

AI 모듈

YOLO(You Only Look Once)

team3

정시윤
김종명
노태현
이주연
전아현

YOLO

YOLO란?

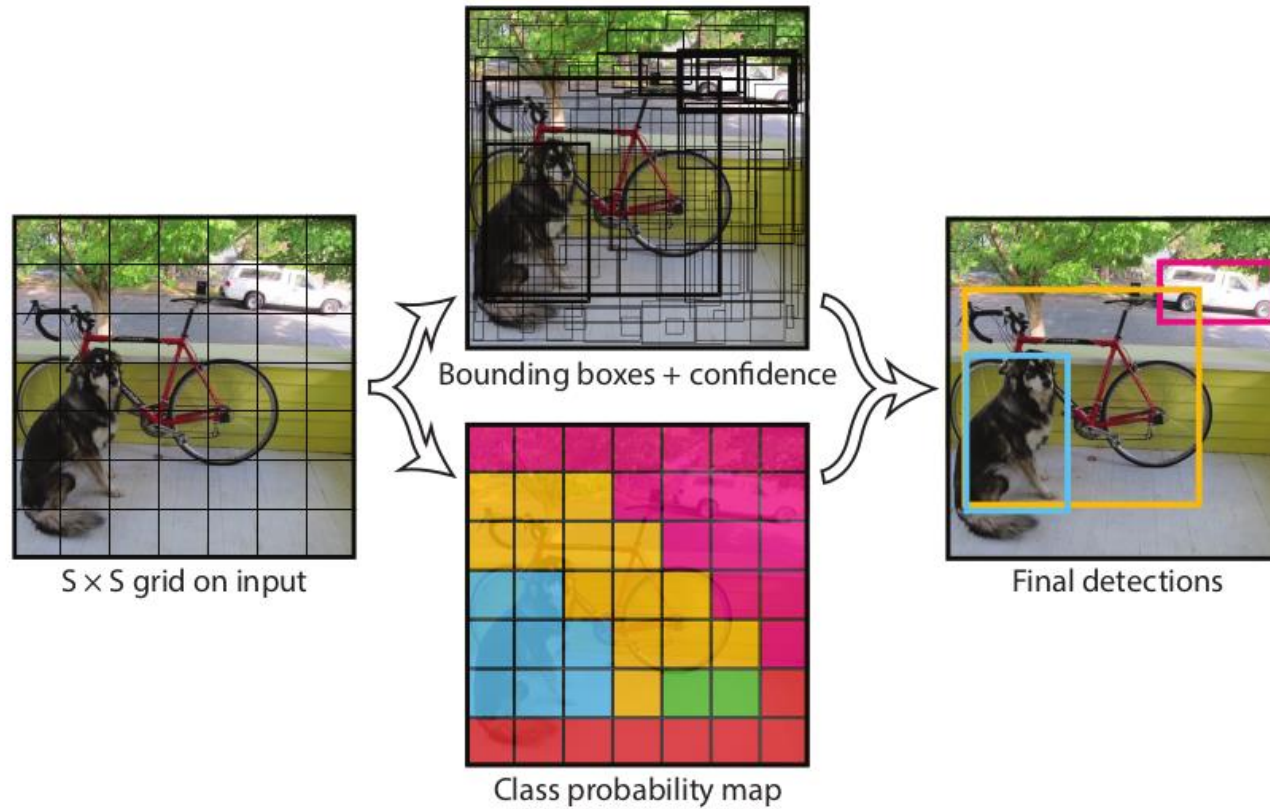


You Only Look Once

이미지를 한번 보는 것으로 Object의 종류와 위치를 추측하는 딥러닝 기반의 물체인식 알고리즘

YOLO

YOLO란?



