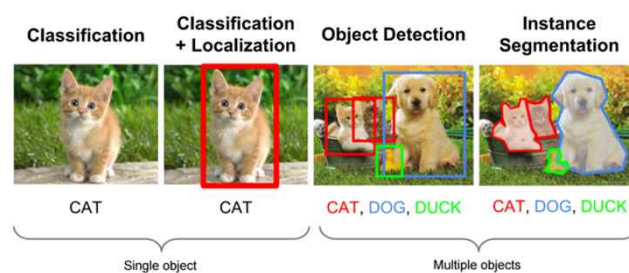


Object Detection in image

31

CNN 의 예

□ 사물 인식(Object Detection)

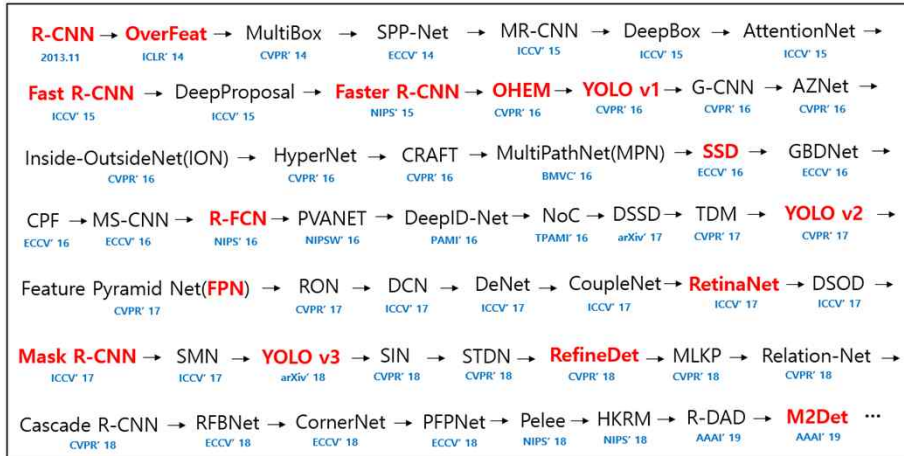


- object detection = **classification(분류)** + **localization**
- 여러가지 object에 대한 classification과 그 object들의 위치정보를 파악하는 것을 동시에 하는 분야

32

CNN 의 예

□ 이미지 내 사물 인식

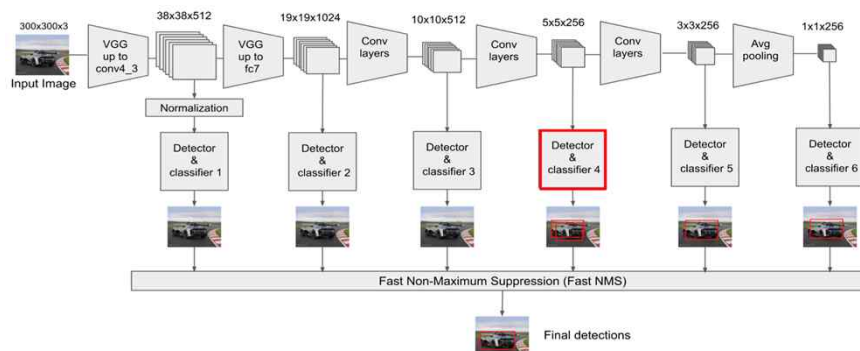


33

CNN 의 예 : SSD

□ SSD(Single Shot MultiBox Detector)

- 2016년 ECCV에서 발표된 논문
- YOLO에서는 이미지를 grid로 나누어서 각 영역에 대해 bounding box를 예측했다면, SSD는 CNN pyramidal feature hierarchy를 이용해 예측
- image feature를 다양한 위치의 layer들에서 추출하여 detector와 classifier를 적용
- SSD는 YOLO v1 보다 속도나 정확도 측면에서 더 높은 성능을 보여줌



34

예제 : 영상 내 사물 인식(ssd)

학습모델과 라벨 설정

```
model = cv2.dnn.readNetFromCaffe(
    '../data/MobileNetSSD_deploy.prototxt',
    '../data/MobileNetSSD_deploy.caffemodel')

CONF_THR = 0.3

LABELS = {1: 'aeroplane', 2: 'bicycle', 3: 'bird', 4: 'boat',
          5: 'bottle', 6: 'bus', 7: 'car', 8: 'cat', 9: 'chair',
          10: 'cow', 11: 'diningtable', 12: 'dog', 13: 'horse',
          14: 'motorbike', 15: 'person', 16: 'pottedplant',
          17: 'sheep', 18: 'sofa', 19: 'train', 20: 'tvmonitor'}
```

35

예제 : 영상 내

영상 내 검출한 사물의 정보 : RET, 결과 라벨, 결과값, 사물의 위치(x0, y0, x1, y1)

[[[[0.	7.	0.999785	0.41983896	0.7232013	0.6157767	1.0015314]
[0.	7.	0.9897425	0.2770408	0.25776207	0.39213035	0.41807252]
[0.	7.	0.90626186	0.84620607	0.79742783	0.9948509	0.99628216]
]	0.	7.	0.2964747	0.30139327	0.23780742	0.41048676	0.36649063]]]]

