**교육일지**

**교육 제목 : 비전**

**교육 장소 : YGL C6 강의실**

**교육 일시 : 2021/11/19**

**OpenCV DNN 얼굴검출**

**#**

[**https://github.com/opencv/opencv/tree/master/samples/dnn/face\_detector**](https://github.com/opencv/opencv/tree/master/samples/dnn/face_detector)

**# deploy.prototxt.txt, download-weights.py.txt, opencv\_face\_detector.pbtxt.text 다운로드**

**# Caffe** [**https://raw.githubusercontent.com/opencv/opencv\_3rdparty/dnn\_samples\_face\_detector\_20180205\_fp16/res10\_300x300\_ssd\_iter\_140000\_fp16.caffemodel**](https://raw.githubusercontent.com/opencv/opencv_3rdparty/dnn_samples_face_detector_20180205_fp16/res10_300x300_ssd_iter_140000_fp16.caffemodel)

**# Tensorflow** [**https://raw.githubusercontent.com/opencv/opencv\_3rdparty/dnn\_samples\_face\_detector\_20180220\_uint8/opencv\_face\_detector\_uint8.pb**](https://raw.githubusercontent.com/opencv/opencv_3rdparty/dnn_samples_face_detector_20180220_uint8/opencv_face_detector_uint8.pb)

**## Caffe 학습모델**

**model = 'opencv\_face\_detector/res10\_300x300\_ssd\_iter\_140000\_fp16.caffemodel'**

**config = 'opencv\_face\_detector/deploy.prototxt'**

**## Tensorflow 학습모델**

**# model = 'opencv\_face\_detector/opencv\_face\_detector\_uint8.pb'**

**# config = 'opencv\_face\_detector/opencv\_face\_detector.pbtxt'**

**cap = cv2.VideoCapture(0)**

**if not cap.isOpened():**

**print('Camera open failed!')**

**sys.exit()**

**net = cv2.dnn.readNet(model, config)**

**if net.empty():**

**print('Net open failed!')**

**sys.exit()**

**while True:**

**ret, frame = cap.read()**

**if not ret:**

**break**

**blob = cv2.dnn.blobFromImage(frame, 1, (300, 300), (104, 177, 123))**

**net.setInput(blob)**

**out = net.forward()# out.shape=(1,1, 200, 7)**

**detect = out[0, 0, :, :] ##0, 0, 사용안함**

**(h, w) = frame.shape[:2]**

**for i in range(detect.shape[0]):**

**confidence = detect[i, 2]**

**if confidence > 0.5:**

**x1 = int(detect[i, 3] \* w)**

**y1 = int(detect[i, 4] \* h)**

**x2 = int(detect[i, 5] \* w)**

**y2 = int(detect[i, 6] \* h)**

**cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0))**

**label = f'Face: {confidence:4.2f}'**

**cv2.putText(frame, label, (x1, y1 - 1), cv2.FONT\_HERSHEY\_SIMPLEX, 0.8, (0, 255, 0), 1, cv2.LINE\_AA)**

**cv2.imshow('frame', frame)**

**if cv2.waitKey(1) == 27:**

**break**

**cap.release()**

**cv2.destroyAllWindows()**