



YOLK

(You Only Look Keras)

NAVER 김수정

연사 소개



NAVER AI Engineer

- 쇼핑 검색
- Attribute Value Extraction

Opensource Contribution 2회 참가

- Keras tutorial 번역
- YOLK, object detection API



<https://www.facebook.com/sd.hi.940>



<https://github.com/SooDevv>

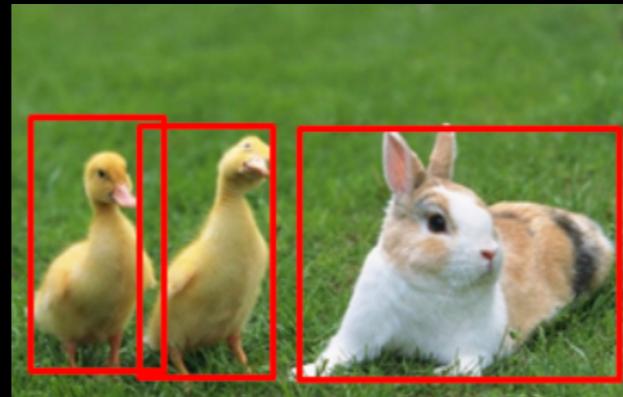
Image Recognition

이미지에 있는 대상이 무엇인지, 어떤 위치에 있는지 파악하는 Task

Classification



Detection



Segmentation



토끼

오리, $[x_1, y_1, x_2, y_2]$
오리, $[x_3, y_3, x_4, y_4]$
토끼, $[x_5, y_5, x_6, y_6]$

오리1
오리2
토끼

