected
2021-09-29 03:36:06.921089: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (45889216c9a2): /proc/driver/nvidia/version does not exist
WARNING:tensorflow:There are non-GPU devices in `tf.distribute.Strategy`, not using nccl allreduce.
W0929 03:36:06.925468 139662171756416 cross_device_ops.py:1387] There are non-GPU devices in `tf.distribute.Strategy`, not using nccl allreduce.
INFO:tensorflow:Using MirroredStrategy with devices ('/job:localhost/replica:0/task:0/device:CPU:0',)
I0929 03:36:06.928079 139662171756416 mirrored_strategy.py:369] Using MirroredStrategy with devices ('/job:localhost/replica:0/task:0/device:CPU:0',)
INFO:tensorflow:Maybe overwriting train_steps: 2000
I0929 03:36:06.932240 139662171756416 config_util.py:552] Maybe overwriting train_steps: 2000
INFO:tensorflow:Maybe overwriting use_bfloat16: False
I0929 03:36:06.932374 139662171756416 config_util.py:552] Maybe overwriting use_bfloat16: False
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/object_detection/model_lib_v2.py:558: StrategyBase.experimental_distribute_datasets_from_function (from tensorflow.python.distribute.distribute_lib) is deprecated and will be removed in a future
Instructions for updating:
rename to distribute_datasets_from_function
W0929 03:36:07.058054 139662171756416 deprecation.py:345] From /usr/local/lib/python3.7/dist-packages/object_detection/model_lib_v2.py:558: StrategyBase.experimental_distribute_datasets_from_function (from tensorflow.python.distribute.distribute_lib) is depr
Instructions for updating:
rename to distribute_datasets_from_function
INFO:tensorflow:Reading unweighted datasets: ['/content/TensorFlow/tooth-sample-dsl/annotations/train.record']
I0929 03:36:07.070387 139662171756416 dataset_builder.py:163] Reading unweighted datasets: ['/content/TensorFlow/tooth-sample-dsl/annotations/train.record']
INFO:tensorflow:Reading record datasets for input file: ['/content/TensorFlow/tooth-sample-dsl/annotations/train.record']
I0929 03:36:07.070605 139662171756416 dataset_builder.py:80] Reading record datasets for input file: ['/content/TensorFlow/tooth-sample-dsl/annotations/train.record']
INFO:tensorflow:Number of filenames to read: 1
I0929 03:36:07.070681 139662171756416 dataset_builder.py:81] Number of filenames to read: 1
WARNING:tensorflow:num_readers has been reduced to 1 to match input file shards.
W0929 03:36:07.070734 139662171756416 dataset_builder.py:88] num_readers has been reduced to 1 to match input file shards.
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/object_detection/builders/dataset_builder.py:105: parallel_interleave (from tensorflow.python.data.experimental.ops.interleave_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use `tf.data.Dataset.interleave(map_func, cycle_length, block_length, num_parallel_calls=tf.data.AUTOTUNE)` instead. If sloppy execution is desired, use `tf.data.Options.experimental_deterministic`.
W0929 03:36:07.083856 139662171756416 deprecation.py:345] From /usr/local/lib/python3.7/dist-packages/object_detection/builders/dataset_builder.py:105: parallel_interleave (from tensorflow.python.data.experimental.ops.interleave_ops) is deprecated and will b
Instructions for updating:
Use `tf.data.Dataset.interleave(map_func, cycle_length, block_length, num_parallel_calls=tf.data.AUTOTUNE)` instead. If sloppy execution is desired, use `tf.data.Options.experimental_deterministic`.
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/object_detection/builders/dataset_builder.py:237: DatasetV1.map_with_legacy_function (from tensorflow.python.data.ops.dataset_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use `tf.data.Dataset.map()
W0929 03:36:07.115654 139662171756416 deprecation.py:345] From /usr/local/lib/python3.7/dist-packages/object_detection/builders/dataset_builder.py:237: DatasetV1.map_with_legacy_function (from tensorflow.python.data.ops.dataset_ops) is deprecated and will be
Instructions for updating:
Use `tf.data.Dataset.map()
```

위에서 **ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8** 모델의 config를 수정하여 만든 모델을 Train 시키기 위한 코드이다. num_train_steps로 train step수를 조절하고, 우리가 수정한 config를 가져오기 위해 경로(checkpoint_path)와 config파일(PIPELINE_CONFIG)를 train command에 넣은 것이다.

7. Evaluate the Model

7. Evaluate the Model

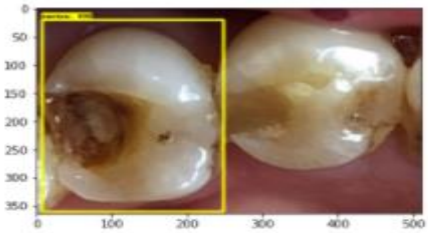
```
[ ] command = "python {} --model_dir={} --pipeline_config_path={} --checkpoint_dir={}".format(TRAINING_SCRIPT, paths['CHECKPOINT_PATH'],files['PIPELINE_CONFIG'], paths['CHECKPOINT_PATH'])
```

```
[ ] !{command}
    tensors instead of numpy arrays. It's easy to convert a tf eager tensor to
    an ndarray (just call tensor.numpy()) but having access to eager tensors
    means `tf.py_function`s can use accelerators such as GPUs as well as
    being differentiable using a gradient tape.
    - tf.numpy_function maintains the semantics of the deprecated tf.py_func
    (it is not differentiable, and manipulates numpy arrays). It drops the
    stateful argument making all functions stateful.