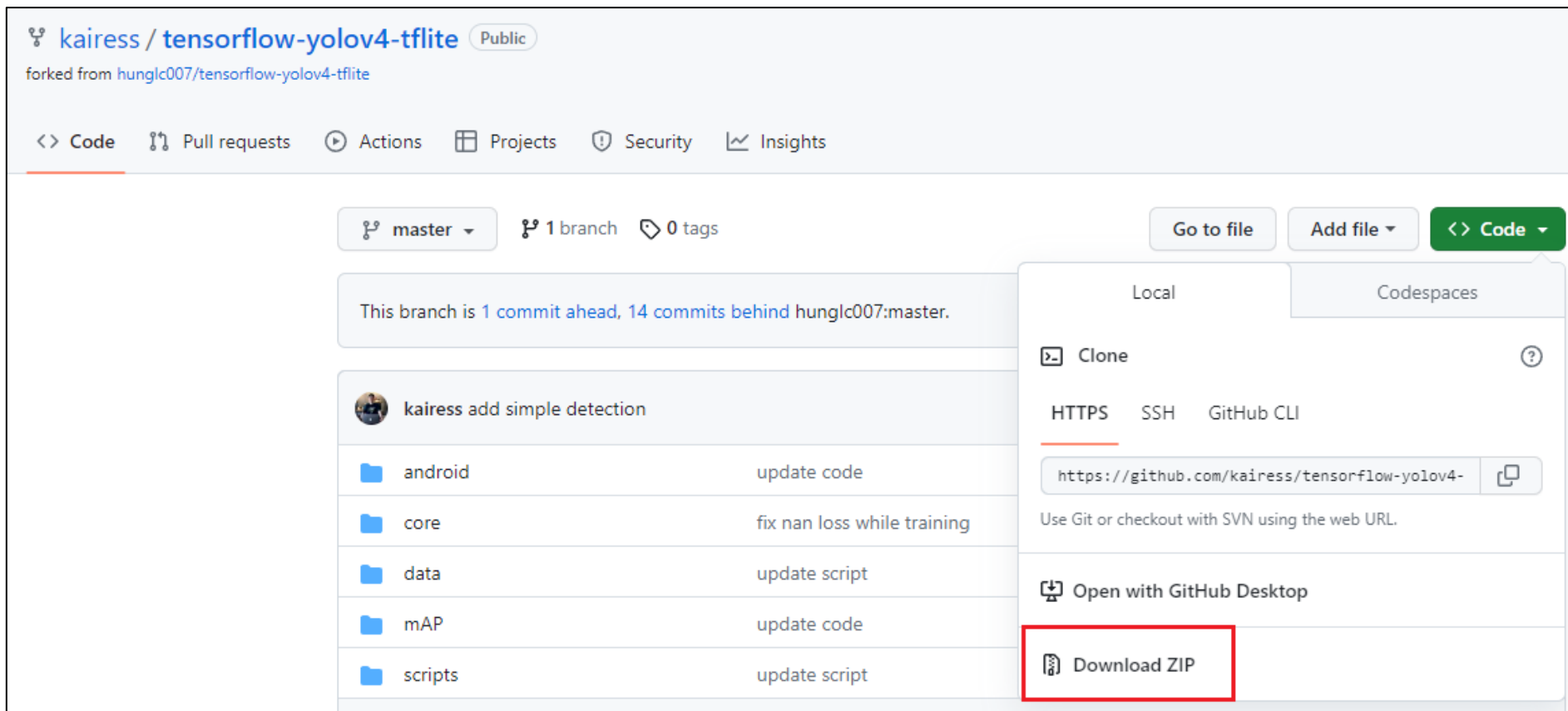
이미지를 그리드라는 단위로 분할

이미지 전체를 신경망에 넣고 특징 추출을 통해 그리드 정보, 신뢰 점수, 분류 클래스 등의 정보로 예측 텐서를 생성

그리드별 예측 정보를 바탕으로 이미지에 테두리 분류 작업을 수행

YOLO

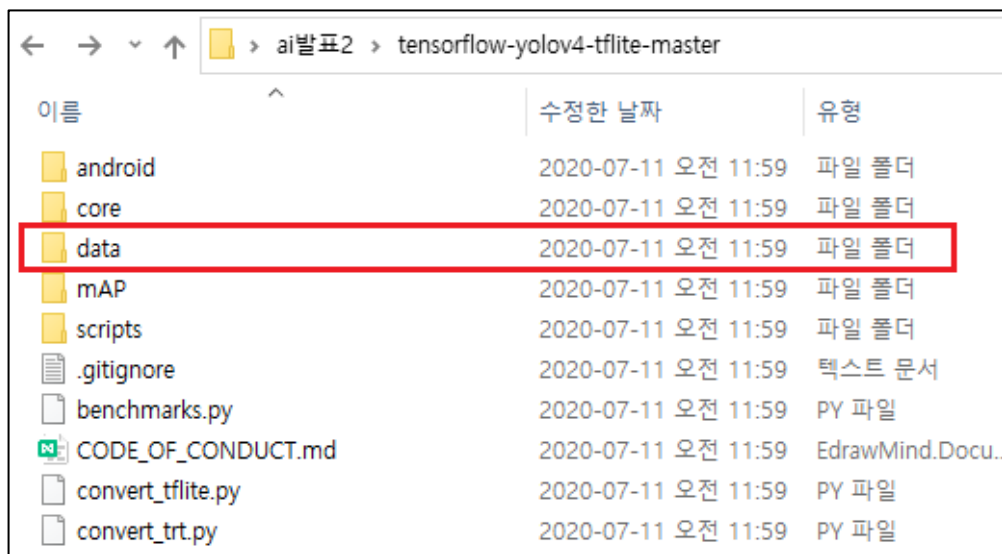
YOLO 사용법 1 : 다운로드



Tensorflow를 이용한 yolov4 다운로드 <https://github.com/kairess/tensorflow-yolov4-tflite>
설치한 yolov4에서 필요한 라이브러리 설치 `pip install easydict`

YOLO

YOLO 사용법 2 : 가중치



이름	수정한 날짜	유형
android	2020-07-11 오전 11:59	파일 폴더
core	2020-07-11 오전 11:59	파일 폴더
data	2020-07-11 오전 11:59	파일 폴더
mAP	2020-07-11 오전 11:59	파일 폴더
scripts	2020-07-11 오전 11:59	파일 폴더
.gitignore	2020-07-11 오전 11:59	텍스트 문서
benchmarks.py	2020-07-11 오전 11:59	PY 파일
CODE_OF_CONDUCT.md	2020-07-11 오전 11:59	EdrawMind.Docu..
convert_tflite.py	2020-07-11 오전 11:59	PY 파일
convert_trt.py	2020-07-11 오전 11:59	PY 파일



이름	수정한 날짜	유형	크기
anchors	2020-07-11 오전 11:59	파일 폴더	
classes	2020-07-11 오전 11:59	파일 폴더	
dataset	2020-07-11 오전 11:59	파일 폴더	
girl.png	2020-07-11 오전 11:59	PNG 파일	649KB
kite.jpg	2020-07-11 오전 11:59	JPG 파일	1,383KB
performance.png	2020-07-11 오전 11:59	PNG 파일	147KB
road.mp4	2020-07-11 오전 11:59	MP4 파일	783KB
yolov3.weights	2023-06-14 오전 10:11	WEIGHTS 파일	242,195KB
yolov4.weights	2023-06-12 오후 4:37	WEIGHTS 파일	251,678KB

COCO dataset을 이용하여 학습시킨 가중치 파일을 다운받은 폴더내 data폴더 안에 넣는다.

<https://github.com/AlexeyAB/darknet/releases> 에서 원하는 버전의 가중치를 선택할 수 있다.