Image Recognition

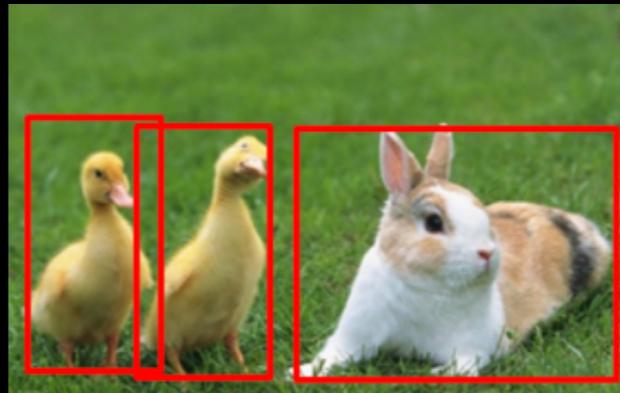
이미지에 있는 대상이 무엇인지, 어떤 위치에 있는지 파악하는 Task

Classification



토끼

Detection



오리, $[x_1, y_1, x_2, y_2]$
오리, $[x_3, y_3, x_4, y_4]$
토끼, $[x_5, y_5, x_6, y_6]$

Segmentation

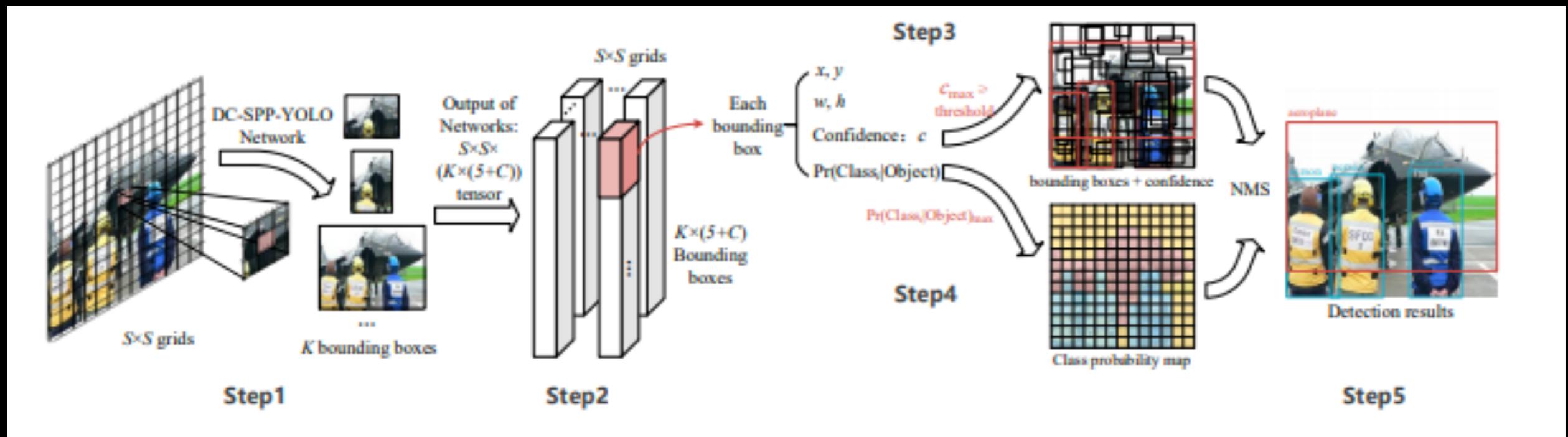


오리|1
오리|2
토끼

Object Detection

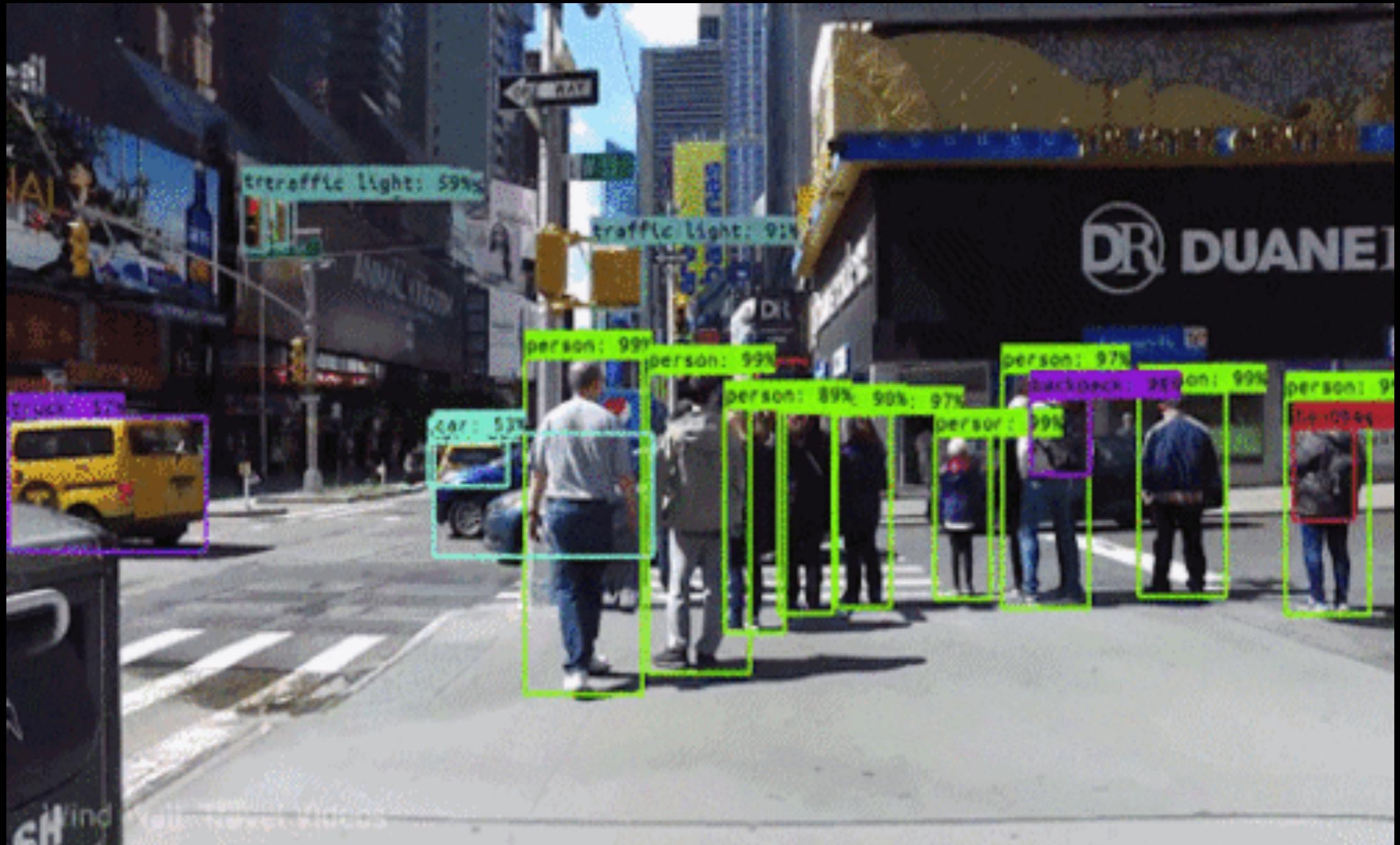
Object detection = Classification + Localization

어떤 물체인지 분류하고 그 물체가 어느 위치에 있는지 박스 형태로 검출하는 모델

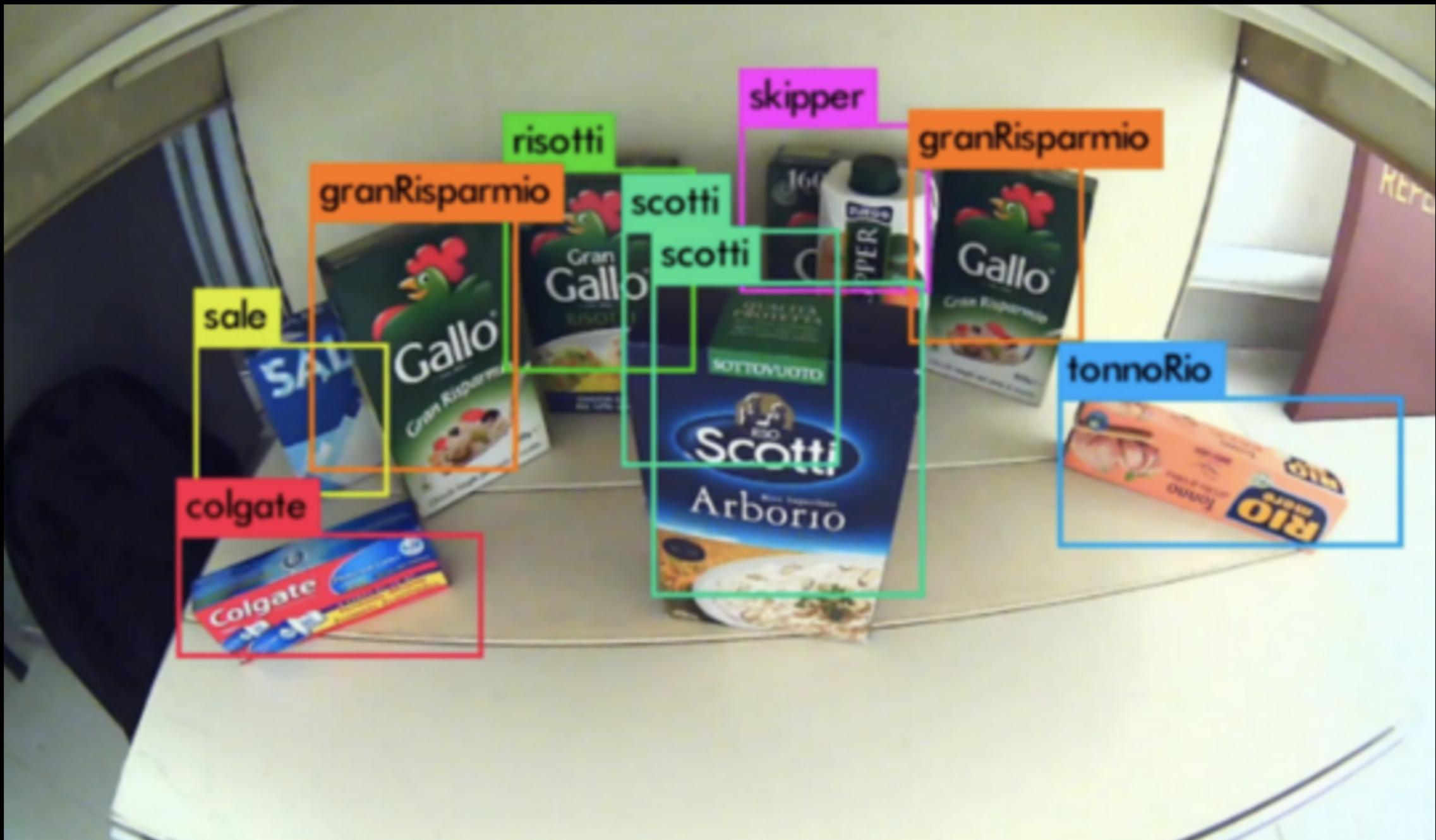


그래서 이 기술이 어디에 적용되나요?

