SOSCON

YOLO? YOLK!

Keras Korea K | 박정현

SOSCON2019
SAMSUNG OPEN SOURCE CONFERENCE 2019

whoami

- 싱 엔지니어
- 중
-
- 트와이스 좋아합니다

Linkedin : https://www.linkedin.com/in/junghyun-park-733991b1/
Facebook : https://www.facebook.com/Junghyun1221
Github : https://github.com/parkjh688

YOLO?

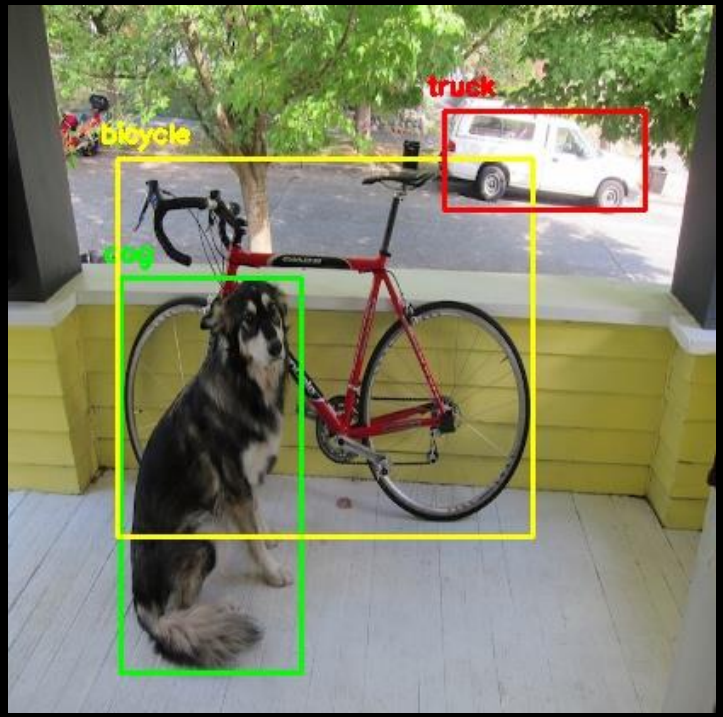
Object Detection?

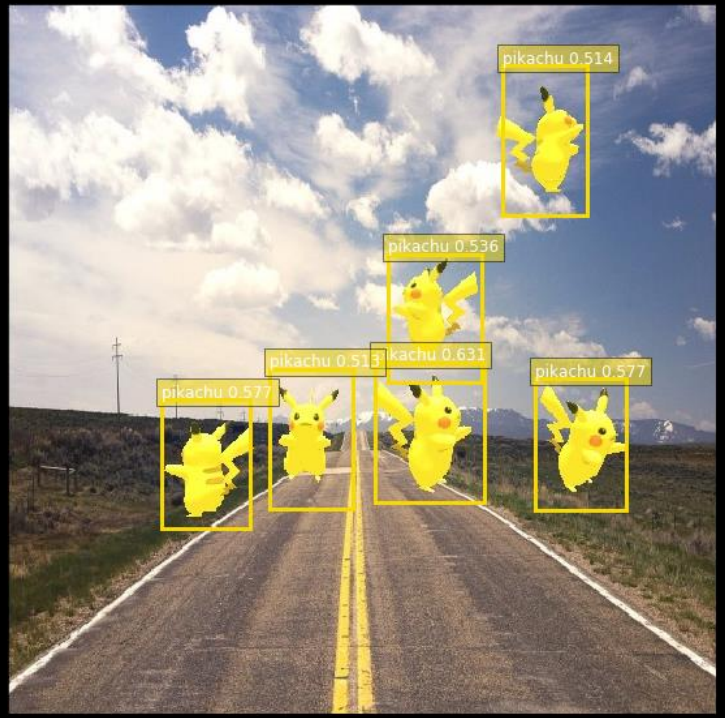
Classification

고양이?

Classification + Localization

어디있는 고양이?