INFO:tensorflow:Finished eval step 100
I0929 02:56:29.584028 139686968981376 model_lib_v2.py:958] Finished eval step 100
INFO:tensorflow:Performing evaluation on 134 images.
I0929 02:56:31.391796 139686968981376 coco_evaluation.py:293] Performing evaluation on 134 images.
creating index...
index created!
INFO:tensorflow:Loading and preparing annotation results...
I0929 02:56:31.392872 139686968981376 coco_tools.py:116] Loading and preparing annotation results...
INFO:tensorflow:DONE (t=0.01s)
I0929 02:56:31.401059 139686968981376 coco_tools.py:138] DONE (t=0.01s)
creating index...
index created!
Running per image evaluation...
Evaluate annotation type *bbox*
DONE (t=0.87s).
Accumulating evaluation results...
DONE (t=0.10s).
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.702
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.848
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.822
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = -1.000
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.707
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.534
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.864
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.868
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = -1.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.868
```

Train된 모델의 성능을 확인할 수 있는 evaluation 코드이다. !{command}하단의 코드 결과 중, Average Precision (AP)중 IoU=0.5:0.95, IoU=0.5의 mAP값을 기준으로 모델의 성능을 평가했다.

8. Detect from an Image

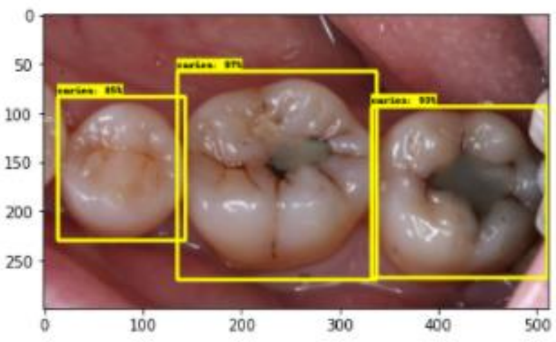
```
[ ]  # Executing inference using trained detection_model

     # Loading image
     img = cv2.imread(image_src)
     image_np = np.array(img)

     input_tensor = tf.convert_to_tensor(np.expand_dims(image_np, 0), dtype=tf.float32)
     detections = detect_fn(input_tensor)

     num_detections = int(detections.pop('num_detections'))
     detections = {key: value[0, :num_detections].numpy()
                   for key, value in detections.items()}
     detections['num_detections'] = num_detections

     # Detection_classes should be ints
     detections['detection_classes'] = detections['detection_classes'].astype(np.int64)

     label_id_offset = 1
     image_np_with_detections = image_np.copy()

     # Screening only cavity(label2 in pbtxt file)
     screening = np.where(detections['detection_classes'] < 1)
     detections['detection_boxes']= np.delete(detections['detection_boxes'],screening[0], axis = 0)
     detections['detection_classes'] = np.delete(detections['detection_classes'],screening[0])
     detections['detection_scores'] = np.delete(detections['detection_scores'],screening[0])
     detections['raw_detection_boxes'] = np.delete(detections['raw_detection_boxes'],screening[0], axis = 0)
     detections['detection_multiclass_scores'] = np.delete(detections['detection_multiclass_scores'],screening[0], axis = 0)
     detections['detection_anchor_indices'] = np.delete(detections['detection_anchor_indices'],screening[0])

     # Visualizing inference
     viz_utils.visualize_boxes_and_labels_on_image_array(
                 image_np_with_detections,
                 detections['detection_boxes'],
                 detections['detection_classes']+label_id_offset,
                 detections['detection_scores'],
                 category_index,
                 use_normalized_coordinates=True,
                 max_boxes_to_draw=5,
                 min_score_thresh=.5,
                 agnostic_mode=False)

     plt.imshow(cv2.cvtColor(image_np_with_detections, cv2.COLOR_BGR2RGB))
     plt.show()
```



새로운 충치 이미지 데이터를 불러와 훈련시킨 모델이 새로운 충치 이미지에서도 carries(충치)를 잘 구분해내는지 확인하는 코드로, 결과값으로 나온 이미지를 통해 labeling이 잘 되어있는 것을 통해 모델이 carries(충치)를 잘 구분해내고 있음을 확인할 수 있다.

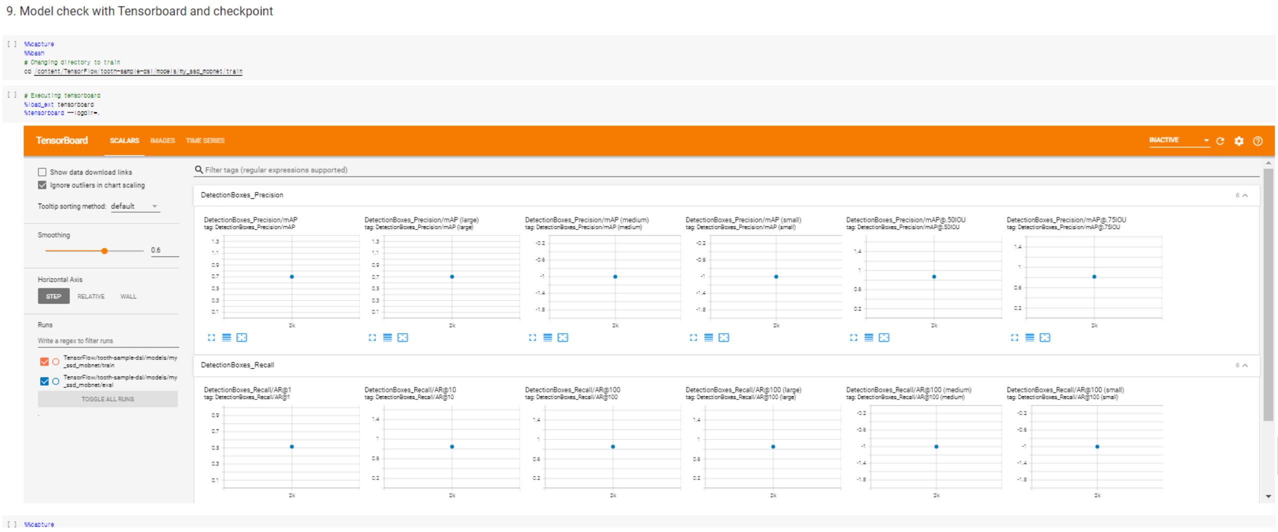
```
[ ] # Function for executing inference and saving into image file
    import imageio