자율주행 자동차



무인 점포 - 상품 인식

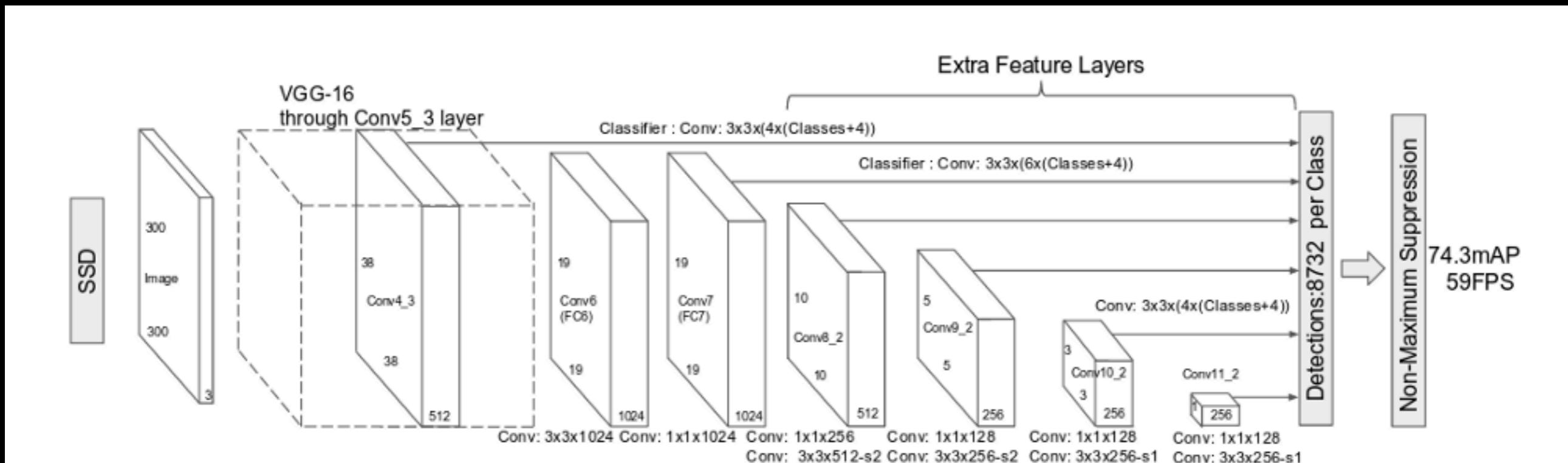


제조업 - 불량품 검출



와! 그럼 이 기술을 어떻게 쓸 수 있나요?

Single Shot MultiBox Detector Architecture



TODO 🤔🤔

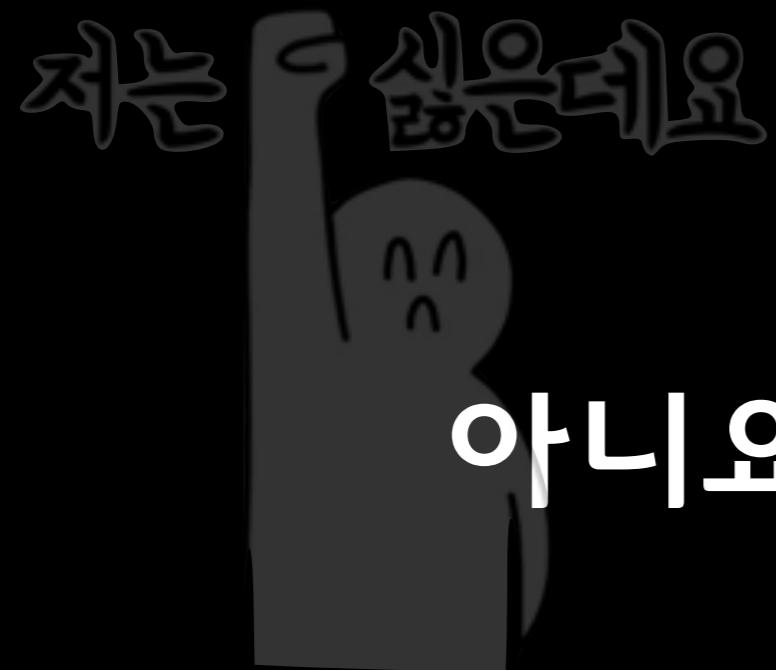
- 모델 이해
- Data Loader 만들기
- Training code 작성
- Inference code 작성
- Training, Training, Training.. many many GPUs..

...

참 쉽죠?



구현하시면 됩니다 ^_^



아니요.. 저는 안쉬운데요...

