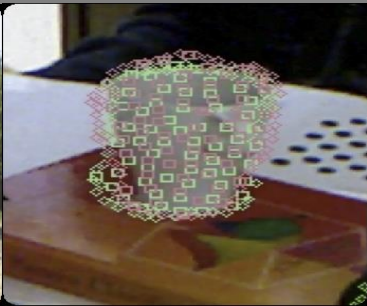
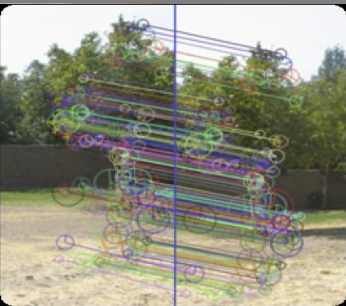


Deep Learning Basics

(#xx: Keras-based Convolutional Neural Network Practice-Part3)

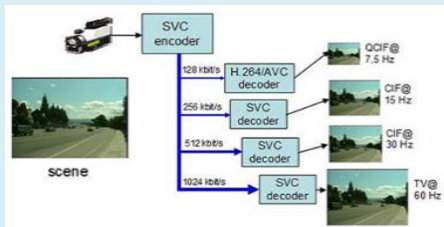
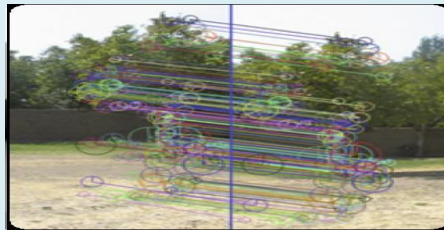


2023 Autumn

Prof. Byung-Gyu Kim
Intelligent Vision Processing Lab. (IVPL)
<http://ivpl.sookmyung.ac.kr>
Dept. of IT Engineering, Sookmyung Women's University
E-mail: bg.kim@ivpl.sookmyung.ac.kr

Gaal of this lecture

- ❖ Understand structure and how to set Google Tensorflow-based Objection Detection API
 - How to set the environment for Google Tensorflow API?
 - How to run object detection of Google Tensorflow API?
 - Run the object detection API using Webcam (live)?



Contents

- What is Google Object Detection API?

Google Objection Detection API (1)

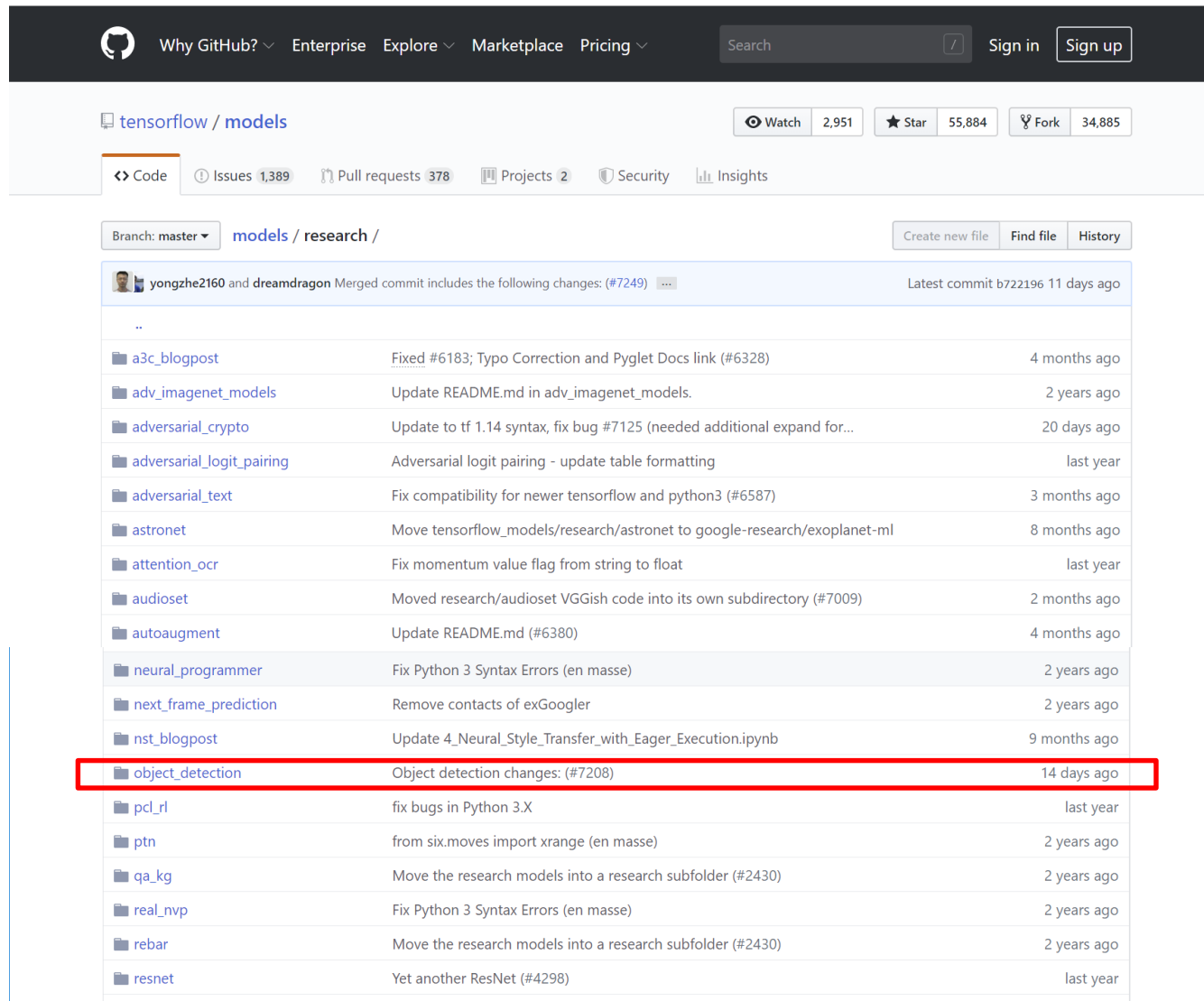
- ❖ Creating accurate machine learning models capable of localizing and identifying multiple objects in a single image remains a core challenge in computer vision.
 - Based on Tensorflow framework (open source framework)
 - <https://github.com/tensorflow/models/tree/master/research>