    # Setting folder path
    output_folder = '/content/TensorFlow/tooth-sample-dsl/test_image_predict_result'

    def predict_and_save_result(IMAGE_PATH, min_score_thresh = 0.5):
      img = cv2.imread(IMAGE_PATH)
      image_np = np.array(img)
      input_tensor = tf.convert_to_tensor(np.expand_dims(image_np, 0), dtype=tf.float32)
      detections = detect_fn(input_tensor)

      num_detections = int(detections.pop('num_detections'))
      detections = {key: value[0, :num_detections].numpy() for key, value in detections.items()}
      detections['num_detections'] = num_detections

      # detection_classes should be ints.
      detections['detection_classes'] = detections['detection_classes'].astype(np.int64)
      label_id_offset = 1
      image_np_with_detections = image_np.copy()

      # Screening only cavity(label2 in pbtxt file)
      screening = np.where(detections['detection_classes'] < 1)
      detections['detection_boxes']= np.delete(detections['detection_boxes'],screening[0], axis = 0)
      detections['detection_classes'] = np.delete(detections['detection_classes'],screening[0])
      detections['detection_scores'] = np.delete(detections['detection_scores'],screening[0])
      detections['raw_detection_boxes'] = np.delete(detections['raw_detection_boxes'],screening[0], axis = 0)
      detections['detection_multiclass_scores'] = np.delete(detections['detection_multiclass_scores'],screening[0], axis = 0)
      detections['detection_anchor_indices'] = np.delete(detections['detection_anchor_indices'],screening[0])

      # Visualizing inference
      viz_utils.visualize_boxes_and_labels_on_image_array(
                image_np_with_detections,
                detections['detection_boxes'],
                detections['detection_classes']+label_id_offset,
                detections['detection_scores'],
                category_index,
                use_normalized_coordinates=True,
                max_boxes_to_draw=5,
                min_score_thresh=.8,
                agnostic_mode=False)

      # Saving result
      plt.imshow(cv2.cvtColor(image_np_with_detections, cv2.COLOR_BGR2RGB))
      plt.savefig(os.path.join(output_folder,'[Result]' + IMAGE_PATH.split('/')[-1]))
```

```
[ ] #  Sample image inference testing
    predict_and_save_result(IMAGE_PATH = image_src)
```



훈련시킨 모델로 충치 데이터에서 충치를 구별해낸 결과인 labeling된 이미지를 이미지 파일로 저장하기 위한 함수를 정의하는 코드이다. 우측의 sample그림을 보면 세개의 충치도 잘 구별해 내는 것을 labeling과, 그림에 표시된 accuracy를 통해 확인할 수 있다.

9. Model check with Tensorboard and checkpoint



Tensorboard의 Checkpoint를 통해 모델을 확인하였다.

10. Freezing the Graph

10. Freezing the Graph

```
[ ] FREEZE_SCRIPT = os.path.join(paths['APIMODEL_PATH'], 'research', 'object_detection', 'exporter_main_v2.py ')
```

```
[ ] command = "python {} --input_type=image_tensor --pipeline_config_path={} --trained_checkpoint_dir={} --output_directory={}".format(FREEZE_SCRIPT ,files['PIPELINE_CONFIG'], paths['CHECKPOINT_PATH'], paths['OUTPUT_PATH'])
```