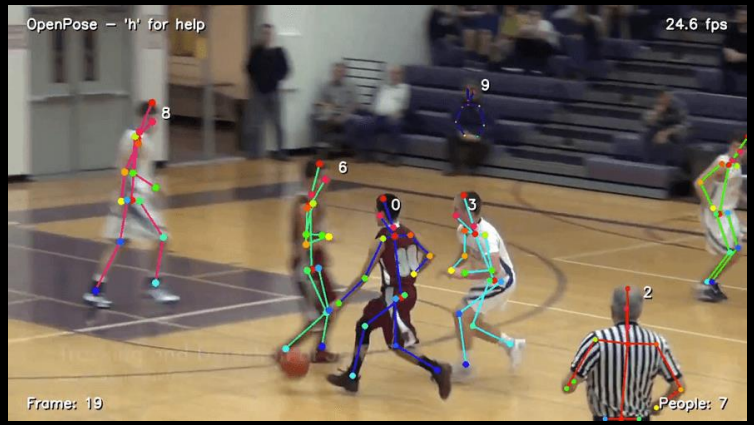


SOSCON2019

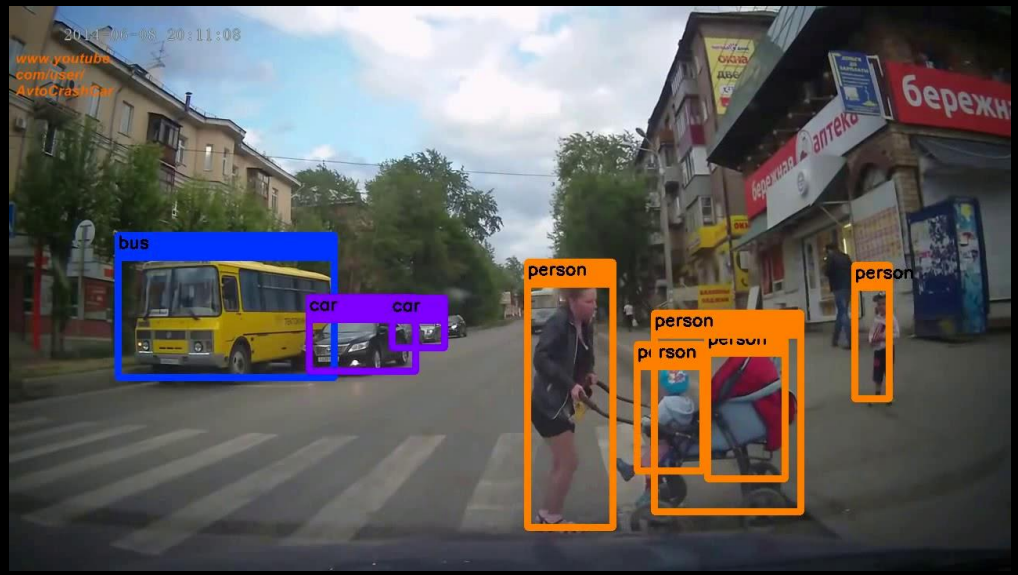
SAMSUNG OPEN SOURCE CONFERENCE 2019

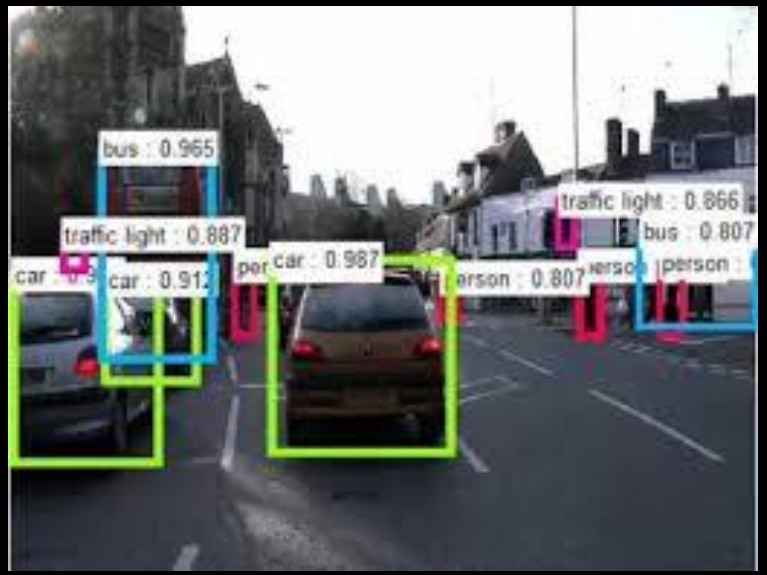
OCR (Optical Character Recognition)