[Yolov3.weights](#) / [Yolov4.weights](#)

```

8 flags.DEFINE_string('weights', './data/yolov4.weights', 'path to weights file')
9 flags.DEFINE_string('output', './checkpoints/yolov4-416', 'path to output')
10 flags.DEFINE_boolean('tiny', False, 'is yolo-tiny or not')
11 flags.DEFINE_integer('input_size', 416, 'define input size of export model')
12 flags.DEFINE_float('score_thres', 0.2, 'define score threshold')
13 flags.DEFINE_string('framework', 'tf', 'define what framework do you want to convert (tf, trt, tflite)')
14 flags.DEFINE_string('model', 'yolov4', 'yolov3 or yolov4')

```

save_model.py 옵션 설정 구조

옵션(설정명)	설정값	설명
weights	./data/yolov4.weights	본인이 다운받은 weight 버전으로 설정
output	./checkpoints/yolov4-416	변환된 모델이 저장될 장소
tiny	False	경량화 버전으로 True 선택 시 tiny 모델로 변환됨
input_size	416	이미지의 사이즈를 의미함(416 * 416)
score_thres	0.2	0.2보다 낮은 정확도를 가지는 박스는 필터링됨
Framework	tf	tf(TensorFlow), trt(TensorRT), tflite(TensorFlow Lite)
Model	yolov4	변환에 사용할 YOLO 버전을 정의

YOLO

YOLO 사용법 3 : 모델 변환

tensorflow-yolov4-tflite-master > checkpoints > yolov4-416			
이름	수정한 날짜	유형	크기
assets	2023-06-15 오후 3:34	파일 폴더	
variables	2023-06-15 오후 3:34	파일 폴더	
keras_metadata.pb	2023-06-15 오후 3:34	PB 파일	1,126KB
saved_model.pb	2023-06-15 오후 3:34	PB 파일	7,592KB

.pb(Protobuf 형식) 파일은 모델 아키텍처, 가중치, 그래프 연산 등의 정보를 담고 있는 TensorFlow에서 사용되는 모델 저장 형식

설치 받은 파일의 core/utils.py 코드를 수정해준다.

수정 전)

```
147     fontScale = 0.5
148     score = out_scores[0][i]
149     class_ind = int(out_classes[0][i])
150     bbox_color = colors[class_ind]
151     bbox_thick = int(0.6 * (image_h + image_w) / 600)
152     c1, c2 = (coor[1], coor[0]), (coor[3], coor[2])
153     cv2.rectangle(image, c1, c2, bbox_color, bbox_thick)
```

수정 후)

```
147     fontScale = 0.5
148     score = out_scores[0][i]
149     class_ind = int(out_classes[0][i])
150     bbox_color = colors[class_ind]
151     bbox_thick = int(0.6 * (image_h + image_w) / 600)
152     c1, c2 = (int(coor[1]), int(coor[0])), (int(coor[3]), int(coor[2]))
153     cv2.rectangle(image, c1, c2, bbox_color, bbox_thick)
```

Line 152:

c1, c2 = (coor[1], coor[0]), (coor[3], coor[2])

-> c1, c2 = (int(coor[1]), int(coor[0])), (int(coor[3]), int(coor[2]))

수정 전)

```
155     if show_label:
156         bbox_mess = '%s: %.2f' % (classes[class_ind], score)
157         t_size = cv2.getTextSize(bbox_mess, 0, fontScale, thickness=bbox_thick // 2)[0]
158         c3 = (c1[0] + t_size[0], c1[1] - t_size[1] - 3)
159         cv2.rectangle(image, c1, (np.float32(c3[0]), np.float32(c3[1])), bbox_color, -1) #filled
160
161         cv2.putText(image, bbox_mess, (c1[0], np.float32(c1[1] - 2)), cv2.FONT_HERSHEY_SIMPLEX,
162                     fontScale, (0, 0, 0), bbox_thick // 2, lineType=cv2.LINE_AA)
163     return image
```

수정 후)

```
155     if show_label:
156         bbox_mess = '%s: %.2f' % (classes[class_ind], score)
157         t_size = cv2.getTextSize(bbox_mess, 0, fontScale, thickness=bbox_thick // 2)[0]
158         c3 = (c1[0] + t_size[0], c1[1] - t_size[1] - 3)
159         cv2.rectangle(image, c1, (int(np.float32(c3[0])), int(np.float32(c3[1]))), bbox_color, -1) #f
160
161         cv2.putText(image, bbox_mess, (c1[0], int(np.float32(c1[1] - 2))), cv2.FONT_HERSHEY_SIMPLEX,
162                     fontScale, (0, 0, 0), bbox_thick // 2, lineType=cv2.LINE_AA)
163     return image
```