```
[ ] !{command}
```

```
2021-09-27 08:06:44.223860: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:06:44.233704: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:06:44.234584: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:06:44.245056: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:06:44.245910: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:06:44.246672: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:06:44.766769: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:06:44.767678: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:06:44.768525: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:06:44.769282: 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 set. Original config value was 0.
2021-09-27 08:06:44.769360: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1510] Created device /job:localhost/replica:0/task:0/device:GPU:0 with 10819 MB memory:  -> device: 0, name: Tesla K80, pci bus id: 0000:00:04.0, compute capability: 3.7
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/python/autograph/impl/api.py:463: calling map_fn_v2 (from tensorflow.python.ops.map_fn) with back_prop=False is deprecated and will be removed in a future version.
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))
W0927 08:06:44.952352 140251897329536 deprecation.py:616] From /usr/local/lib/python3.7/dist-packages/tensorflow/python/autograph/impl/api.py:463: calling map_fn_v2 (from tensorflow.python.ops.map_fn) with back_prop=False is deprecated and will be removed in
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 0x7f8e80737b10>, because it is not built.
W0927 08:07:05.276836 140251897329536 save_impl.py:72] Skipping full serialization of Keras layer <object_detection.meta_architectures.ssd_meta_arch.SSDMetaArch object at 0x7f8e80737b10>, because it is not built.
WARNING:tensorflow:Skipping full serialization of Keras layer <keras.layers.convolutional.SeparableConv2D object at 0x7f8e805e60d0>, because it is not built.
W0927 08:07:05.570000 140251897329536 save_impl.py:72] Skipping full serialization of Keras layer <keras.layers.convolutional.SeparableConv2D object at 0x7f8e805e60d0>, because it is not built.
WARNING:tensorflow:Skipping full serialization of Keras layer <object_detection.core.freezable_batch_norm.FreezableBatchNorm object at 0x7f8e0a793890>, because it is not built.
W0927 08:07:05.570286 140251897329536 save_impl.py:72] Skipping full serialization of Keras layer <object_detection.core.freezable_batch_norm.FreezableBatchNorm object at 0x7f8e0a793890>, because it is not built.
WARNING:tensorflow:Skipping full serialization of Keras layer <keras.layers.core.Lambda object at 0x7f8e0a793c50>, because it is not built.
W0927 08:07:05.570501 140251897329536 save_impl.py:72] Skipping full serialization of Keras layer <keras.layers.core.Lambda object at 0x7f8e0a793c50>, because it is not built.
WARNING:tensorflow:Skipping full serialization of Keras layer <keras.layers.convolutional.SeparableConv2D object at 0x7f8e0a793b50>, because it is not built.
W0927 08:07:05.570681 140251897329536 save_impl.py:72] Skipping full serialization of Keras layer <keras.layers.convolutional.SeparableConv2D object at 0x7f8e0a793b50>, because it is not built.
WARNING:tensorflow:Skipping full serialization of Keras layer <object_detection.core.freezable_batch_norm.FreezableBatchNorm object at 0x7f8e0a7ffb50>, because it is not built.
W0927 08:07:05.570855 140251897329536 save_impl.py:72] Skipping full serialization of Keras layer <object_detection.core.freezable_batch_norm.FreezableBatchNorm object at 0x7f8e0a7ffb50>, because it is not built.
```

Freezing the graph를 통해 그래프와 체크포인트 변수에 대한 정보를 포함하는 단일 파일을 생성한다. 우리가 수정한 hyperparmeter를 그래프 구조 내의 상수로 저장함으로써, 반복해서 똑같은 모델의 정확도로 충치를 구분하도록 하기 위한 과정이다.

11. Conversion to TFLite

11. Conversion to TFLite

```
[ ] TFLITE_SCRIPT = os.path.join(paths['APIMODEL_PATH'], 'research', 'object_detection', 'export_tflite_graph_tf2.py ')
```

```
[ ] command = "python {} --pipeline_config_path={} --trained_checkpoint_dir={} --output_directory={}".format(TFLITE_SCRIPT ,files['PIPELINE_CONFIG'], paths['CHECKPOINT_PATH'], paths['TFLITE_PATH'])
```

```
[ ] !{command}
```

```
2021-09-27 08:20:16.330708: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:20:16.341374: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:20:16.342211: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:20:16.352904: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:20:16.353770: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:20:16.354576: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:20:16.861759: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:20:16.862649: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:20:16.863481: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:20:16.864263: 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 set. Original config value was 0.
2021-09-27 08:20:16.864339: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1510] Created device /job:localhost/replica:0/task:0/device:GPU:0 with 10819 MB memory:  -> device: 0, name: Tesla K80, pci bus id: 0000:00:04.0, compute capability: 3.7
2021-09-27 08:20:25.521687: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:20:25.522562: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:20:25.523331: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:20:25.524175: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:20:25.524919: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:20:25.525597: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1510] Created device /job:localhost/replica:0/task:0/device:GPU:0 with 10819 MB memory:  -> device: 0, name: Tesla K80, pci bus id: 0000:00:04.0, compute capability: 3.7
2021-09-27 08:20:27.907761: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:20:27.908592: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:20:27.909383: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:20:27.910196: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:20:27.910928: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:20:27.911626: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1510] Created device /job:localhost/replica:0/task:0/device:GPU:0 with 10819 MB memory:  -> device: 0, name: Tesla K80, pci bus id: 0000:00:04.0, compute capability: 3.7
WARNING:tensorflow:Skipping full serialization of Keras layer <object_detection.meta_architectures.ssd_meta_arch.SSDMetaArch object at 0x7f4f00711c90>, because it is not built.
W0927 08:20:28.853965 139979249452928 save_impl.py:72] Skipping full serialization of Keras layer <object_detection.meta_architectures.ssd_meta_arch.SSDMetaArch object at 0x7f4f00711c90>, because it is not built.
WARNING:tensorflow:Skipping full serialization of Keras layer <keras.layers.convolutional.SeparableConv2D object at 0x7f4f005c4210>, because it is not built.
W0927 08:20:29.178107 139979249452928 save_impl.py:72] Skipping full serialization of Keras layer <keras.layers.convolutional.SeparableConv2D object at 0x7f4f005c4210>, because it is not built.
WARNING:tensorflow:Skipping full serialization of Keras layer <object_detection.core.freezable_batch_norm.FreezableBatchNorm object at 0x7f4e8f74e290>, because it is not built.
W0927 08:20:29.178437 139979249452928 save_impl.py:72] Skipping full serialization of Keras layer <object_detection.core.freezable_batch_norm.FreezableBatchNorm object at 0x7f4e8f74e290>, because it is not built.
WARNING:tensorflow:Skipping full serialization of Keras layer <keras.layers.core.Lambda object at 0x7f4e8f65c1d0>, because it is not built.
W0927 08:20:29.178657 139979249452928 save_impl.py:72] Skipping full serialization of Keras layer <keras.layers.core.Lambda object at 0x7f4e8f65c1d0>, because it is not built.
WARNING:tensorflow:Skipping full serialization of Keras layer <keras.layers.convolutional.SeparableConv2D object at 0x7f4f00090c10>, because it is not built.
W0927 08:20:29.178846 139979249452928 save_impl.py:72] Skipping full serialization of Keras layer <keras.layers.convolutional.SeparableConv2D object at 0x7f4f00090c10>, because it is not built.
WARNING:tensorflow:Skipping full serialization of Keras layer <object_detection.core.freezable_batch_norm.FreezableBatchNorm object at 0x7f4f0007c250>, because it is not built.
W0927 08:20:29.179049 139979249452928 save_impl.py:72] Skipping full serialization of Keras layer <object_detection.core.freezable_batch_norm.FreezableBatchNorm object at 0x7f4f0007c250>, because it is not built.
```