Tracking Object



Autonomous Driving





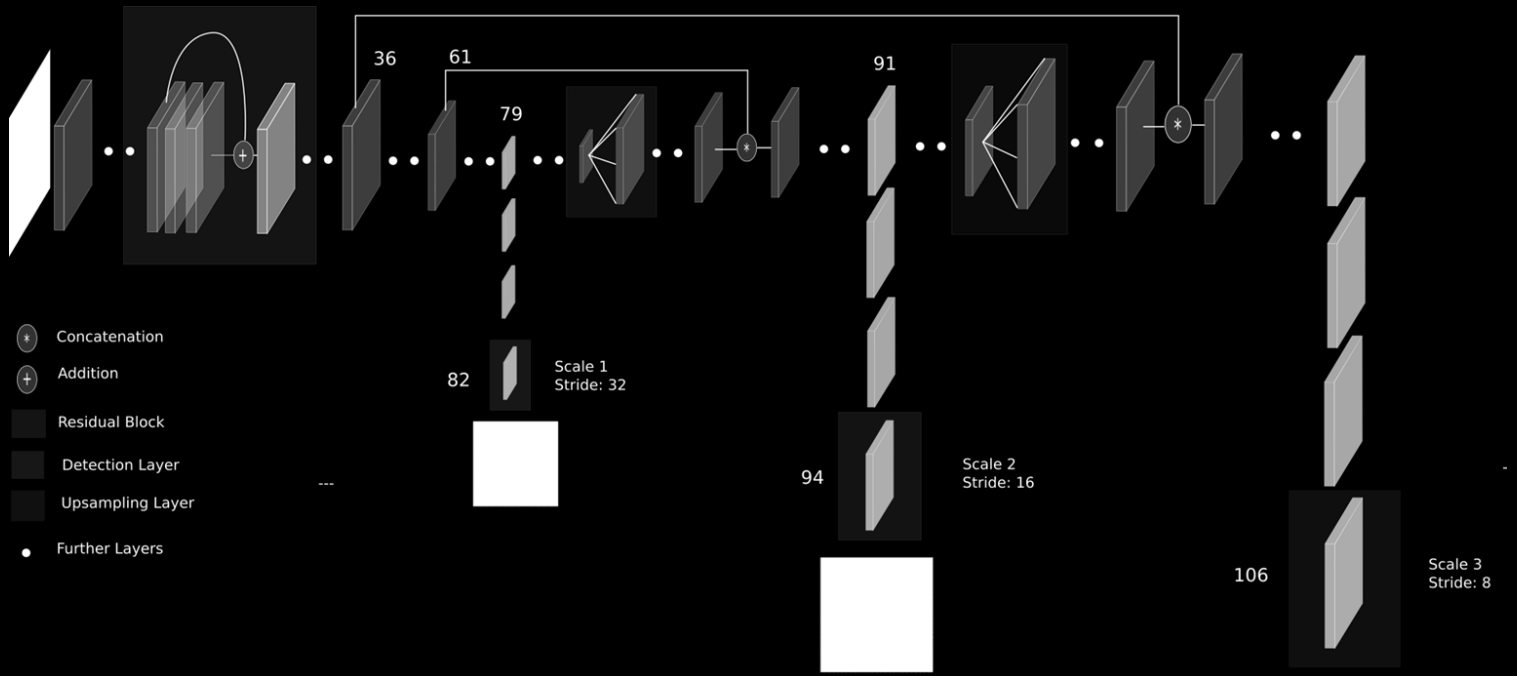
SOSCON 2019

SAMSUNG OPEN SOURCE CONFERENCE 2019

저도 Object Detection 쓰고 싶어요! 

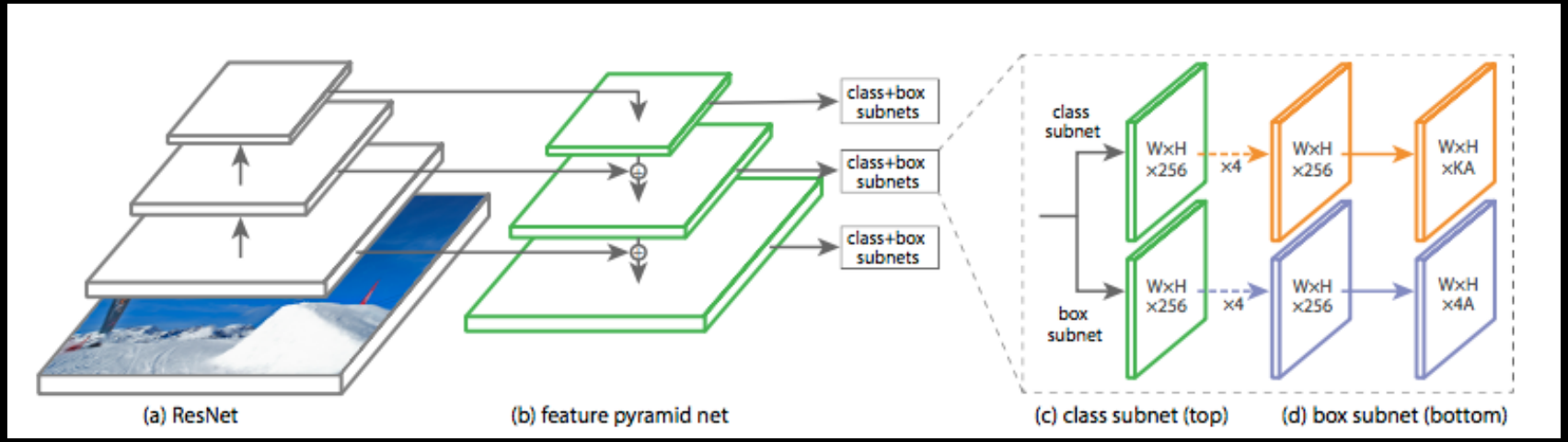
그럼 Object Detection을 어떻게 하면 되나요?

네 이런 모델 만드시면 됩니다 □



SOSCON2019

SAMSUNG OPEN SOURCE CONFERENCE 2019



SOSCON 2019

SAMSUNG OPEN SOURCE CONFERENCE 2019

모델의 input 에 맞는

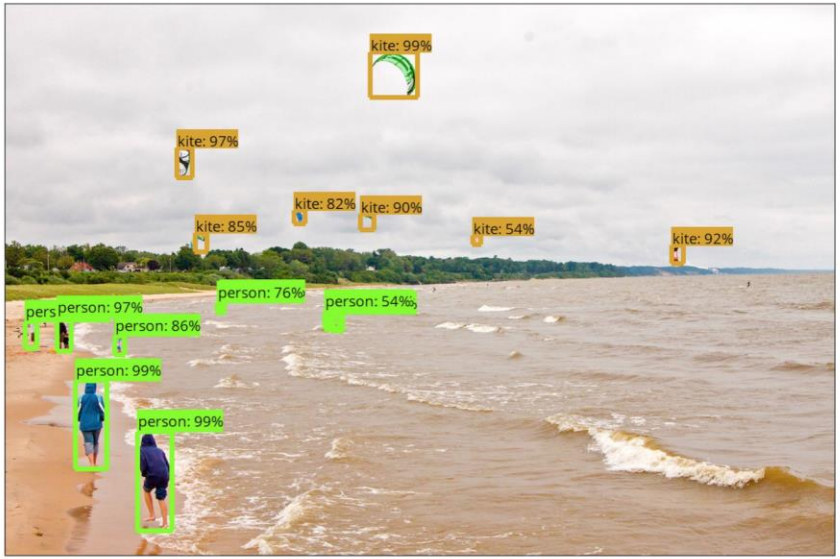
학습시키려면 데이터 로더(loader) 도 만드셔야해요!

다 만들기 어렵잖아요 □□

그렇다면 Object Detection API를 사용해보세요!