Google Objection Detection API (1)

❖ Getting the source

- <https://github.com/tensorflow/models/tree/master/research>

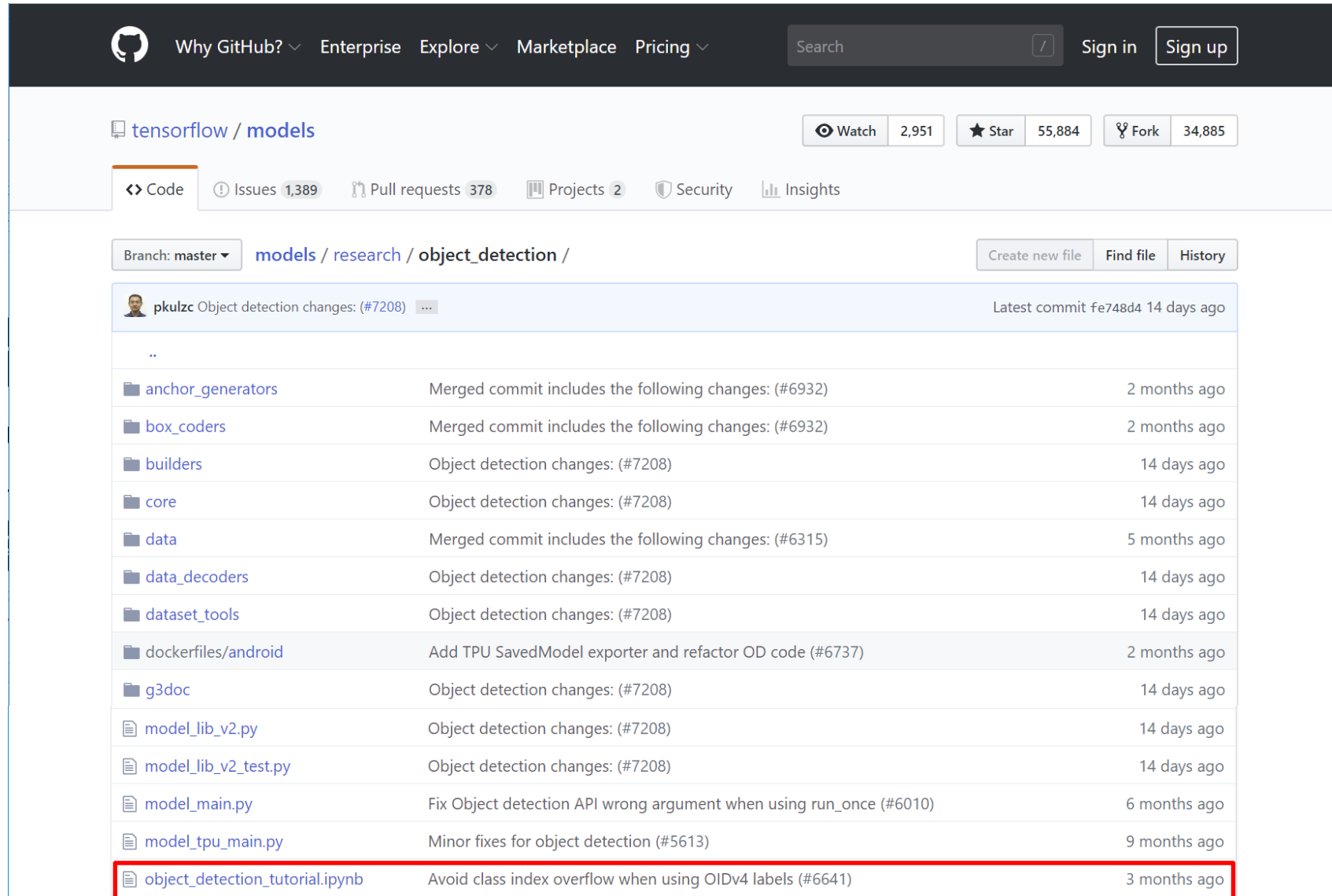


The screenshot shows the GitHub repository page for tensorflow/models. The repository is located at tensorflow/models and has 2,951 watchers, 55,884 stars, and 34,885 forks. The current branch is master, and the directory being viewed is models/research. The latest commit is b722196, made 11 days ago. A table of recent commits is displayed, with the 'object_detection' directory highlighted in red. The table lists the following commits:

Commit	Description	Time
..		
a3c_blogpost	Fixed #6183; Typo Correction and Pyglet Docs link (#6328)	4 months ago
adv_imagenet_models	Update README.md in adv_imagenet_models.	2 years ago
adversarial_crypto	Update to tf 1.14 syntax, fix bug #7125 (needed additional expand for...	20 days ago
adversarial_logit_pairing	Adversarial logit pairing - update table formatting	last year
adversarial_text	Fix compatibility for newer tensorflow and python3 (#6587)	3 months ago
astronet	Move tensorflow_models/research/astronet to google-research/exoplanet-ml	8 months ago
attention_ocr	Fix momentum value flag from string to float	last year
audioset	Moved research/audioset VGGish code into its own subdirectory (#7009)	2 months ago
autoaugment	Update README.md (#6380)	4 months ago
neural_programmer	Fix Python 3 Syntax Errors (en masse)	2 years ago
next_frame_prediction	Remove contacts of exGoogle	2 years ago
nst_blogpost	Update 4_Neural_Style_Transfer_with_Eager_Execution.ipynb	9 months ago
object_detection	Object detection changes: (#7208)	14 days ago
pcl_rl	fix bugs in Python 3.X	last year
ptn	from six.moves import xrange (en masse)	2 years ago