```
[ ] # Setting path for TFlite conversion
    FROZEN_TFLITE_PATH = os.path.join(paths['TFLITE_PATH'], 'saved_model')
    TFLITE_MODEL = os.path.join(paths['TFLITE_PATH'], 'saved_model', 'detect.tflite')
```

```
[ ] command = "tflite_convert \
    --saved_model_dir={} \
    --output_file={} \
    --input_shapes=1,300,300,3 \
    --input_arrays=normalized_input_image_tensor \
    --output_arrays='TFLite_Detection_PostProcess','TFLite_Detection_PostProcess:1','TFLite_Detection_PostProcess:2','TFLite_Detection_PostProcess:3' \
    --inference_type=FLOAT \
    --allow_custom_ops".format(FROZEN_TFLITE_PATH, TFLITE_MODEL, )
```

```
[ ] !{command}
```

```
2021-09-27 08:22:02.979629: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1096] Device interconnect StreamExecutor with strength 1 edge matrix:
2021-09-27 08:22:02.979675: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1102]      0
2021-09-27 08:22:02.979696: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1115] 0:   N
2021-09-27 08:22:02.982019: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:981] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:22:02.982857: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:981] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:22:02.986059: 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 set. Original config value was 0.
2021-09-27 08:22:02.986168: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1241] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 10813 MB memory) -> physical GPU (device: 0, name: Tesla K80, pci bus id: 0000:00:04.0, compute 
2021-09-27 08:22:21.368910: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:981] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:22:21.369739: I tensorflow/core/grappler/devices.cc:55] Number of eligible GPUs (core count >= 8, compute capability >= 0.0): 1
2021-09-27 08:22:21.369875: I tensorflow/core/grappler/clusters/single_machine.cc:356] Starting new session
2021-09-27 08:22:21.370885: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:981] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:22:21.371679: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1555] Found device 0 with properties:
pciBusID: 0000:00:04.0 name: Tesla K80 computeCapability: 3.7
coreClock: 0.8235GHz coreCount: 13 deviceMemorySize: 11.17GiB deviceMemoryBandwidth: 223.96GiB/s
2021-09-27 08:22:21.371769: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcudart.so.10.1
2021-09-27 08:22:21.371818: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcublas.so.10
2021-09-27 08:22:21.371866: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcufft.so.10
2021-09-27 08:22:21.371915: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcurand.so.10
2021-09-27 08:22:21.371960: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcusolver.so.10
2021-09-27 08:22:21.372001: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcusparse.so.10
2021-09-27 08:22:21.372051: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcudnn.so.7
2021-09-27 08:22:21.372230: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:981] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:22:21.373094: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:981] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-09-27 08:22:21.373816: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1697] Adding visible gpu devices: 0
2021-09-27 08:22:21.373875: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1096] Device interconnect StreamExecutor with strength 1 edge matrix:
2021-09-27 08:22:21.373901: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1102]      0
```

TFLite로 변환함으로써 mobile, embedded machine에서의 모델을 사용할 수 있도록 했다.

12. Zip and Export Models

12. Zip and Export Models

```
[ ] # Export Models as tar.gz file
    !tar -czf models.tar.gz {paths['CHECKPOINT_PATH']}
```