Tensorflow Object Detection API

Creating accurate machine learning models capable of localizing and identifying multiple objects in a single image remains a core challenge in computer vision. 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. At Google we've certainly found this codebase to be useful for our computer vision needs, and we hope that you will as well.



Detectron2

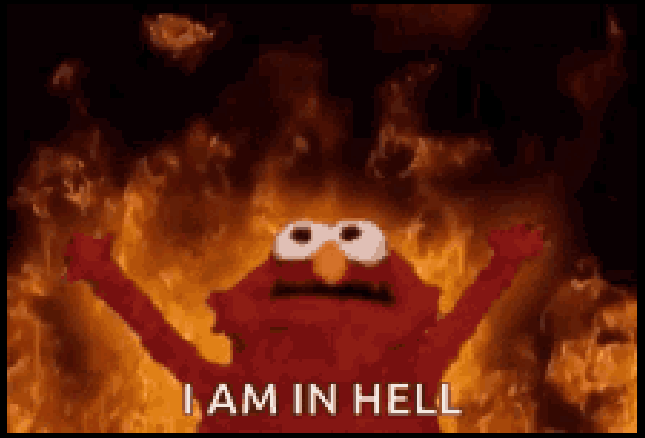
Detectron2 is Facebook AI Research's next generation software system that implements state-of-the-art object detection algorithms. It is a ground-up rewrite of the previous version, Detectron, and it originates from maskrcnn-benchmark.

rch/detectron2/blob/master/.github/Detectron2-Logo-Horz.svg

설치가 헬 🔥

Import "object_detection/protos/ssd.proto" was not found or had errors.
models: research stat:awaiting response stat:awaiting tensorflower
#4538 opened on 14 Jun 2018 by edisonchensy

"protoc --python_out =. .\object_detection\protos\calibration.proto " Error
#6405 opened on 20 Mar by smalgan



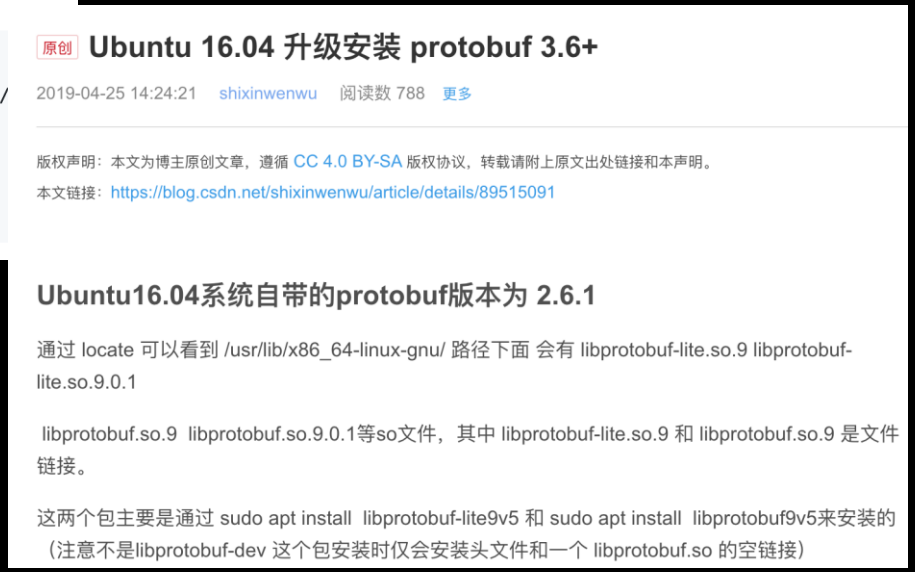
설치가 헬

are you admin?
alternatively, have you tried the following:

```
# From tensorflow/models/research/
wget -O protobuf.zip https://github.com/google/protobuf/releases/download/

unzip protobuf.zip

./bin/protoc object_detection/protos/*.proto --python_out=.
```

공식같은 에러 답변

에러 해결 방법 찾다가 중국 웹사이트까지



原创 Ubuntu 16.04 升级安装 protobuf 3.6+

2019-04-25 14:24:21 shixinwenwu 阅读数 788 更多

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。
本文链接：https://blog.csdn.net/shixinwenwu/article/details/89515091