주어진 이미지 (frame)를 학습모델 (model)에 테스트하여 결과 (output)을 사각형으로 그리고 라벨을 적음

```
h, w = frame.shape[0:2]
blob = cv2.dnn.blobFromImage(frame, 1/127.5, (300*w//h, 300), (127.5, 127.5, 127.5), False)
model.setInput(blob)
output = model.forward()
```

blobFromImage(입력 영상, scaleFactor, Size, mean, SwarR&B)

```
for i in range(output.shape[2]):
    conf = output[0,0,i,2] # i번째 사물 정보의 결과값
    if conf > CONF_THR: # 임계값(CONF_THR) 이상인 경우만
        label = output[0,0,i,1]
        x0,y0,x1,y1 = (output[0,0,i,3:7] * [w,h,w,h]).astype(int)
        cv2.rectangle(frame, (x0,y0), (x1,y1), (0,255,0), 2)
        cv2.putText(frame, '{}: {:.2f}'.format(LABELS[label], conf),
                    (x0,y0), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,255,0), 2)
```

36

예제 : 영상 내 사물 인식(ssd)

주어진 이미지 (frame)의 크기는 ssd 실행 속도에 영향을 미치므로, 크기를 줄여서 실행

```
img = cv2.resize(frame, dsize=(int(frame.shape[1]/2), int(frame.shape[0]/2)))
```

```
ih = int(h/2)
```

```
iw = int(w/2)
```

```
for i in range(output.shape[2]):
```

```
    conf = output[0,0,i,2]
```

```
    if conf > CONF_THR:
```

```
        label = output[0,0,i,1]
```

```
        x0,y0,x1,y1 = (output[0,0,i,3:7] * [iw,ih,iw,ih]).astype(int)
```

```
        cv2.rectangle(img, (x0,y0), (x1,y1), (0,255,0), 2)
```

```
        cv2.putText(img, '{}: {:.2f}'.format(LABELS[label], conf),
```

```
                    (x0,y0), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,255,0), 2)
```

```
c += 1
```

```
if c == 100:
```

```
    cv2.imwrite('./09results/ch5_car_detections.png', img)
```

```
cv2.imshow('frame', img)
```

37

CNN의 예: YOLO

FAST!!!

□ YOLO(You Only Look Once)

□ CVPR 2016

- 현재 최고 수준의 사물 인식 알고리즘으로 R-CNN 종류를 능가(2018 기준)

- 이미지 내의 물체의 위치(bounding box)와 물체의 종류 분류(class probability)를 하나의 문제로 간주
 - 하나의 CNN을 통해 여러 bounding box에 대한 class probability를 계산하는 방식



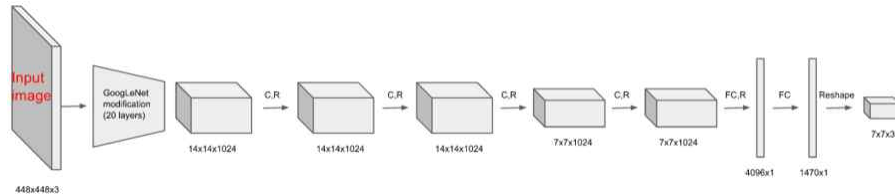
38

CNN의 예: YOLO

□ YOLO(You Only Look Once)

□ 구조

- GoogleLeNet을 약간 변형시켜서 특징 추출기로 사용
- 이후 컨볼루션 레이어 4회, 풀 커넥션 레이어 2번 하고 사이즈를 7x7x30으로 조정



□ 마지막 특징 데이터 7x7x30이 예측 결과

- 7x7 : 이미지를 7x7로 분할한 각 cell 정보 저장
- 30 : 각 cell에 대한 특징정보 저장

□ yolo 사이트에서는 다양한 버전의 미리 학습된 yolo 신경망들을 제공

- <https://pjreddie.com/darknet/yolo/>

39

YOLO v3

<https://pjreddie.com/darknet/yolo/>

Performance on the COCO Dataset

Model	Train	Test	mAP	FLOPS	FPS	Cfg	Weights
SSD300	COCO trainval	test-dev	41.2	-	46	link	link
SSD500	COCO trainval	test-dev	46.5	-	19	link	link
YOLOv2 608x608	COCO trainval	test-dev	48.1	62.94 Bn	40	cfg	weights
Tiny YOLO	COCO trainval	test-dev	23.7	5.41 Bn	244	cfg	weights
SSD321	COCO trainval	test-dev	45.4	-	16	link	link
DSSD321	COCO trainval	test-dev	46.1	-	12	link	link
R-FCN	COCO trainval	test-dev	51.9	-	12	link	link
SSD513	COCO trainval	test-dev	50.4	-	8	link	link
DSSD513	COCO trainval	test-dev	53.3	-	6	link	link
FPN FRCN	COCO trainval	test-dev	59.1	-	6	link	link
Retinanet-50-500	COCO trainval	test-dev	50.9	-	14	link	link
Retinanet-101-500	COCO trainval	test-dev	53.1	-	11	link	link
Retinanet-101-800	COCO trainval	test-dev	57.5	-	5	link	link
YOLOv3-320	COCO trainval	test-dev	51.5	38.97 Bn	45	cfg	weights
YOLOv3-416	COCO trainval	test-dev	55.3	65.86 Bn	35	cfg	weights
YOLOv3-608	COCO trainval	test-dev	57.9	140.69 Bn	20	cfg	weights
YOLOv3-tiny	COCO trainval	test-dev	33.1	5.56 Bn	220	cfg	weights
YOLOv3-spp	COCO trainval	test-dev	60.6	141.45 Bn	20	cfg	weights