```
tar: Removing leading `/' from member names
```

이 모델을 압축하여 밖으로 추출하는 과정이다.

3. 모델 학습 결과 및 인사이트

Train Step 과 Batch Size 조정				
Pretrained Model	Train Steps	mAP (IOU 0.5)	mAP (IOU 0.5:0.95)	수정한 parameter
ssd_mobilenet _v2_fpnlite_320_x320_coco17_tpu-8	4000	0.765	0.632	Batch_size=70 Iou_threshold=0.58 Batch_norm.decay=0.998 classification_weight=1.25 localization_weight=1.25
ssd_mobilenet _v2_fpnlite_320_x320_coco17_tpu-8	4000	0.779	0.779	Batch_size=55 Iou_threshold=0.58 Batch_norm.decay=0.998 classification_weight=1.25 localization_weight=1.25
ssd_mobilenet _v2_fpnlite_320_x320_coco17_tpu-8	2500	0.777	0.625	Batch_size=40 Iou_threshold=0.58 Batch_norm.decay=0.998 classification_weight=1.25 localization_weight=1.25
ssd_mobilenet _v2_fpnlite_320_x320_coco17_tpu-8	2500	0.728	0.556	Batch_size=20 Iou_threshold=0.58 Batch_norm.decay=0.998 classification_weight=1.25 localization_weight=1.25
Train step 과 batch size 를 변화시켜본 결과, batch size 가 40, 55 일 때 가장 높은 결과값을 보여주었다. 또한, train step 을 무조건 늘린다고 성능이 좋아지는 것이 아니라는 것을 확인할 수 있었다.				
Aspect Ratio 조정				
Pretrained Model	Train Steps	mAP (IOU 0.5)	mAP (IOU 0.5:0.95)	수정한 parameter
ssd_mobilenet _v2_fpnlite_320_x320_coco17_tpu-8	2000	0.819893	0.651016	Batch_size=20 Classification_weight=1.1 Localization_weight=1.0 Aspect_ratio=3.0 / 0.3
ssd_mobilenet _v2_fpnlite_320_x320_coco17_tpu-8	2000	0.823996	0.675252	Batch_size=50 Classification_weight=1.1 Localization_weight=1.25 Aspect_ratio=3.0 / 0.3
ssd_mobilenet _v2_fpnlite_320_x320_coco17_tpu-8	2000	0.834240	0.667254	Batch_size=75 Classification_weight=1.25 Localization_weight=1.25
ssd_mobilenet _v2_fpnlite_320_x320_coco17_tpu-8	2000	0.824226	0.628054	Batch_size=20 Classification_weight=1.1 Localization_weight=1.0 Aspect_ratio=2.85 / 0.35
ssd_mobilenet _v2_fpnlite_320_x320_coco17_tpu-8	2000	0.803816	0.656874	Batch_size=50 Classification_weight=1.1 Localization_weight=1.25 Aspect_ratio=2.85 / 0.35

ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	2000	0.881516	0.718	Batch_size=50 Classification_weight=1.25 Localization_weight=1.25 Aspect_ratio=2.85 / 0.35
최적의 aspect ratio 를 찾기 위해 다양한 조합을 실행해본 결과, aspect ratio 가 2.85 / 0.35 로 설정되었을 때 가장 좋은 결과값을 도출한다는 것을 알 수 있었다.				
Classification_weight 조정				
Pretrained Model	Train Steps	mAP (IOU 0.5)	mAP (IOU 0.5:0.95)	수정한 parameter
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	2000	0.862439	0.676	Batch_size=20 Classification_weight=1.2 Learning_rate_base=0.02 Warmup_learning_rate=0.01
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	3000	0.882	0.701	Batch_size=20 Classification_weight=1.1 Learning_rate_base=0.02 Warmup_learning_rate=0.01
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	3000	0.836158	0.673	Batch_size=20 Classification_weight=1.0 Learning_rate_base=0.02 Warmup_learning_rate=0.01
Classification weight 변수는 1.0 일 때보다 1.1 혹은 1.2 로 올렸을 때 더 좋은 결과값을 배출한다는 것을 알 수 있었다.				
Batch_size, classification_weight, localization_weight 조정				
Pretrained Model	Train Steps	mAP (IOU 0.5)	mAP (IOU 0.5:0.95)	수정한 parameter
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	2000	0.864	0.726	Batch_size=32 Classification_weight=1.25 Learning_rate_base=0.03 Localization_weight=1.25
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	2000	0.861	0.709	Batch_size=50 Classification_weight=1.25 Learning_rate_base=0.03 Localization_weight=1.25
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	2000	0.864	0.726	Batch_size=50 Classification_weight=1.15 Learning_rate_base=0.03 Localization_weight=1.15
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	2000	0.892	0.748	Batch_size=50 Classification_weight=1.2 Learning_rate_base=0.03
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	2000	0.851	0.701	Batch_size=50 Classification_weight=1.25 Learning_rate_base=0.03
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	2000	0.883	0.722	Batch_size=50 Classification_weight=1.3 Learning_rate_base=0.03
Classification weight 와 localization weight는 1.0~1.25 사이로 동시에 변경해보았을 때에는 별 다른 차이가 없었지만 Classification_weight만 1.2로 높였을 때 결과값이 크게 향상했으며 1.2 이상으로 높이자 다시 하락하는 경향을 보였다. Batch size 의 32 와 50 사이에는 큰 결과값 차이가 있지 않았다.				

Iou_threshold 조정				
Pretrained Model	Train Steps	mAP (IOU 0.5)	mAP (IOU 0.5:0.95)	수정한 parameter
ssd_mobilenet_v2_fpnlite_320_x320_coco17_tpu-8	3000	0.776	0.624	Batch_size=100 Iou_threshold=0.57 Batch_norm.decay=0.998 classification_weight=1.25 localization_weight=1.25