qa_kg	Move the research models into a research subfolder (#2430)	2 years ago
real_nvq	Fix Python 3 Syntax Errors (en masse)	2 years ago
rebar	Move the research models into a research subfolder (#2430)	2 years ago
resnet	Yet another ResNet (#4298)	last year

Google Objection Detection API (2)

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



The screenshot shows the GitHub repository page for tensorflow/models. The repository is currently on the master branch. The file tree for the research/object_detection directory is displayed, showing various subdirectories and files. The file object_detection_tutorial.ipynb is highlighted with a red box.

File/Directory	Description	Time
..		
anchor_generators	Merged commit includes the following changes: (#6932)	2 months ago
box_coders	Merged commit includes the following changes: (#6932)	2 months ago
builders	Object detection changes: (#7208)	14 days ago
core	Object detection changes: (#7208)	14 days ago
data	Merged commit includes the following changes: (#6315)	5 months ago
data_decoders	Object detection changes: (#7208)	14 days ago
dataset_tools	Object detection changes: (#7208)	14 days ago
dockerfiles/android	Add TPU SavedModel exporter and refactor OD code (#6737)	2 months ago
g3doc	Object detection changes: (#7208)	14 days ago
model_lib_v2.py	Object detection changes: (#7208)	14 days ago
model_lib_v2_test.py	Object detection changes: (#7208)	14 days ago
model_main.py	Fix Object detection API wrong argument when using run_once (#6010)	6 months ago
model_tpu_main.py	Minor fixes for object detection (#5613)	9 months ago
object_detection_tutorial.ipynb	Avoid class index overflow when using OIDv4 labels (#6641)	3 months ago

