**교육일지**

**교육 제목 : 비전**

**교육 장소 : YGL C6 강의실**

**교육 일시 : 2021/11/18**

**Googlenet 영상인식 모델**

**#** [**https://github.com/opencv/opencv/wiki/Deep-Learning-in-OpenCV**](https://github.com/opencv/opencv/wiki/Deep-Learning-in-OpenCV)

**# Caffe Model Zoo : github.com/BVLC/caffe**

**## 모델 파일 : dl.caffe.berkeleyvision.org/bvlc\_googlenet.caffemodel**

**## 설정 파일 : github.com/BVLC/caffe/blob/master/models/bvlc\_googlenet/deploy.prototxt**

**# ONNX Model Zoo : github.com/onnx/models**

**# 모델파일:** [**https://github.com/onnx/models/tree/master/vision/classification/inception\_and\_googlenet/googlenet**](https://github.com/onnx/models/tree/master/vision/classification/inception_and_googlenet/googlenet)

**# 클래스 이름 파일 : github.com/opencv/opencv/blob/4.1.0/samples/data/dnn/**

**# readNet(model, config)**

**# model, config**

**# 실행순서**

**# cv2.dnn.readNet(model, config)-> ret, 객체생성**

**# blobFromImage(image, scalefactor, size, mean, swapRB, crop) -> retval**

**# scalefactor: Multiply by factor**

**# image has BGR ordering and swapRB is true.**

**# 입력 영상 불러오기**

**# filename = 'googlenet/apple2.png'**

**filename = './googlenet/scooter.jpg'**

**# if len(sys.argv) > 1:**

**# filename = sys.argv[1]**

**img = cv2.imread(filename)**

**if img is None:**

**print('Image load failed!')**

**sys.exit()**

**# 네트워크 불러오기**

**# Caffe**

**model = 'googlenet/bvlc\_googlenet.caffemodel'**

**config = 'googlenet/deploy.prototxt'**

**# ONNX**

**# model = 'googlenet/googlenet-9.onnx'**

**# config = ''**

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

**if net.empty():**

**print('Network load failed!')**

**sys.exit()**

**# 클래스 이름 불러오기**

**classNames = []**

**with open('googlenet/classification\_classes\_ILSVRC2012.txt', 'rt') as f:**

**classNames = f.read().rstrip('\n').split('\n')**

**# print(len(classNames))**

**# print(type(classNames))**

**# print(classNames[0])**

**# print(classNames)**

**#추론**

**blob = cv2.dnn.blobFromImage(img, 1, (224, 224), (104, 117, 123),**

**swapRB = False)**

**net.setInput(blob)**

**prob = net.forward()**

**print(prob.shape)**

**# # 추론 결과 확인 & 화면 출력**

**out = prob.flatten() # 1d array**

**classId = np.argmax(out)**

**confidence = out[classId]**

**print(confidence)**

**text = f'{classNames[classId]} ({confidence \* 100:4.2f}%)'**

**cv2.putText(img, text, (10, 30), cv2.FONT\_HERSHEY\_SIMPLEX, 0.8, (0, 0, 255), 1, cv2.LINE\_AA)**

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

**cv2.waitKey()**

**cv2.destroyAllWindows()**