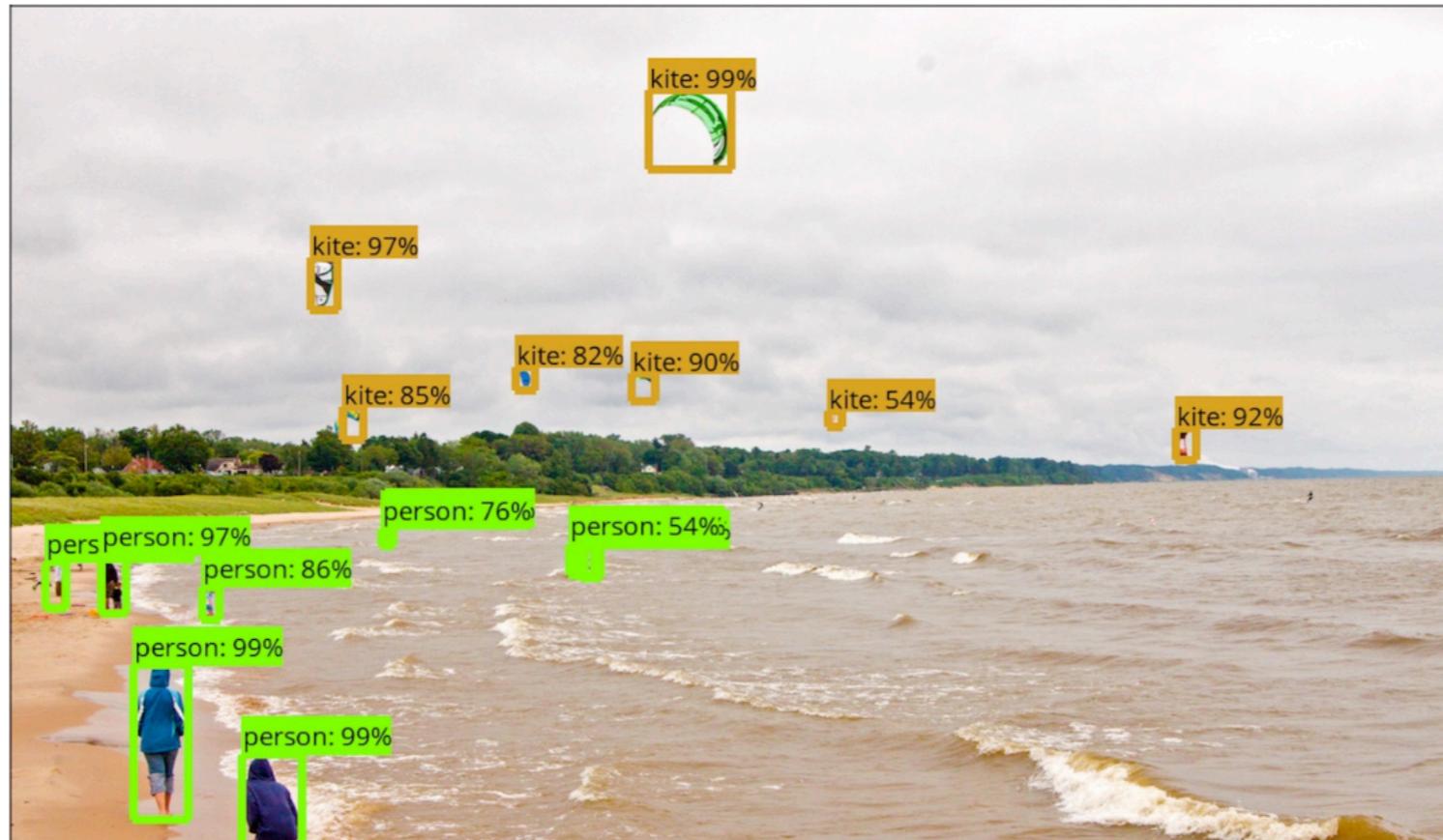
처음부터 다 구현하지 말고,
Object Detection API를 사용해보면 어떨까요?

Make it easy to construct, train and deploy object detection models!!



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.



그런데 말입니다



설치가.... 헬 🔥

➊ Import "object_detection/protos/ssd.proto" was not found or had errors.