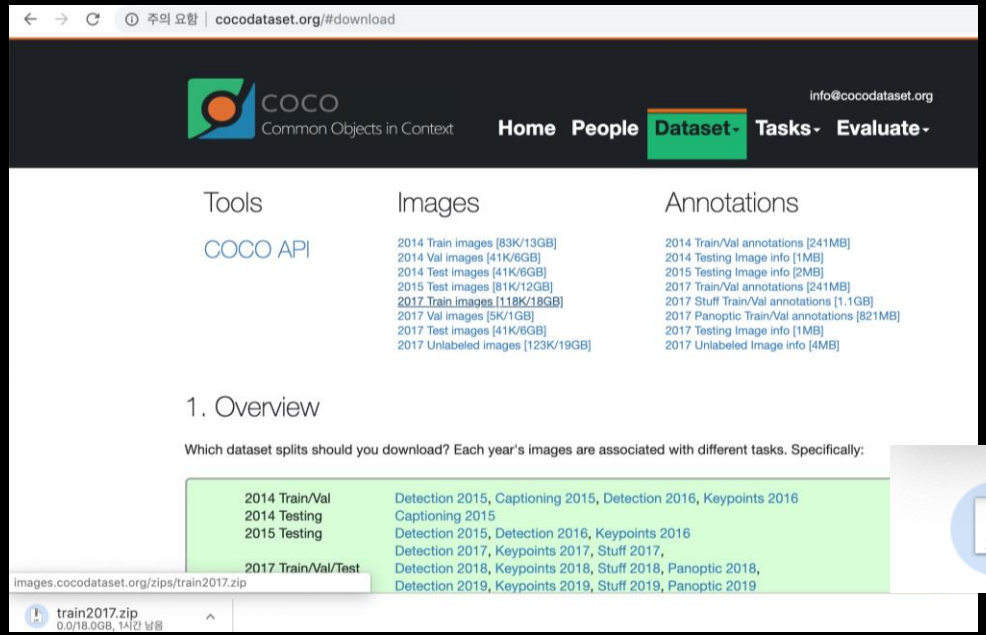
Ubuntu16.04系统自带的protobuf版本为 2.6.1

通过 locate 可以看到 /usr/lib/x86_64-linux-gnu/ 路径下面 会有 libprotobuf-lite.so.9 libprotobuf-lite.so.9.0.1

libprotobuf.so.9 libprotobuf.so.9.0.1等so文件，其中 libprotobuf-lite.so.9 和 libprotobuf.so.9 是文件链接。

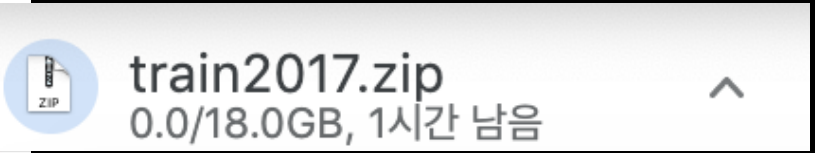
这两个包主要是通过 sudo apt install libprotobuf-lite9v5 和 sudo apt install libprotobuf9v5来安装的（注意不是libprotobuf-dev 这个包安装时仅会安装头文件和一个 libprotobuf.so 的空链接）

데이터 다운로드 받는데 한오백년



train 데이터만 다운받는데 1시간 소요

validation은 언제받고, test는 언제받지?



아웃풋을 얻기까지 너무 어려워요

```
def run_inference_for_single_image(image, graph):
  with graph.as_default():
    with tf.Session() as sess:
      # Get handles to input and o
      ops = tf.get_default_graph()
      all_tensor_names = {output.n
      tensor_dict = {}
      for key in [
          'num_detections', 'detection_boxes', 'detection_scores',
          'detection_classes', 'detection_masks'
      ]:
        tensor_name = key + ':0'
        if tensor_name in all_tensor_names:
          tensor_dict[key] = tf.get_default_graph().get_tensor_by_name(
              tensor_name)
      if 'detection_masks' in tens
        # The following processing
        detection_boxes = tf.squee
        detection_masks = tf.squee
        # Reframe is required to t
the image size.
        real_num_detection = tf.ca
        detection_boxes = tf.slice(detection_boxes, [0, 0], [real_num_detection, -1])
        detection_masks = tf.slice(detection_masks, [0, 0, 0], [real_num_detection, -1, -1])
        detection_masks_reframed = utils_ops.reframe_box_masks_to_image_masks(
            detection_masks, detection_boxes, image.shape[1], image.shape[2])
        detection_masks_reframed = tf.cast(
            tf.greater(detection_m
        # Follow the convention by
        tensor_dict['detection_mas
            detection_masks_reframe
      image_tensor = tf.get_defaul

      # Run inference
      output_dict = sess.run(tensor_dict,
                             feed_dict={image_tensor: image})

      # all outputs are float32 numpy arrays, so convert types as appropriate
      output_dict['num_detections'] = int(output_dict['num_detections'][0])
      output_dict['detection_classes'] = output_dict[
          'detection_classes'][0].astype(np.int64)
      output_dict['detection_boxes'] = output_dict['detection_boxes'][0]
      output_dict['detection_scores'] = output_dict['detection_scores'][0]
      if 'detection_masks' in output_dict:
        output_dict['detection_masks'] = output_dict['detection_masks'][0]
  return output_dict
```

```
ops = tf.get_default_graph().get_operations()
```

```
# The following processing is only for single image
detection_boxes = tf.squeeze(tensor_dict['detection_boxes'], [0])
detection_masks = tf.squeeze(tensor_dict['detection_masks'], [0])
```

```
tensor_dict['detection_masks'] = tf.expand_dims(
    detection_masks_reframed, 0)
image_tensor = tf.get_default_graph().get_tensor_by_name('image_tensor:0')
```

딥러닝의 대중화를 이끄는 숄레형아의 정신을 이어받아

누구나 쉽게 설치하고 output도 쉽게 사용할 수 있는 Object Detection API를 만드는건 어떨까?

+ 덤으로 데이터까지?

You Only Look Once & Keras (feat. Contributhon)
Keras Object Detection API

SOSCON2019

SAMSUNG OPEN SOURCE CONFERENCE 2019

어색한 미소를 가진
이분들과 함께합니다

SOSCON 2019

SAMSUNG OPEN SOURCE CONFERENCE 2019

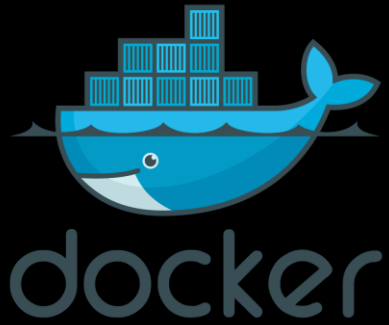
그렇다면 YOLK의 목표는 무엇인가요?