Google Object Detection API (3): Configurations

❖ My folder (after copying)

vicl > practices > TesorflowAPI > research > object_detection

이름	수정한 날짜	유형	크기
models	2019-07-29 오후...	파일 폴더	
protos	2019-07-29 오후...	파일 폴더	
samples	2019-07-29 오후...	파일 폴더	
ssd_mobilenet_v1_coco_2017_11_17	2019-07-29 오후...	파일 폴더	
test_ckpt	2019-07-29 오후...	파일 폴더	
test_data	2019-07-29 오후...	파일 폴더	
test_images	2019-07-29 오후...	파일 폴더	
utils	2019-07-29 오후...	파일 폴더	
init.py	2018-04-25 오후...	Python File	0KB
CONTRIBUTING.md	2018-04-25 오후...	MD 파일	1KB
eval.py	2018-04-25 오후...	Python File	6KB
eval_util.py	2018-04-25 오후...	Python File	30KB
eval_util_test.py	2018-04-25 오후...	Python File	5KB
evaluator.py	2018-04-25 오후...	Python File	12KB
export_inference_graph.py	2018-04-25 오후...	Python File	7KB
exporter.py	2018-04-25 오후...	Python File	18KB
exporter_test.py	2018-04-25 오후...	Python File	41KB
inputs.py	2018-04-25 오후...	Python File	19KB
inputs_test.py	2018-04-25 오후...	Python File	24KB
Manual-20181011	2018-11-01 오후...	텍스트 문서	1KB
model_hparams.py	2018-04-25 오후...	Python File	2KB
model_lib.py	2018-04-25 오후...	Python File	28KB
model_lib_test.py	2018-04-25 오후...	Python File	17KB
model_main.py	2018-04-25 오후...	Python File	3KB
model_tpu_main.py	2018-04-25 오후...	Python File	5KB
object_detection_streaming.py	2018-10-18 오후...	Python File	9KB
object_detection_streaming_room_102.py	2018-11-02 오후...	Python File	9KB
object_detection_tutorial.ipynb	2018-04-26 오후...	IPYNB 파일	13KB
object_detection_tutorial.py	2018-04-26 오후...	Python File	8KB
object_detection_tutorial_video.py	2018-05-10 오후...	Python File	8KB
object_detection_tutorial_video_rev.py	2018-10-11 오후...	Python File	8KB
object_detections_GPU-then-CPU.ipynb	2018-05-10 오후...	IPYNB 파일	8KB
object_detections_GPU-then-CPU.py	2018-10-11 오후...	Python File	6KB
object_detections_GPU-then-CPU_Final.py	2018-10-11 오후...	Python File	6KB
object_detections_GPU-then-CPU_Final2.py	2019-07-29 오후...	Python File	7KB
object_detections_GPU-then-CPU_rev.py	2018-05-10 오후...	Python File	6KB

Google Objection Detection API (4): Configurations

❖ Install cv2 (open CV module) package

```
(BGKim) C:\Users\Wvicl>pip install opencv-python
```

```
(BGKim) C:\Users\Wvicl>pip install opencv-python
Collecting opencv-python
  Downloading https://files.pythonhosted.org/packages/a3/50/04d0669afe884f137c2f490642756e8c4a658254300a9eaf253d1e643085/opencv_python-4.1.0.25-cp37-cp37m-win_amd64.whl (37.3MB)
    |████████████████████████████████████████| 37.4MB 3.3MB/s
Requirement already satisfied: numpy>=1.14.5 in c:\programdata\anaconda3\envs\Wbgkim\lib\site-packages (from opencv-python) (1.16.4)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.1.0.25

(BGKim) C:\Users\Wvicl>
```

Google (object detection) API Installation:

https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/installation.md

Google Object Detection API (5): Run API with Cam..!!!