models: research stat:awaiting response stat:awaiting tensorflow

#4538 opened on 14 Jun 2018 by edisonchensy

➊ "protoc --python_out =. .\object_detection\protos\calibration.proto " Error

#6405 opened on 20 Mar by smalgan

I AM IN HELL

버전도.... 헬 🔥

SSD
ver 2.1

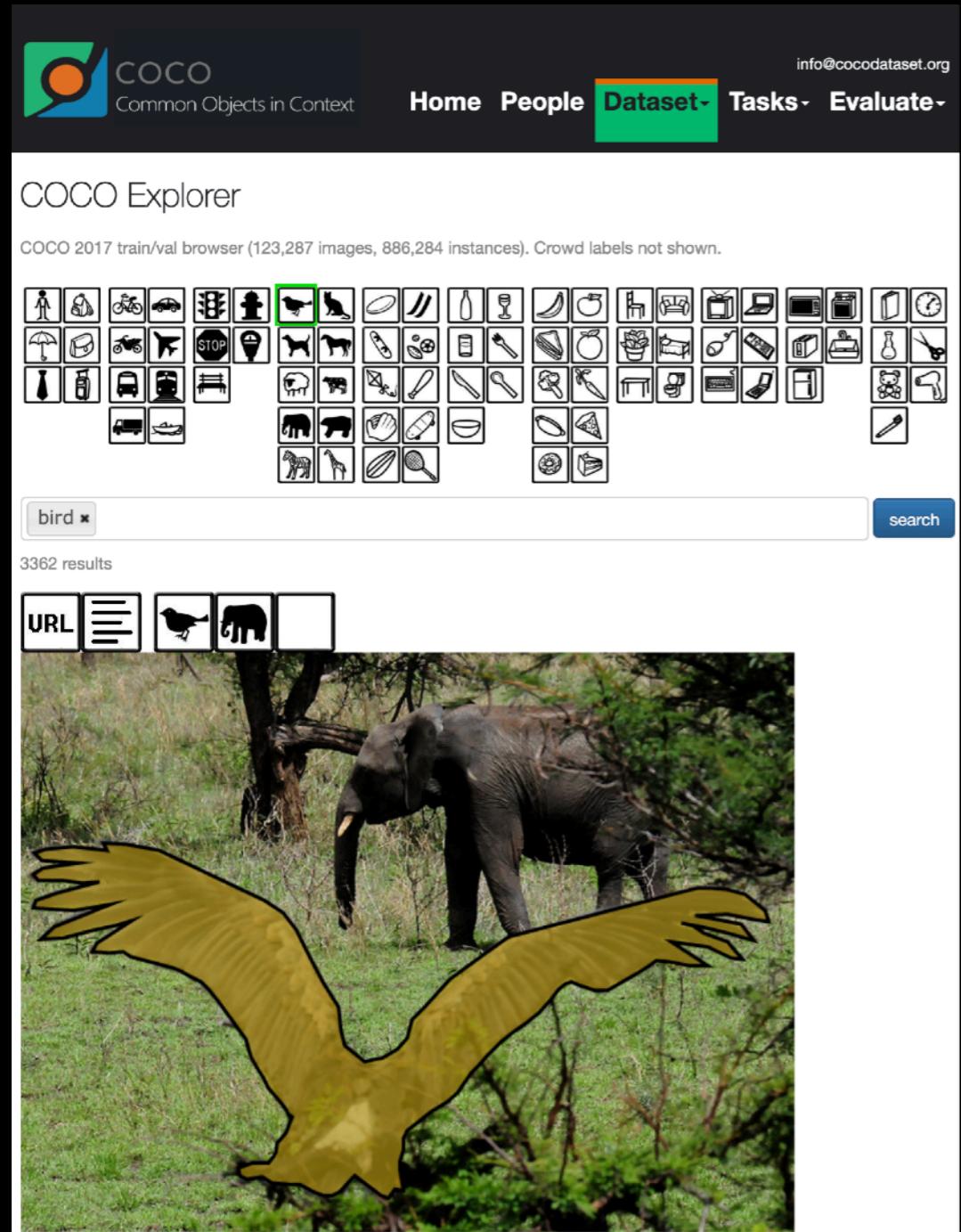
RetinaNet
ver 2.3

YOLOv3
ver 1.9

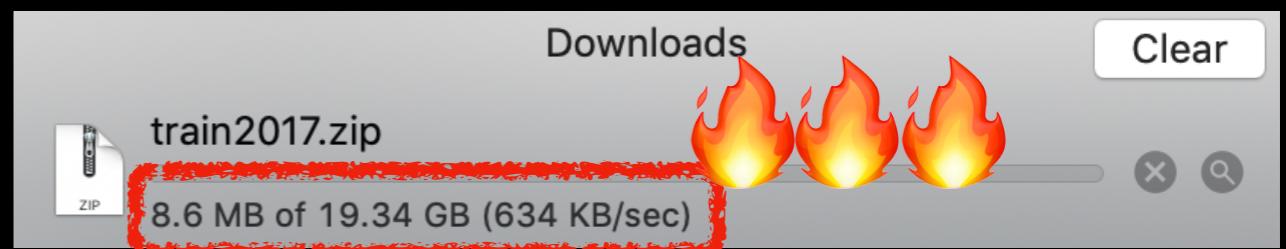


```
$ git clone github/ssd  
$ git clone github/retinanet  
$ git clone github/yolov3
```

데이터도.... 헬 🔥



무려 8.47시간!!



코드도.... 헬



결과값 하나 얻는데
이렇게나 복잡할수가!! 😱

```
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: output
                for output in ops.get_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 match 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.float32)
                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
```

누구나 쉽게 설치하고
결과값도 좀 더 간결하고 쉽게 사용할 수 있는
Object Detection API를 만들어보면 어떨까?

딥러닝의 대중화

딥러닝 장벽 봉괴



프랑소와 솔레 (n세) | 캐라스 창시자

You Only Look Keras



Keras Object Detection API

(feat. Opensource Contribution)

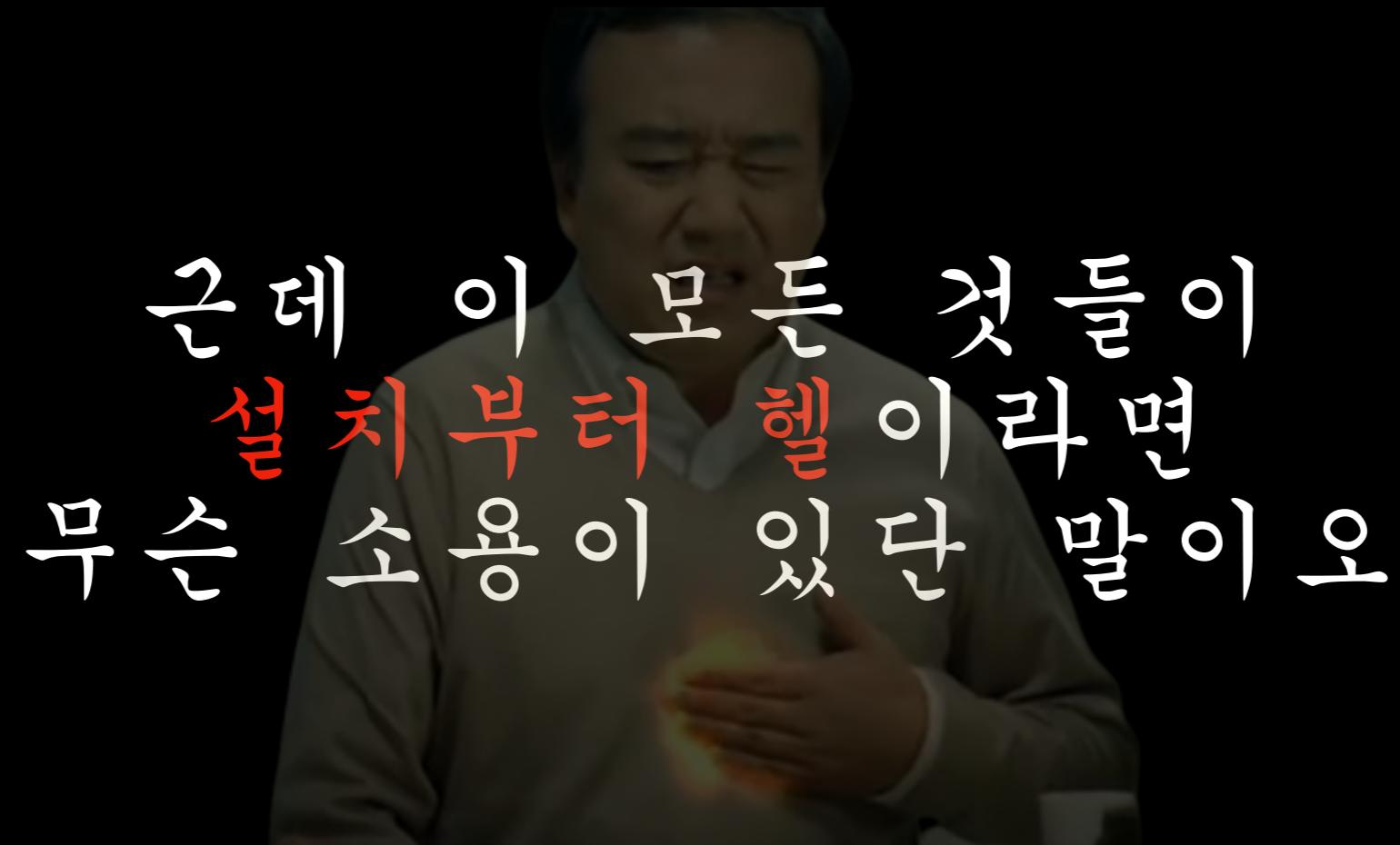
멋진분들과 함께합니다 😍



YOLK의 목표는

Keras만 알면

- 쉽고 빠르게 데이터를 로드하여
- 쉽고 빠르게 모델도 로드
- 결과값을 쉽게 확인할 수 있도록!