Keras만 알면 쉽게 원하는 데이터를 로드해서, 모델을 학습시킨 후

바로 모델을 사용할 수 있도록 해주는 것!

그런걸 어떻게 해줄건가요? 👀

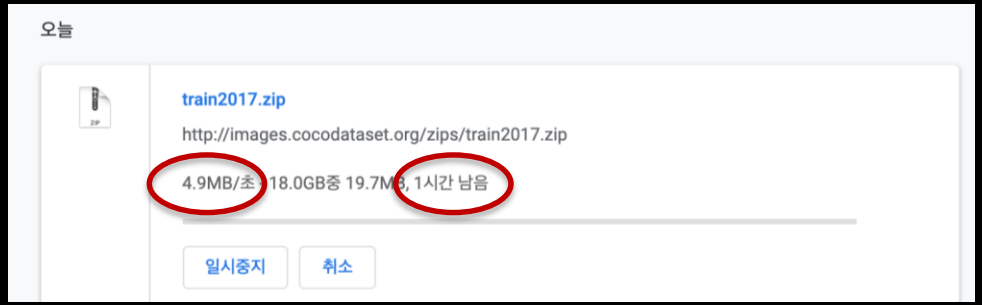
- 도커 이미지 배포를 통한 **아름다운** 설치 환경 제공



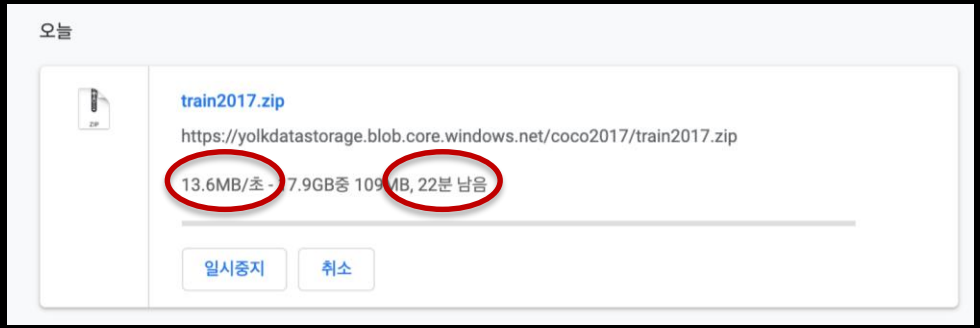
- Azure를 이용한 데이터 미러링 서버 운영으로 다운로드 속도 향상 (COCO, PASCAL VOC)

  기존 대비 3배 이상 빠른 속도(COCO기준)

기존 데이터 서버



YOLK 미러링 데이터 서버

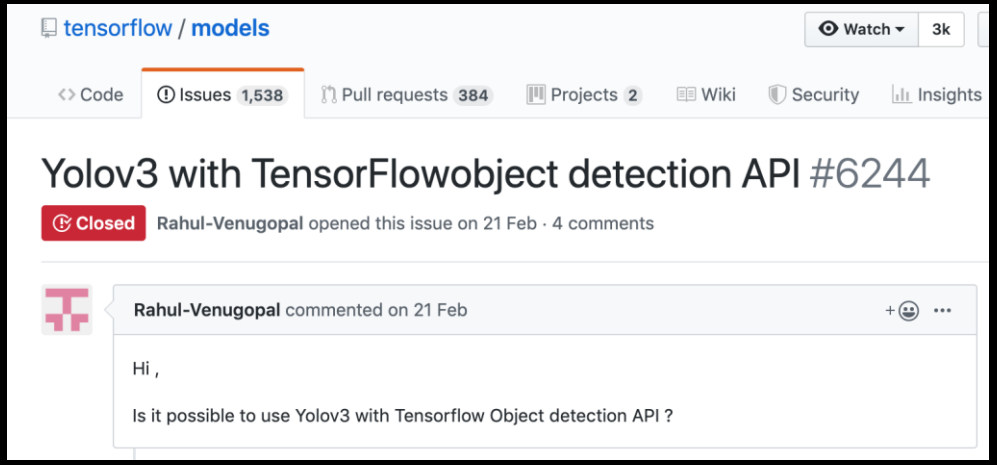


- 데이터를 모델에 사용할 수 있는 형태로 변환
  YOLK에서 제공할 모델들의 input에 맞춰서 제공

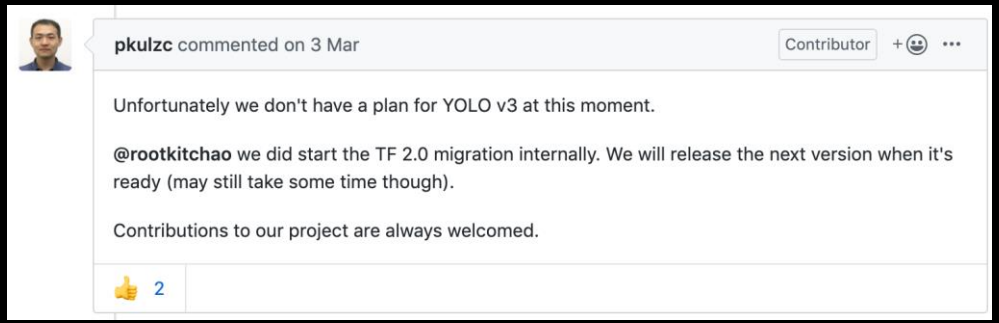
- 이해하기 쉬운 튜토리얼 제공

아래의 3가지 모델을 제공합니다!