ssd_mobilenet_v2_fpnlite_320_x320_coco17_tpu-8	2500	0.788	0.619	Batch_size=100 Iou_threshold=0.58 Batch_norm.decay=0.998 classification_weight=1.25 localization_weight=1.25
Train step 을 500 줄였음에도 불구하고 기존 0.60 으로 설정되어있던 Non-max suppression 의 Iou_threshold 를 0.01 높였더니 MAP 값이 약 0.012 증가하여 Iou_throshold 를 낮추는 방향으로 수정해보기로 하였다.				
Iou_threshold, Batch_norm.decay 조정				
Pretrained Model	Train Steps	mAP (IOU 0.5)	mAP (IOU 0.5:0.95)	수정한 parameter
ssd_mobilenet_v2_fpnlite_320_x320_coco17_tpu-8	4000	0.793	0.648	Batch_size=20 Iou_threshold=0.58 Batch_norm.decay=0.998 classification_weight=1.25 localization_weight=1.25
ssd_mobilenet_v2_fpnlite_320_x320_coco17_tpu-8	4000	0.778	0.633	Batch_size=20 Batch_norm.decay=0.998 classification_weight=1.25 localization_weight=1.25
ssd_mobilenet_v2_fpnlite_320_x320_coco17_tpu-8	4000	0.776	0.625	Batch_size=20 Iou_threshold=0.58 classification_weight=1.25 localization_weight=1.25
Train_step, Iou_threshold 조정				
Pretrained Model	Train Steps	mAP (IOU 0.5)	mAP (IOU 0.5:0.95)	수정한 parameter
ssd_mobilenet_v2_fpnlite_320_x320_coco17_tpu-8	2000	0.892	0.748	Batch_size=50 Learning_rate_base=0.03 Classification_weight=1.2 Iou_threshold=0.6
ssd_mobilenet_v2_fpnlite_320_x320_coco17_tpu-8	3000	0.884	0.707	Batch_size=50 Learning_rate_base=0.03 Classification_weight=1.2 Iou_threshold=0.6
ssd_mobilenet_v2_fpnlite_320_x320_coco17_tpu-8	2000	0.879	0.732	Batch_size=50 Learning_rate_base=0.03 Classification_weight=1.2 Iou_threshold=0.5
ssd_mobilenet_v2_fpnlite_320_x320_coco17_tpu-8	2500	0.862	0.702	Batch_size=50 Learning_rate_base=0.03 Classification_weight=1.2 Iou_threshold=0.5

Pretrained Model	Train Steps	mAP (IOU 0.5)	mAP (IOU 0.5:0.95)	수정한 parameter
Non_max_suppresion의 iou_threshold를 0.5로 낮춰본 결과, 결과값이 떨어졌다. 또한, Train_step을 2000에서 2500과 3000으로 늘려본 결과, 미세하게 결과값이 떨어지는 것을 확인할 수 있었다. 따라서, train_step 은 2000, iou_threshold 는 처음 주어진 0.6 으로 설정해두기로 하였다.				
Batch_norm.decay 조정				
Pretrained Model	Train Steps	mAP (IOU 0.5)	mAP (IOU 0.5:0.95)	수정한 parameter
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	3000	0.787	0.663	Batch_size=50 Iou_threshold=0.58 Batch_norm.decay=0.004 classification_weight=1.2 localization_weight=1.2 learning_rate_base=0.08
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	3000	0.726	0.584	Batch_size=50 Iou_threshold=0.58 Batch_norm.decay=0.004 classification_weight=1.2 localization_weight=1.2
Batch_norm.decay 를 0.998 에서부터 0.004 까지 변화시켜본 결과, 결과값에 큰 차이를 불러일으키지 않는 것으로 보아 batch_norm.decay 는 모델 성능에 큰 영향을 끼치지 않는 것으로 판단하여 수정하지 않기로 하였다.				
Learning_rate 조정				
Pretrained Model	Train Steps	mAP (IOU 0.5)	mAP (IOU 0.5:0.95)	수정한 parameter
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	2000	0.892	0.748	Batch_size=50 Learning_rate_base=0.03 Classification_weight=1.2
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	2500	0.826	0.63	Batch_size=50 Learning_rate_base=0.01 Warmup_learning_rate=0.005 Classification_weight=1.25
최상값을 기준으로 learning rate 를 0.03 에서 0.01 로 바꿔본 결과, 결과값이 떨어진 것으로 보아, learning rate 는 0.03 으로 두는 것이 최선의 선택일 것이라고 판단했다.				
그 외 시도해본 parameter 조합과 성능				
Pretrained Model	Train Steps	mAP (IOU 0.5)	mAP (IOU 0.5:0.95)	수정한 parameter
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	2000	0.667	0.442	Batch_size=20 Batch_norm.decay=0.98
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	2000	0.762	0.578	Batch_size=45 Batch_norm.decay=0.998
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	2000	0.772	0.596	Batch_size=100 Batch_norm.decay=0.998 Iou_threshold=0.58
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	2000	0.633	0.317	Batch_size=100 Batch_norm.decay=0.998 Iou_threshold=0.58 Learning_rate_base=0.01 Warmup_learning_rate=0.005

ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	3000	0.794	0.645	Batch_size=50 Iou_threshold=0.58 Batch_norm.decay=0.998 classification_weight=1.2 localization_weight=1.2
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	3000	0.574	0.384	Batch_size=50 Iou_threshold=0.58 Batch_norm.decay=0.998 classification_weight=1.15 localization_weight=1.15
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	3000	0.753	0.602	Batch_size=50 Iou_threshold=0.58 Batch_norm.decay=0.998 classification_weight=1.25 localization_weight=1.25
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	3000	0.749	0.588	Batch_size=50 Iou_threshold=0.58 Batch_norm.decay=0.998 classification_weight=1.21 localization_weight=1.21