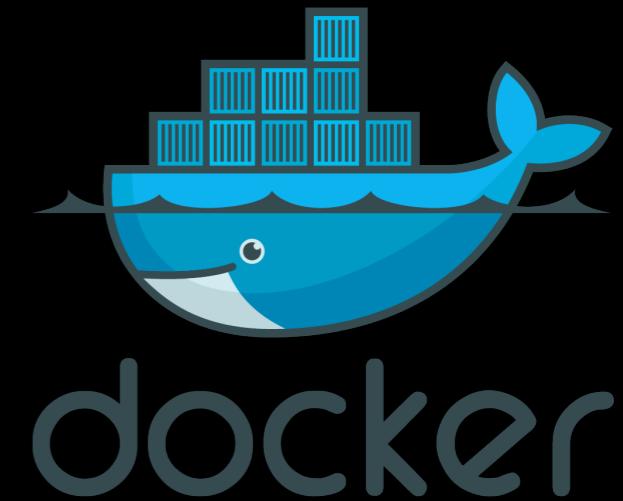


근데 이 모든 것들이
설치부터 헬이라면
무슨 소용이 있단 말이오

도커 이미지 배포를 통한 아름다운 설치 환경 제공



```
# pull yolk docker image  
$ docker pull kerasyolk/yolk
```



버전 통일

SSD
ver 2.3

RetinaNet
ver 2.3

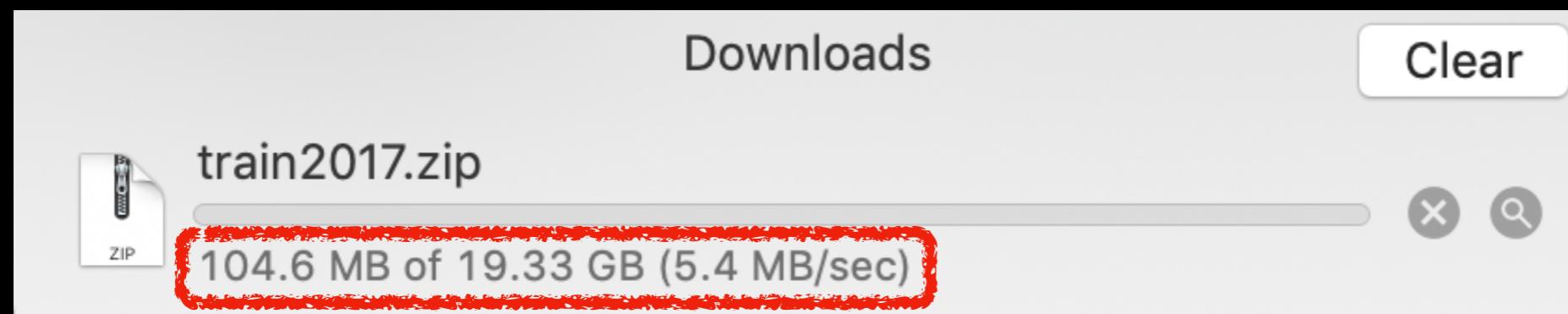
YOLOv3
ver 2.3

미러링 서버 운영으로 데이터 다운로드 시간 단축



Azure를 이용한 데이터 미러링 서버 운영으로
다운로드 속도 향상 (COCO, PASCAL VOC)
기존 대비 8배 이상 빠른 속도

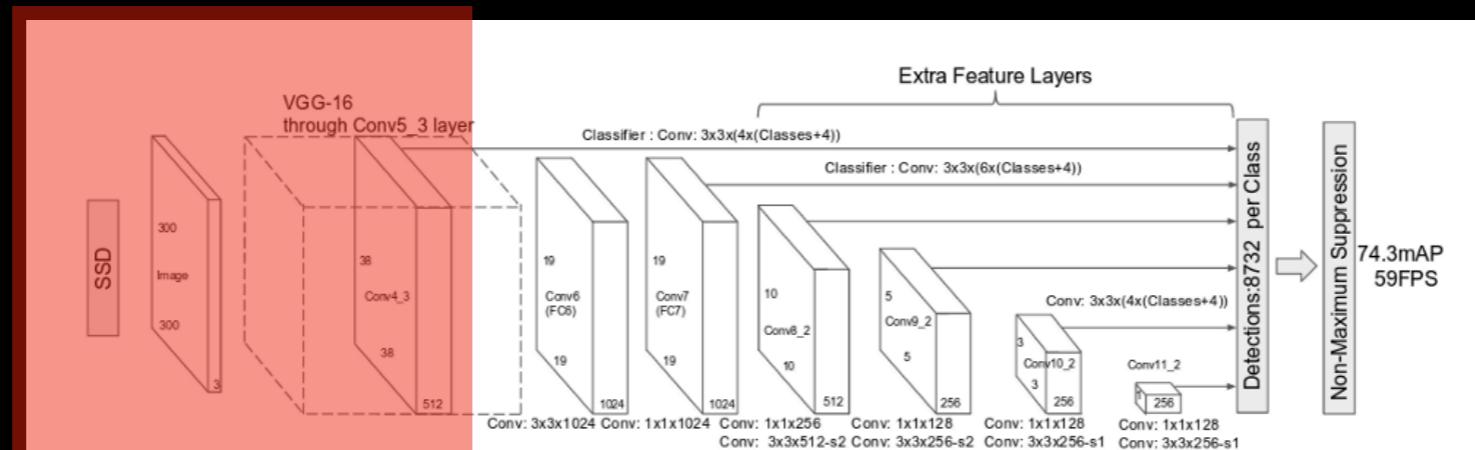
겨우 0.99 시간!!