- SSD (Single Shot Multibox Detector)
- RetinaNet
- YOLOv3



Q : YOLOv3 Tensorflow API 에서 못쓰나요?



A : YOLOv3를 위한 계획은 아직 없습니다.

YOLK는 YOLOv3 있습니다! 😎

```python
import keras
import yolk

# Load Data
voc_train, voc_valid = yolk.datasets.pascal_voc07()

# Load Base Model
base_model = yolk.models.yolov3()

# Custom Model
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
predictions = Dense(200, activation='softmax')(x)

# Compile Model
model.compile()

# Train Model
model.fit_generator(
generator = voc_train,
validation_data = voc_valid,
epochs = 20
)

# Predict
preds = model.predict_detection(x)
```

간단하게 YOLK 모듈 import

케라스의 장점인

간단한 데이터 로드

간단한 모델 로드

간단한 모델 커스텀

간단한 모델 학습

간단한 예측 결과 얻기

간단한 모델 로드

Load a (frozen) Tensorflow model into memory.

```
detection_graph = tf.Graph()
with detection_graph.as_default():
  od_graph_def = tf.GraphDef()
  with tf.gfile.GFile(PATH_TO_FROZEN_GRAPH, 'rb') as fid:
    serialized_graph = fid.read()
    od_graph_def.ParseFromString(serialized_graph)
    tf.import_graph_def(od_graph_def, name='')
```

```
# Load Base Model
base_model = yolk.models.yolov3()
```

간단한 모델 커스텀

```
model {
(... Add model config here...)
}

train_config : {
(... Add train_config here...)
}

train_input_reader: {
(... Add train_input configuration here...)
}

eval_config: {
}

eval_input_reader: {
(... Add eval_input configuration here...)
}
```

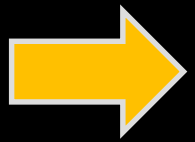
```
model {
    ssd {
        num_classes: 1 # Set this to the number of different label classes
        box_coder {
            faster_rcnn_box_coder {
                y_scale: 10.0
                x_scale: 10.0
                height_scale: 5.0
                width_scale: 5.0
            }
        }
        matcher {
            argmax_matcher {
                matched_threshold: 0.5
                unmatched_threshold: 0.5
                ignore_thresholds: false
                negatives_lower_than_unmatched: true
                force_match_for_each_row: true
            }
        }
        similarity_calculator {
            iou_similarity {
            }
        }
        anchor_generator {
            ssd_anchor_generator {
                num_layers: 6
                min_scale: 0.2
                max_scale: 0.95
                aspect_ratios: 1.0
                aspect_ratios: 2.0
                aspect_ratios: 0.5
                aspect_ratios: 3.0
                aspect_ratios: 0.3333
                reduce_boxes_in_lowest_layer: true
            }
```

간단한 모델 커스텀

```
import keras
import yolk

# Load Base Model
base_model = yolk.models.yolov3()

# Custom Model
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
predictions = Dense(200, activation='softmax')(x)
```