- ❖ Go to the "object_dection" folder and run "object_detection_tutorial.py"

```
(BGKim) C:\Users#vivi\practices#TensorflowAPI#research#object_detection>python object_detections_GPU-then-CPU_Final2.py
width = 30, height=30 300 720
WARNING: Logging before flag parsing goes to stderr.
W0729 18:12:05.312280 26652 deprecation_wrapper.py:119] From object_detections_GPU-then-CPU_Final2.py:55: The name tf.Session is deprecated. Please use tf.compat.v1.Session instead.

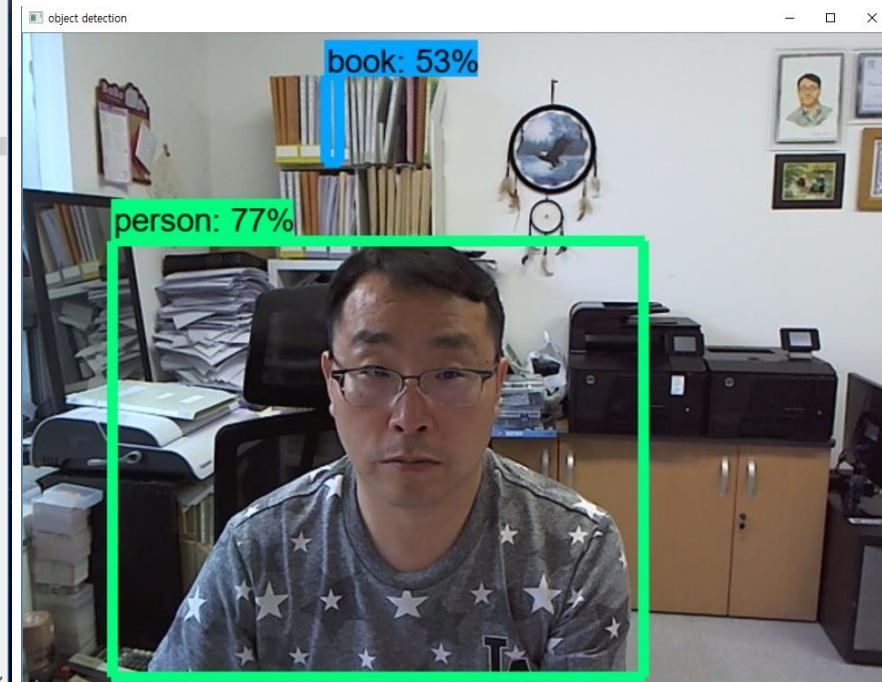
2019-07-29 18:12:05.314197: I tensorflow/stream_executor/platform/default/dso_loader.cc:42] Successfully opened dynamic library nvcuda.dll
2019-07-29 18:12:05.422849: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1640] Found device 0 with properties:
name: GeForce GTX 1070 Ti major: 6 minor: 1 memoryClockRate(GHz): 1.683
pciBusID: 0000:01:00.0
2019-07-29 18:12:05.430335: I tensorflow/stream_executor/platform/default/dlopen_checker_stub.cc:25] GPU libraries are statically linked, skip dlopen check.
2019-07-29 18:12:05.438956: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1763] Adding visible gpu devices: 0
2019-07-29 18:12:05.442641: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2
2019-07-29 18:12:05.450515: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1640] Found device 0 with properties:
name: GeForce GTX 1070 Ti major: 6 minor: 1 memoryClockRate(GHz): 1.683
pciBusID: 0000:01:00.0
2019-07-29 18:12:05.456573: I tensorflow/stream_executor/platform/default/dlopen_checker_stub.cc:25] GPU libraries are statically linked, skip dlopen check.
2019-07-29 18:12:05.465620: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1763] Adding visible gpu devices: 0
2019-07-29 18:12:05.939247: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1181] Device interconnect StreamExecutor with strength 1 edge matrix:
2019-07-29 18:12:05.943823: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1187] 0
2019-07-29 18:12:05.946833: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1200] 0: N
2019-07-29 18:12:05.954891: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1326] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 6357 MB memory) -> physical GPU (device: 0, name: GeForce GTX 1070 Ti, pci bus id: 0000:01:00.0, compute capability: 6.1)
W0729 18:12:05.964537 26652 deprecation_wrapper.py:119] From object_detections_GPU-then-CPU_Final2.py:56: The name tf.placeholder is deprecated. Please use tf.compat.v1.placeholder instead.

W0729 18:12:05.966533 26652 deprecation_wrapper.py:119] From object_detections_GPU-then-CPU_Final2.py:67: The name tf.GraphDef is deprecated. Please use tf.compat.v1.GraphDef instead.

W0729 18:12:05.966533 26652 deprecation_wrapper.py:119] From object_detections_GPU-then-CPU_Final2.py:68: The name tf.gfile.GFile is deprecated. Please use tf.io.gfile.GFile instead.

W0729 18:12:07.006775 26652 deprecation_wrapper.py:119] From object_detections_GPU-then-CPU_Final2.py:153: The name tf.ConfigProto is deprecated. Please use tf.compat.v1.ConfigProto instead.

2019-07-29 18:12:07.010673: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1640] Found device 0 with properties:
name: GeForce GTX 1070 Ti major: 6 minor: 1 memoryClockRate(GHz): 1.683
pciBusID: 0000:01:00.0
2019-07-29 18:12:07.017149: I tensorflow/stream_executor/platform/default/dlopen_checker_stub.cc:25] GPU libraries are statically linked, skip dlopen check.
2019-07-29 18:12:07.026666: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1763] Adding visible gpu devices: 0
2019-07-29 18:12:07.031536: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1181] Device interconnect StreamExecutor with strength 1 edge matrix:
2019-07-29 18:12:07.036820: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1187] 0
2019-07-29 18:12:07.041286: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1200] 0: N
2019-07-29 18:12:07.048895: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1326] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 6357 MB memory) -> physical GPU (device: 0, name: GeForce GTX 1070 Ti, pci bus id: 0000:01:00.0, compute capability: 6.1)
```