Line 159: cv2.rectangle(image, c1, (np.float32(c3[0]), np.float32(c3[1])), bbox_color, -1)
-> cv2.rectangle(image, c1, (int(np.float32(c3[0])), int(np.float32(c3[1]))), bbox_color, -1)

Line 161: cv2.putText(image, bbox_mess, (c1[0], np.float32(c1[1] - 2)), cv2.FONT_HERSHEY_SIMPLEX,
-> cv2.putText(image, bbox_mess, (c1[0], int(np.float32(c1[1] - 2))), cv2.FONT_HERSHEY_SIMPLEX,

```

16 flags.DEFINE_string('framework', 'tf', '(tf, tflite, trt)')
17 flags.DEFINE_string('weights', './checkpoints/yolov4-416', 'path to weights file')
18 flags.DEFINE_integer('size', 416, 'resize images to')
19 flags.DEFINE_boolean('tiny', False, 'yolo or yolo-tiny')
20 flags.DEFINE_string('model', 'yolov4', 'yolov3 or yolov4')
21 flags.DEFINE_string('image', './data/apple.png', 'path to input image')
22 flags.DEFINE_string('output', 'result.png', 'path to output image')
23 flags.DEFINE_float('iou', 0.45, 'iou threshold')
24 flags.DEFINE_float('score', 0.25, 'score threshold')

```

detect.py 옵션 설정 구조

옵션(설정명)	설정값	설명
framework	tf	tf(TensorFlow), trt(TensorRT), tflite(TensorFlow Lite)
weights	./checkpoints/yolov4-416	사용할 모델의 위치(경로)
tiny	False	True 선택 시 tiny 모델(경량화 버전)로 변환됨
model	yolov4	이미지 디텍팅에 사용할 YOLO 버전을 정의
image	./data/kite.jpg	디텍팅에 사용할 이미지 경로
output	result.png	디텍팅된 결과가 들어간 이미지를 저장할 경로
iou	0.45	객체들이 겹치는 박스간의 중복되는 경계값이 (0이면 겹치지 않음 1이면 완전히 겹침) 0.45만큼 가까우면 같은 객체로 지정
score	0.25	0.2보다 낮은 정확도를 가지는 박스는 필터링됨