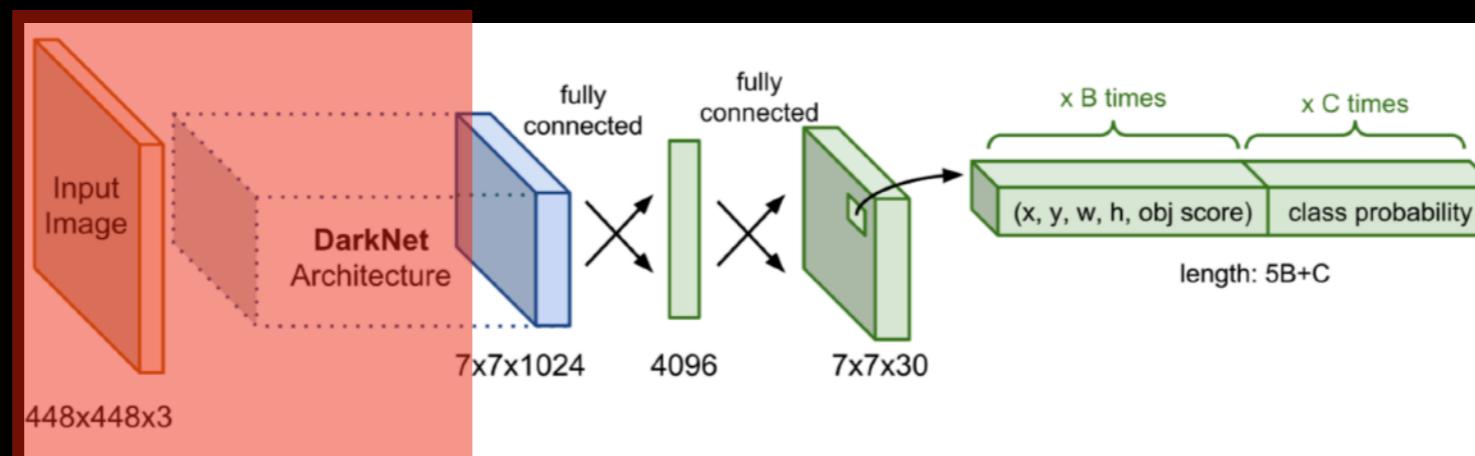


환경 셋팅 완료
데이터 준비 완료

모델마다 다른 input의 형태
YOLO에서 제공하는 모델들의 input에 맞춰서 제공



< Single Shot MultiBox Detector Architecture >



< YOLOv3 Architecture >

아래의 3가지 모델을 제공합니다! (2019. 12. 17기준)

SSD
ver 2.3

RetinaNet
ver 2.3

YOLOv3
ver 2.3

틈새 자랑

tensorflow / models

Code Issues Pull requests Projects Wiki Security Insights

Yolov3 with TensorFlow Object detection API #6244

Closed Rahul-Venugopal opened this issue on 21 Feb · 4 comments

Rahul-Venugopal commented on 21 Feb

Hi ,

Is it possible to use Yolov3 with Tensorflow Object detection API ?

pkulzc commented on 3 Mar

Contributor

Unfortunately we don't have a plan for YOLO v3 at this moment.

@rootkitchao we did start the TF 2.0 migration internally. We will release the next version when it's ready (may still take some time though).

Contributions to our project are always welcomed.

2

Q.
TensorFlow Object API에서
YOLOv3 못쓰나요?

A.
YOLOv3를 위한 계획은
아직 없습니다.

★★★
여러분
YOLK는 YOLOv3 있습니다 🤘

Keras만 알면

- 쉽고 빠르게 데이터를 로드하여
- 쉽고 빠르게 모델도 로드
- 결과값을 쉽게 확인할 수 있도록!

우리 코드로 얘기해봐요

```
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)
```

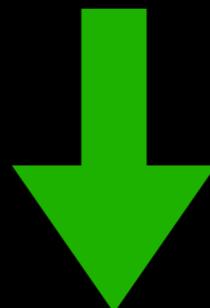
간단하게 YOLK 모듈 Import

케라스의 장점을 YOLK에 적용

- 간단한 데이터 로드
- 간단한 모델 로드
- 간단한 모델 커스텀
- 간단한 모델 학습
- 간단한 예측값 얻기

간단한 이미지 전처리

```
image_np = np.array(Image.open(image_path))
image = np.asarray(image_np)
# The input needs to be a tensor, convert it using `tf.convert_to_tensor`.
input_tensor = tf.convert_to_tensor(image)
# The model expects a batch of images, so add an axis with `tf.newaxis`.
input_tensor = input_tensor[tf.newaxis,...]
```



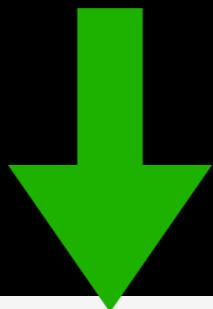
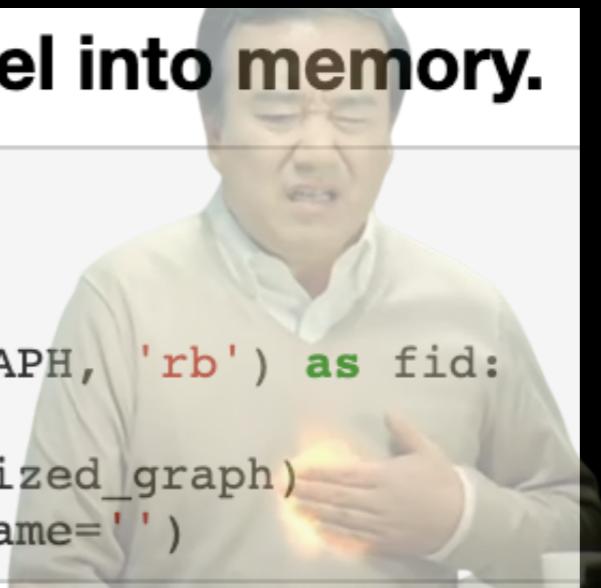
```
image_path = 'examples/yolk.jpg'
image = yolk.detector.preprocessing_image(image_path)
```



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()
```

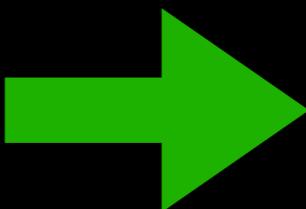


간단한 예측값 얻기

```
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
```

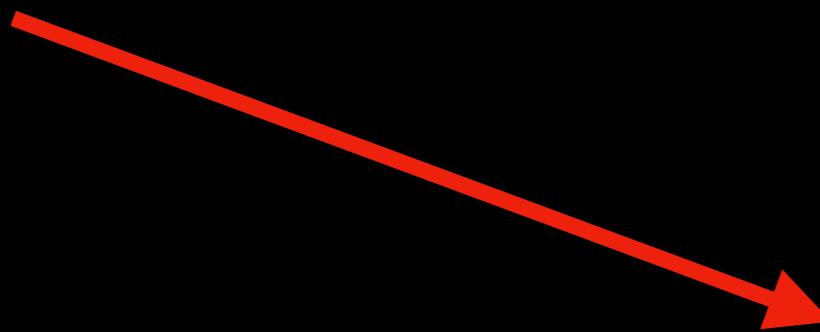