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	3000	0.755	0.612	Batch_size=50 Iou_threshold=0.58 Batch_norm.decay=0.998 classification_weight=1.2 localization_weight=1.2 learning_rate_base=0.03
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	3000	0.772	0.621	Batch_size=50 Iou_threshold=0.58 Batch_norm.decay=0.998 classification_weight=1.2 localization_weight=1.2 learning_rate_base=0.02
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	3000	0.735	0.596	Batch_size=50 Iou_threshold=0.58 Batch_norm.decay=0.998 classification_weight=1.2 localization_weight=1.2 learning_rate_base=0.08
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	2000	0.806	0.635	Batch_size=1
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	2000	0.864	0.730	Warmup_steps=1000 Batch_size=50
ssd_mobilenet _v2_fpnlite_64 0_x640_coco17_tpu-8	2500	0.806	0.644	Batch_size=50 Learning_rate_base=0.03 Classification_weight=1.2
ssd_mobilenet _v2_fpnlite_64 0_x640_coco17_tpu-8	2500	0.808	0.652	Batch_size=32 Learning_rate_base=0.03 Classification_weight=1.2

ssd_mobilenet_v2_fpnlite_640_x640_coco17_tpu-8	2500	0.830	0.645	Batch_size=32 Learning_rate_base=0.01 Warmup_learning_rate=0.005 Classification_weight=1.2
ssd_mobilenet_v2_fpnlite_640_x640_coco17_tpu-8	2500	0.879	0.707	Batch_size=75 Learning_rate_base=0.01 Warmup_learning_rate=0.005 Classification_weight=1.2
ssd_mobilenet_v2_fpnlite_320_x320_coco17_tpu-8	1500	0.775	0.229	Batch_size=30 Classification_weight=1.3
ssd_mobilenet_v2_fpnlite_320_x320_coco17_tpu-8	2000	-	0.651338	Aspect_scale=3.0
ssd_mobilenet_v2_fpnlite_320_x320_coco17_tpu-8	2000	-	0.696061	Batch_size=80
ssd_mobilenet_v2_fpnlite_320_x320_coco17_tpu-8	2000	-	0.707331	Learning_rate=0.025
ssd_mobilenet_v2_fpnlite_320_x320_coco17_tpu-8	3000	0.882	0.701	Batch_size=20 Classification_weight=1.1 Localization_weight=1.0 Learning_rate_base=0.02 Warmup_learning_rate=0.01
ssd_mobilenet_v2_fpnlite_320_x320_coco17_tpu-8	3000	0.860149	0.701	Batch_size=50 Classification_weight=1.1 Localization_weight=1.25
ssd_mobilenet_v2_fpnlite_320_x320_coco17_tpu-8	3000	0.8797210	0.710371	Batch_size=50 Classification_weight=1.25 Localization_weight=1.25
ssd_mobilenet_v2_fpnlite_320_x320_coco17_tpu-8	3000	0.873	0.697	Batch_size=32 Classification_weight=1.2
ssd_mobilenet_v2_fpnlite_320_x320_coco17_tpu-8	3000	0.808	0.602	Batch_size=32 Classification_weight=1.2 Learning_rate_base=0.04
ssd_mobilenet_v2_fpnlite_320_x320_coco17_tpu-8	3000	0.833	0.690	Batch_size: 32 Classification_weight=1.2 Learning_rate_base=0.095
ssd_mobilenet_v2_fpnlite_320_x320_coco17_tpu-8	3000	0.844	0.655	Batch_size=32 Classification_weight=1.2 Localization_weight=1.1
ssd_mobilenet_v2_fpnlite_320_x320_coco17_tpu-8	3000	0.864	0.712	Batch_size=32 Classification_weight=1.2 Localization_weight=1.2
ssd_mobilenet_v2_fpnlite_320_x320_coco17_tpu-8	3000	0.840	0.685	Batch_size=32 Classification_weight=1.2 Localization_weight=1.25

ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	3000	0.901	0.708	Batch_size=32 Classification_weight=1.25 Localization_weight=1.25
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	3000	0.882	0.681	Batch_size=32 Classification_weight=1.28 Localization_weight=1.28
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	3000	0.846	0.684	Batch_size=32 Classification_weight=1.25 Localization_weight=1.25 batch_norm_epsilon=0.002
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	3000	0.841	0.678	Batch_size=32 Classification_weight=1.25 Localization_weight=1.25 batch_norm_epsilon=0.01
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	3000	0.839	0.660	Batch_size=32 Classification_weight=1.24 Localization_weight=1.26 batch_norm_epsilon=0.01
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	3000	0.847	0.696	Batch_size=32 Classification_weight=1.25 Localization_weight=1.25 Learning_rate_base=0.05
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	3000	0.841	0.674	Batch_size=32 Classification_weight=1.3 Localization_weight=1.3 Learning_rate_base=0.05
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	3500	0.836	0.679	Batch_size=50 Classification_weight=1.5 Localization_weight=1.5
ssd_mobilenet _v2_fpnlite_32 0_x320_coco17_tpu-8	3500	0.810	0.642	Batch_size=50 Classification_weight=1.25 Localization_weight=1.25
ssd_resnet50 _v1_fpn_64 0_x640_coco17_tpu-8	3500	0.886	0.723	Batch_size=50 Classification_weight=1.2 Localization_weight=1.2