Google Objection Detection API (6): Algorithms

■ Source Insight

```
import numpy as np
import os
import sys

# This is needed since the notebook is stored in
# the object_detection folder.
sys.path.append("../")

import tensorflow as tf
import time
import copy

from tensorflow.core.framework import graph_pb2
from utils import label_map_util
from utils import visualization_utils as vis_util
from matplotlib import pyplot as plt
from PIL import Image

import cv2
cap = cv2.VideoCapture(0)
    (continue)
```

```
def _node_name(n):
    if n.startswith("^"):
        return n[1:]
    else:
        return n.split(":")[0]

input_graph = tf.Graph()
with tf.Session(graph=input_graph):
    score = tf.placeholder(tf.float32, shape=(None, 1917,
90), name="Postprocessor/convert_scores")
    expand = tf.placeholder(tf.float32, shape=(None, 1917,
1, 4), name="Postprocessor/ExpandDims_1")
    for node in input_graph.as_graph_def().node:
        if node.name == "Postprocessor/convert_scores":
            score_def = node
        if node.name == "Postprocessor/ExpandDims_1":
            expand_def = node

    (continue)
```

Google Objection Detection API (7): Algorithms

```
detection_graph = tf.Graph()
with detection_graph.as_default():
    od_graph_def = tf.GraphDef()
    with tf.gfile.GFile('./ssd_mobilenet_v1_coco_2017_11_17/frozen_inference_graph.pb', 'rb') as fid:
        serialized_graph = fid.read()
        od_graph_def.ParseFromString(serialized_graph)
        dest_nodes = ['Postprocessor/convert_scores', 'Postprocessor/ExpandDims_1']

    edges = {}
    name_to_node_map = {}
    node_seq = {}
    seq = 0
    for node in od_graph_def.node:
        n = _node_name(node.name)
        name_to_node_map[n] = node
        edges[n] = [_node_name(x) for x in node.input]
        node_seq[n] = seq
        seq += 1

    for d in dest_nodes:
        assert d in name_to_node_map, "%s is not in graph" % d

    nodes_to_keep = set()
    next_to_visit = dest_nodes[:]
    (continue)
```

Google Objection Detection API (8): Algorithms

```
while next_to_visit:
    n = next_to_visit[0]
    del next_to_visit[0]
    if n in nodes_to_keep:
        continue
    nodes_to_keep.add(n)
    next_to_visit += edges[n]

nodes_to_keep_list = sorted(list(nodes_to_keep), key=lambda n: node_seq[n])

nodes_to_remove = set()
for n in node_seq:
    if n in nodes_to_keep_list: continue
    nodes_to_remove.add(n)
nodes_to_remove_list = sorted(list(nodes_to_remove), key=lambda n: node_seq[n])

keep = graph_pb2.GraphDef()
for n in nodes_to_keep_list:
    keep.node.extend([copy.deepcopy(name_to_node_map[n])])

remove = graph_pb2.GraphDef()
remove.node.extend([score_def])
remove.node.extend([expand_def])
for n in nodes_to_remove_list:
    remove.node.extend([copy.deepcopy(name_to_node_map[n])]) # continue
```

Google Objection Detection API (9): Algorithms

```
with tf.device('/gpu:0'):
    tf.import_graph_def(keep, name='')
with tf.device('/cpu:0'):
    tf.import_graph_def(remove, name='')
```

```
PATH_TO_LABELS = os.path.join('C:/tensorflow/models/research/object_detection/data',
                               'mscoco_label_map.pbtxt')
```

```
NUM_CLASSES = 90
```

```
label_map = label_map_util.load_labelmap(os.path.join(PATH_TO_LABELS))
```

```
#label_map = label_map_util.load_labelmap('data/mscoco_label_map.pbtxt')
```

```
categories = label_map_util.convert_label_map_to_categories(label_map, max_num_classes=NUM_CLASSES,
                                                            use_display_name=True)
```

```
category_index = label_map_util.create_category_index(categories)
```

```
def load_image_into_numpy_array(image):
```

```
    (im_width, im_height) = image.size
```

```
    return np.array(image.getdata()).reshape((im_height, im_width, 3)).astype(np.uint8)
```

```
        (continue)
```

Google Objection Detection API (10): Algorithms

```
# For the sake of simplicity we will use only 2 images:
# image1.jpg
# image2.jpg
# If you want to test the code with your images, just add path to the images to the TEST_IMAGE_PATHS.
PATH_TO_TEST_IMAGES_DIR = 'C:/tensorflow/models/research/object_detection/test_images'
TEST_IMAGE_PATHS = [ os.path.join(PATH_TO_TEST_IMAGES_DIR, 'image{}.jpg'.format(i)) for i in range(1, 3) ]

# Size, in inches, of the output images.
IMAGE_SIZE = (12, 8)

# In[6]:

with detection_graph.as_default():
    with tf.Session(graph=detection_graph, config=tf.ConfigProto(allow_soft_placement=True)) as sess:
        image_tensor = detection_graph.get_tensor_by_name('image_tensor:0')
        score_out = detection_graph.get_tensor_by_name('Postprocessor/convert_scores:0')
        expand_out = detection_graph.get_tensor_by_name('Postprocessor/ExpandDims_1:0')
        score_in = detection_graph.get_tensor_by_name('Postprocessor/convert_scores_1:0')
        expand_in = detection_graph.get_tensor_by_name('Postprocessor/ExpandDims_1_1:0')
        detection_boxes = detection_graph.get_tensor_by_name('detection_boxes:0')
        detection_scores = detection_graph.get_tensor_by_name('detection_scores:0')
        detection_classes = detection_graph.get_tensor_by_name('detection_classes:0')
        num_detections = detection_graph.get_tensor_by_name('num_detections:0')
        i = 0
```