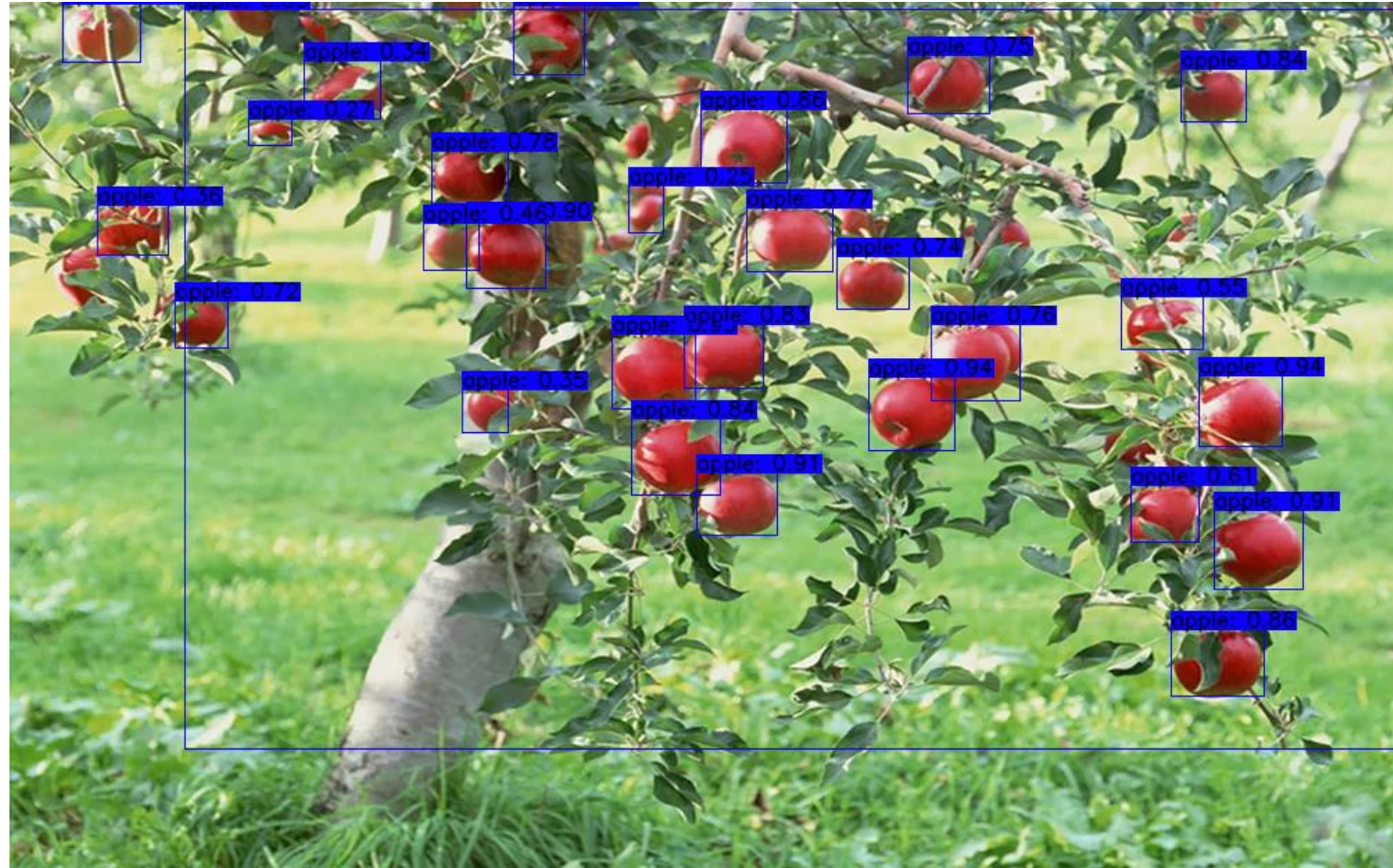
```
flags.DEFINE_string('framework', 'tf', '(tf, tflite, trt)')
flags.DEFINE_string('weights', './checkpoints/yolov4-416/variables/variables', 'path to weights file')
flags.DEFINE_integer('size', 416, 'resize images to')
flags.DEFINE_boolean('tiny', False, 'yolo or yolo-tiny')
flags.DEFINE_string('model', 'yolov4', 'yolov3 or yolov4')
flags.DEFINE_string('video', './data/test2.mp4', 'path to input video')
flags.DEFINE_float('iou', 0.45, 'iou threshold')
flags.DEFINE_float('score', 0.25, 'score threshold')
flags.DEFINE_string('output', './data/result/vidio.mp4', 'path to output video')
flags.DEFINE_string('output_format', 'XVID', 'codec used in VideoWriter when saving video to file')
flags.DEFINE_boolean('dis_cv2_window', False, 'disable cv2 window during the process') # this is good for
```

```
92     print("Video from: ", video_path )
93     # vid = cv2.VideoCapture(video_path)
94     vid = cv2.VideoCapture(0) # 이걸로 사용하면 웹캠 사용가능
```

텐서플로우 버전의 차이로 인한 오류로 detectvideo.py가 아닌
제공된 detectvideo2.py를 detectvideo.py가 위치한 곳으로 옮긴 후
실행하면 영상 디텍팅 가능

Line 93~94 : vid = cv2.VideoCapture(video_path)
-> vid = cv2.VideoCapture(0)
변경 시 웹캠 사용 가능

결과물



END