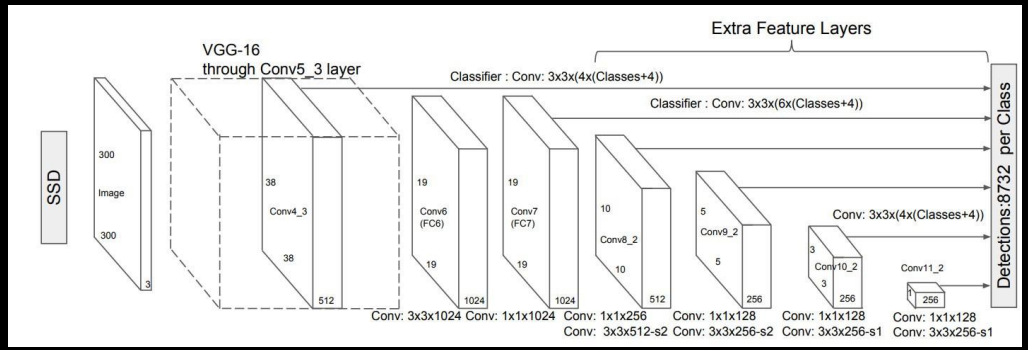
간단한 예측 결과 얻기

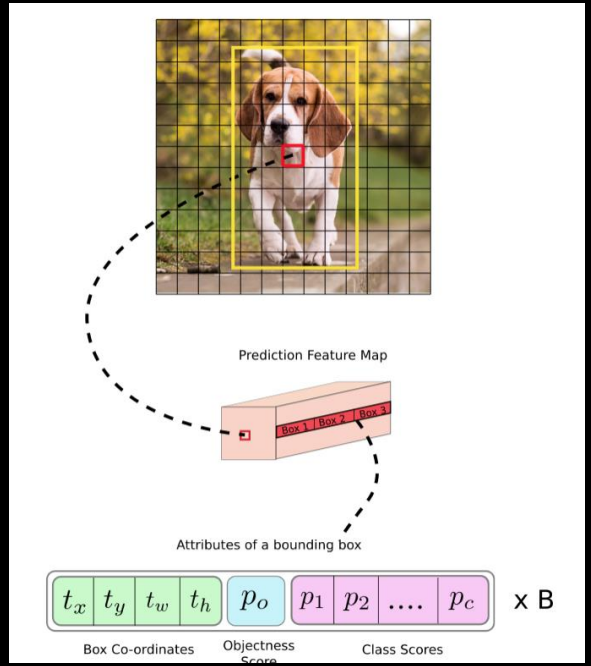
```python
def run_inference_for_single_image(image, graph):
  with graph.as_default():
    with tf.Session() as sess:
      # Get handles to input and output tensors
      ops = tf.get_default_graph().get_operations()
      all_tensor_names = {output.name for op in ops for output in op.outputs}
      tensor_dict = {}
      for key in [
          'num_detections', 'detection_boxes', 'detection_scores',
          'detection_classes', 'detection_masks'
      ]:
        tensor_name = key + ':0'
        if tensor_name in all_tensor_names:
          tensor_dict[key] = tf.get_default_graph().get_tensor_by_name(
              tensor_name)
      if 'detection_masks' in tensor_dict:
        # The following processing is only for single image
        detection_boxes = tf.squeeze(tensor_dict['detection_boxes'], [0])
        detection_masks = tf.squeeze(tensor_dict['detection_masks'], [0])
        # Reframe is required to translate mask from box coordinates to image coordinates and fit
the image size.
        real_num_detection = tf.cast(tensor_dict['num_detections'][0], tf.int32)
        detection_boxes = tf.slice(detection_boxes, [0, 0], [real_num_detection, -1])
        detection_masks = tf.slice(detection_masks, [0, 0, 0], [real_num_detection, -1, -1])
        detection_masks_reframed = utils_ops.reframe_box_masks_to_image_masks(
            detection_masks, detection_boxes, image.shape[1], image.shape[2])
        detection_masks_reframed = tf.cast(
            tf.greater(detection_masks_reframed, 0.5), tf.uint8)
        # Follow the convention by adding back the batch dimension
        tensor_dict['detection_masks'] = tf.expand_dims(
            detection_masks_reframed, 0)
      image_tensor = tf.get_default_graph().get_tensor_by_name('image_tensor:0')

      # Run inference
      output_dict = sess.run(tensor_dict,
                             feed_dict={image_tensor: image})

      # all outputs are float32 numpy arrays, so convert types as appropriate
      output_dict['num_detections'] = int(output_dict['num_detections'][0])
      output_dict['detection_classes'] = output_dict[
          'detection_classes'][0].astype(np.int64)
      output_dict['detection_boxes'] = output_dict['detection_boxes'][0]
      output_dict['detection_scores'] = output_dict['detection_scores'][0]
      if 'detection_masks' in output_dict:
        output_dict['detection_masks'] = output_dict['detection_masks'][0]
  return output_dict
```



```python
# Predict
preds = model.predict_detection(x)
```

각 모델의 output shape에 따른 처리



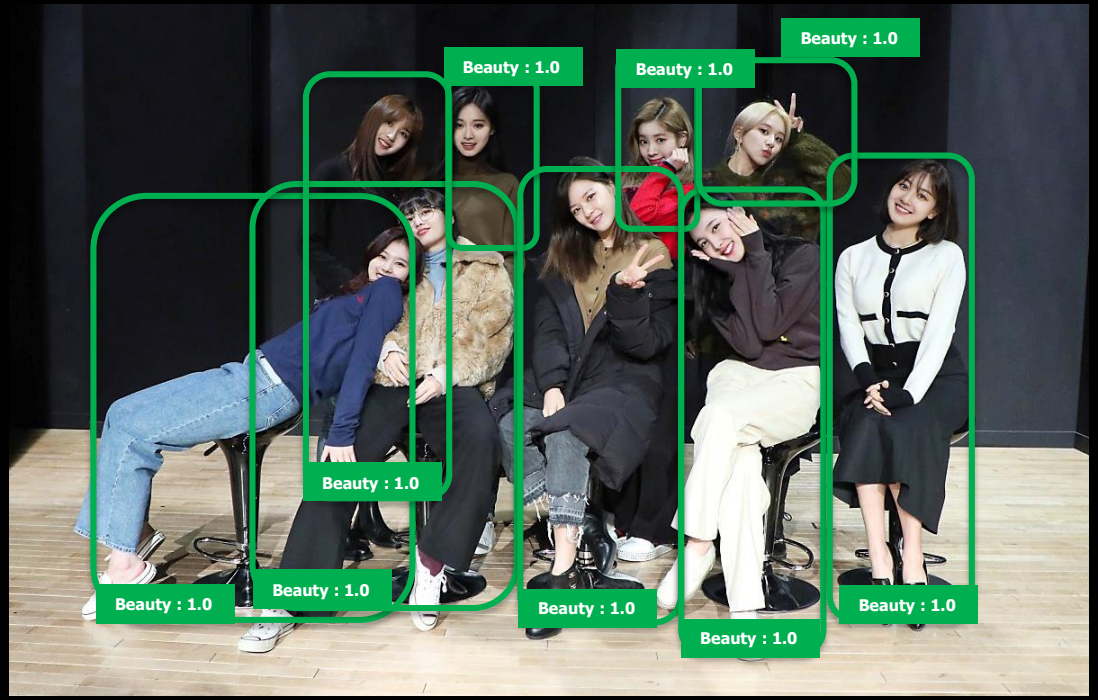


```
# Predict
preds = model.predict_detection(x)
```

Bounding Box

Detection Class

Detection Score



곧 세상에 나올

Keras Object Detection API YOLK를

기대해주세요!

THANK YOU

SOSCON 2019

SAMSUNG OPEN SOURCE CONFERENCE 2019