Google Objection Detection API (10): Algorithms

```
while True:          # Webcam case
#for _ in range(10): # Image case
    #image_path = TEST_IMAGE_PATHS[1]
    #i += 1
    #image = Image.open(image_path)
    #image_np = load_image_into_numpy_array(image)
    ret, image_np = cap.read()
    image_np_expanded = np.expand_dims(image_np, axis=0)

    start_time = time.time()
    (score, expand) = sess.run([score_out, expand_out], feed_dict={image_tensor: image_np_expanded})
    (boxes, scores, classes, num) = sess.run(
        [detection_boxes, detection_scores, detection_classes, num_detections],
        feed_dict={score_in:score, expand_in: expand})
    #print 'Iteration %d: %.3f sec'%(i, time.time()-start_time)

    vis_util.visualize_boxes_and_labels_on_image_array(image_np, np.squeeze(boxes),
    np.squeeze(classes).astype(np.int32),
    np.squeeze(scores),
    category_index,
    use_normalized_coordinates=True,
    line_thickness=8)
        (continue)
```

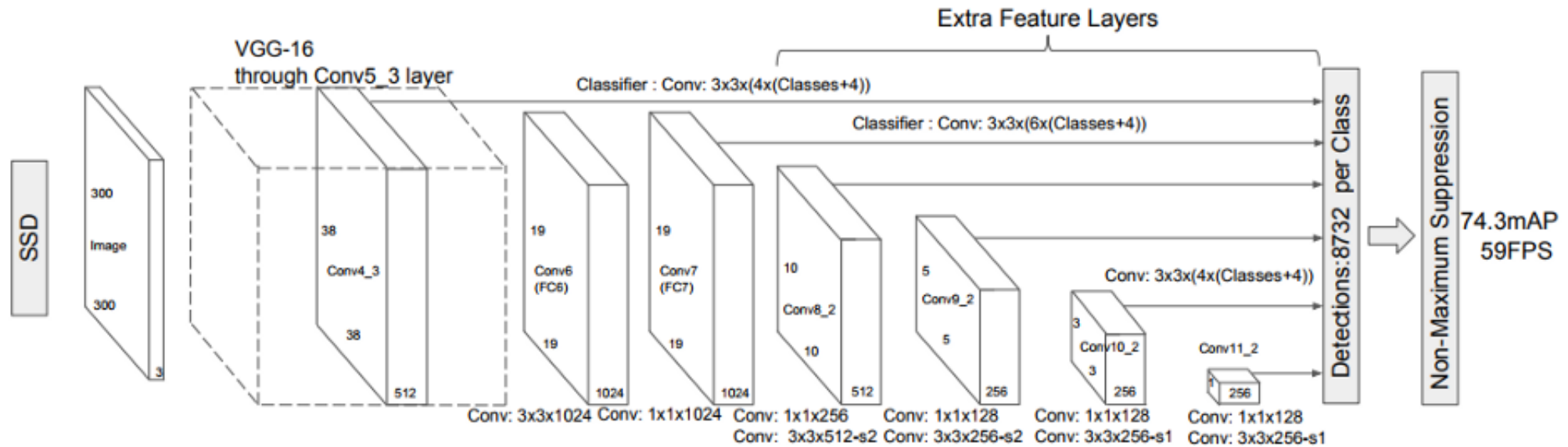
Google Objection Detection API (11): Algorithms

```
        (continue)
#cv2.imshow('object detection', cv2.resize(image_np, (800,600)))
cv2.imshow('object detection', cv2.resize(image_np, (width,height)))
#cv2.imshow('object detection', cv2.resize(image_np, (1080, 720)))
if cv2.waitKey(25) & 0xff == ord('q'):
    cv2.destroyAllWindows()
    break

# plt.figure(figsize=IMAGE_SIZE)
# plt.imshow(image_np)
```


Google Objection Detection API (12): Algorithms

- ❖ Network Model: SSD (Single-Shot Multibox Detector ([SSD](#))) (VGG-16)



Google Objection Detection API (13): Main Procedure of API

❖ Tensorflow Graph():

```
input_graph = tf.Graph() ← A TensorFlow computation, represented as a dataflow graph.  
with tf.Session(graph=input_graph):
```