```
# Predict
preds = model.predict_detection(x)
```



간단한 예측값 얻기

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



class	conf	xmin	ymin	xmax	ymax
[15.	0.9	140.09	158.4	170.87	293.25]
[15.	0.9	204.24	148.23	236.9	291.54]
[15.	0.89	121.46	155.45	151.04	296.76]
[15.	0.87	64.07	154.66	102.36	303.71]
[16.	0.85	225.44	151.63	285.09	210.28]
[15.	0.83	181.8	153.28	210.82	294.92]
[15.	0.81	164.55	159.46	198.82	295.96]
[15.	0.79	99.69	152.57	133.37	299.31]
[15.	0.55	39.12	128.39	77.03	298.81]
[9.	0.52	253.35	190.33	301.64	286.45]]

Step 4. Visualize the predictions

```
yolk.detector.show_result(image_path, y_pred_thresh)
```



https://github.com/KerasKorea/KerasObjectDetector/blob/develop/SSD_inference.ipynb

Another project 케라스 공식문서 번역

 [keras-team / keras-docs-ko](#)  Watch ▾ 15 ★ Star 82  Fork 51

[Code](#) [Issues 20](#) [Pull requests 41](#) [Actions](#) [Projects 0](#) [Wiki](#) [Security](#) [Insights](#)

Korean translation of the Keras documentation.

473 commits 4 branches 0 packages 0 releases 21 contributors

Branch: master ▾ [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#) [Clone or download ▾](#)

 **fuzzythecat** Merge pull request #142 from airplane2230/patch-7 ... Latest commit 14fa64c 18 days ago

 sources	Merge pull request #142 from airplane2230/patch-7	18 days ago
 .gitignore	Create .gitignore	9 months ago
 README.md	Update README.md	2 months ago
 mkdocs.yml	Add mkdocs file	11 months ago

 [README.md](#)

Korean translation of the Keras documentation

This is the repository for the Korean-language `.md` sources files of [keras.io](#).

Existing files in `sources/` should be edited in-line.

케라스 공식 문서 한국어판

케라스 공식 문서의 한국어판입니다. 이미 딥러닝에 익숙한 연구자 및 개발자 외에도 처음 딥러닝을 접하는 사용자들이 최대한 쉽게 이해하고 사용할 수 있도록 그 의미와 용법, 용례가 정확하고 명료하게 그리고 최대한 자연스러운 문장으로 나타나도록 작성되었습니다.   

Opensource Contribution 최우수상 수상



Can you join us?

번역



[keras-team / keras](#)



[keras-team / keras-docs-ko](#)

Object Detection



[KerasKorea / KerasObjectDetector](#)

네이버 쇼핑검색 서비스 개발자 모집



[조직소개]

쇼핑검색개발은 사용자가 상품을 검색하여 상품의 정보를 확인하고 비교하는 [프런트엔드\(웹\)](#)와 [API](#)를 개발하고 있습니다. 또한 사용자가 입력하는 쿼리의 의도를 분석하고 그에 따른 가장 적합한 검색 결과와 검색 도구 등을 보여주기 위한 관리 툴도 개발하고 있습니다. 검색 결과를 빠르고 안정적으로 서비스하며 검색 결과에 대한 사용자 만족도를 높이기 위해 노력합니다.

“너 해보고 싶은거 다 해~”

- Node/React/SpringMVC로 웹 서비스를 개발/운영하고 있습니다.
- Java/Spring으로 API와 배치 및 실시간 처리 시스템을 개발하고 운영합니다.
- Elastic Stack/MongoDB/Kafka도 마음껏 경험할 수 있습니다.
- Hadoop/Spark/Druid등으로 로그 분석도 합니다.
- k8s 기반 서비스 운영도 경험하실 수 있습니다.

“이런 분과 함께하고 싶습니다.”

- 활발한 코드 리뷰와 기술 공유를 통해 동료와 함께 성장하길 원하시는 분
- 긍정적인 마인드로 문제를 같이 해결하실 분

[역할]

- PC/모바일 웹 서비스 개발 및 운영
- 통합 검색 및 기타 서비스를 위한 API 개발 및 운영
- 검색도구 운영을 위한 관리툴 개발 및 운영
- 로그 분석을 통한 서비스 품질 개선

[필요역량]

다음 중 하나의 조건은 만족하셔야 합니다.

- Node/React로 백엔드 혹은 프런트엔드(SPA) 서비스 구현 경험이 있으신 분
- SpringMVC로 백엔드 혹은 프런트엔드 서비스 구현 경험이 있으신 분

[우대사항]

- 검색 서비스 경험이 있으신 분
- React 개발 경험이 많으신 분

네이버 쇼핑검색 플랫폼 개발자 모집



[조직소개]

쇼핑검색플랫폼은 국내외 쇼핑몰들로부터 검색 플랫폼에 노출되는 다양한 정보를 수신하여 서비스로 제공하는 조직입니다. 본 포지션에서는 데이터의 [수급-정제-분석-검색](#)에 이르는 전반적인 개발 업무를 진행하게 되며, 최신 기술 ([ELK](#), [k8s](#), [Spark](#), [Spring](#), [Kafka](#), [AI](#) 등)을 활용하여 사용자에게 최적의 경험을 제공할 수 있는 개발 업무를 하고 있습니다.

[역할]

- 쇼핑 데이터 수급 및 관리 개발
- 가격 비교를 위한 제반 개발
- 쇼핑 데이터 검색 관련 개발
- 상품 정보에 새로운 가치를 부여할 수 있는 R&D 업무

[필요역량]

- 기술 전파, 코드리뷰 등 주변에 자극을 주시는 분
"저 사람 덕분에 이런걸 알게 되다니!"
- 새로운 기술에 호기심이 많으신 분
(보는데 그치지 않고 직접 구현해보고 실험해보는걸 좋아하시는 분)
- Open Source 기여 경험이 있으신 분
- [Hadoop ecosystem](#) 및 [Kafka / Spark](#) 등에 대한 경험이 있으신 분
- Production level의 [machine learning project](#) 경험이 있으신 분

[우대사항]

- 대규모 데이터 처리에 대한 경험을 보유하신 분
- Kafka 기반의 스트리밍 처리에 관심이 많으신 분
- Spring 기반 프레임워크 및 [React / Node.JS / Mongo DB](#)에 대해 능숙하신 분
- [Natural Language Processing](#)에 대한 이해도가 높으신 분
- 기술 : [Spring / React / Node.JS / Elastic Stack / Hadoop / Kafka / Spark / Docker / PostgresDB / Mongo DB / Redis / Tensorflow](#) 등

감사합니다