❖ GraphDef():

```
od_graph_def = tf.GraphDef() ← Create TensorFlow protocol buffer .
```

❖ Tf.placeholder():

- Inserts a placeholder for a tensor that will be always fed.

```
score = tf.placeholder(tf.float32, shape=(None, 1917, 90),  
name="Postprocessor/convert_scores")
```

```
x = tf.compat.v1.placeholder(tf.float32, shape=(1024, 1024))  
y = tf.matmul(x, x)
```

```
with tf.compat.v1.Session() as sess:  
    print(sess.run(y)) # ERROR: will fail because x was not fed.
```

```
rand_array = np.random.rand(1024, 1024)  
print(sess.run(y, feed_dict={x: rand_array})) # Will succeed.
```

Google Objection Detection API (14): Main Procedure of API

❖ Tensorflow Session():

```
with tf.Session(graph=input_graph): ← A class for running TensorFlow operations.
```

❖ tf.gfile.GFile():

- Load or save file data in Python.

```
with tf.gfile.GFile('./ssd_mobilenet_v1_coco_2017_11_17/frozen_inference_graph.pb', 'rb') as fid:
```

Python example

```
with gfile.GFile(filepath, "w") as file:  
    file.write(result_str)
```

❖ with tf.device('/gpu:0'):

- Supporting devices (CPU or GPU) manually.

```
# Creates a graph.  
with tf.device('/cpu:0'):  
    a = tf.constant([1.0, 2.0, 3.0, 4.0, 5.0, 6.0], shape=[2, 3], name='a')  
    b = tf.constant([1.0, 2.0, 3.0, 4.0, 5.0, 6.0], shape=[3, 2], name='b')  
    c = tf.matmul(a, b)  
# Creates a session with log_device_placement set to True.  
sess = tf.Session(config=tf.ConfigProto(log_device_placement=True))  
# Runs the op.  
print(sess.run(c))
```



```
Device mapping:  
/job:localhost/replica:0/task:0/device:GPU:0 -> device: 0, name: Tesla K40c, pci bus  
id: 0000:05:00.0  
b: /job:localhost/replica:0/task:0/cpu:0  
a: /job:localhost/replica:0/task:0/cpu:0  
MatMul: /job:localhost/replica:0/task:0/device:GPU:0  
[[ 22. 28.]  
 [ 49. 64.]
```

Google Objection Detection API (15): Main Procedure of API

- ❖ `PATH_TO_LABELS =`
`os.path.join('C:/tensorflow/models/research/object_detection/data',`
`'mscoco_label_map.pbtxt')`
 - Set the path and file location for Label data.
- ❖ `sess.run(args)`
 - Execute the tensorflow with the predefined parameters and environment.

```
import tensorflow as tf

x = tf.placeholder(tf.float32, shape=[3])
y = tf.square(x)

with tf.Session() as sess:
    # 텐서 y를 계산하려면 필요한 플레이스홀더 이름 x와
    # 플레이스홀더에서 사용할 값을 딕셔너리 형태로 입력해줍니다.
    print(sess.run(y, {x: [1.0, 2.0, 3.0]}))
    print(sess.run(y, {x: [4.0, 5.0, 6.0]}))
```

Google Object Detection API (16): Models

❖ COCO-pre trained models

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
faster_rcnn_resnet50_lowproposals_coco	64		Boxes
rfcn_resnet101_coco	92	30	Boxes
faster_rcnn_resnet101_coco	106	32	Boxes
faster_rcnn_resnet101_lowproposals_coco	82		Boxes
faster_rcnn_inception_resnet_v2_atrous_coco	620	37	Boxes
faster_rcnn_inception_resnet_v2_atrous_lowproposals_coco	241		Boxes
faster_rcnn_nas	1833	43	Boxes
faster_rcnn_nas_lowproposals_coco	540		Boxes
mask_rcnn_inception_resnet_v2_atrous_coco	771	36	Masks
mask_rcnn_inception_v2_coco	79	25	Masks
mask_rcnn_resnet101_atrous_coco	470	33	Masks
mask_rcnn_resnet50_atrous_coco	343	29	Masks

```
# What model to download.  
MODEL_NAME = 'mask_rcnn_inception_v2_coco_2018_01_28'  
MODEL_FILE = MODEL_NAME + '.tar.gz'  
DOWNLOAD_BASE =  
'http://download.tensorflow.org/models/object_detection/'
```

```
# Path to frozen detection graph. This is the actual model that  
is used for the object detection.  
PATH_TO_CKPT = MODEL_NAME + '/frozen_inference_graph.pb'
```

Change Model that you want to use...!!!

Thank you for your attention!!!
QnA

<http://ivpl.sookmyung.ac.kr>