[coco.names](https://github.com/pjreddie/darknet/tree/master/data)

<https://github.com/pjreddie/darknet/tree/master/data>

40

예제 : 영상 내 사물 인식(yolo v3)

```
# Load Yolo
model = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg")
classes = []
with open("coco.names", "r") as f: # yolo v2
    classes = [line.strip() for line in f.readlines()]
    layer_names = model.getLayerNames()
    output_layers = [layer_names[i[0] - 1] for i in model.getUnconnectedOutLayers()]
    colors = np.random.uniform(0, 255, size=(len(classes), 3))

video = cv2.VideoCapture('./traffic.mp4')

# Detecting objects
c = 0
while True:
    ret, frame = video.read()
    if not ret: break

    blob = cv2.dnn.blobFromImage(frame, 0.00392, (416, 416), (0, 0, 0), True, crop=False)
    model.setInput(blob)
    output = model.forward(output_layers)

    img = cv2.resize(frame, dsize=(int(frame.shape[1] / 2), int(frame.shape[0] / 2)))
    height, width = img.shape[0:2]

    # Showing informations on the screen

    cv2.imshow('frame', img)
```

41

예제 : 영상 내 사물

영상 내 검출한 사물의 정보 : x, y, w, h, class에 대한 인식률

```
[[0.04961462 0.03220425 0.21853049 ... 0. 0. 0. ]
 [0.04820277 0.03173435 0.243103 ... 0. 0. 0. ]
 [0.04572324 0.02620311 1.009227 ... 0. 0. 0. ]]
```

```
# Showing informations on the screen
class_ids = []
confidences = []
boxes = []
for out in output:
    for detection in out:
        scores = detection[5:]
        class_id = np.argmax(scores)
        conf = scores[class_id]
        if conf > 0.5:
            # Object detected
            center_x = int(detection[0] * width)
            center_y = int(detection[1] * height)
            w = int(detection[2] * width)
            h = int(detection[3] * height)
            # Rectangle coordinates
            x = int(center_x - w / 2)
            y = int(center_y - h / 2)
            boxes.append([x, y, w, h])
            confidences.append(float(conf))
            class_ids.append(class_id)
indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)

for i in range(len(boxes)):
    if i in indexes:
        x, y, w, h = boxes[i]
        label = str(classes[class_ids[i]])
        color = colors[i]
        conf = confidences[i]
        cv2.rectangle(img, (x, y), (x + w, y + h), color, 2)
        cv2.putText(img, '{}: {:.2f}'.format(label, conf),
                    (x, y), cv2.FONT_HERSHEY_SIMPLEX, 1, color, 2)
```

42

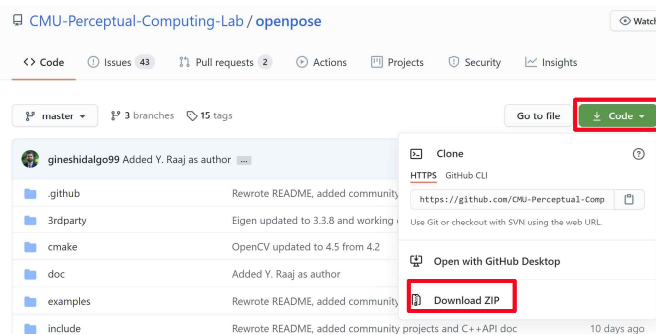
Pose Estimation

43

딥러닝을 이용한 자세 인식

□ OpenPose 다운받기

- <https://github.com/CMU-Perceptual-Computing-Lab/openpose>
- [Code] - [Clone or download] - [Download ZIP]



44

딥러닝을 이용한 자세 인식

■ 압축 해제 후 [models] - "getModels" 파일 실행

- windows 환경에서 실행하는 것이고, 그 외 Linux환경에서는 getModels.sh를 실행시킵니다.
- 파일이 제대로 실행되면 각 모델을 다운로드 해줍니다.

openpose-master > openpose-master > models >

이름	유형	크기
cameraParameters	파일 폴더	
face	파일 폴더	
hand	파일 폴더	
pose	파일 폴더	
getModels	Windows 배치 파...	2KB
getModels.sh	SH 파일	2KB

- 아래 두 파일을 파이썬 코드가 있는 폴더에 복사

openpose-master > models > pose > mpi

이름	유형	크기
pose_deploy_linevec_prototxt	PROTOTXT 파일	46KB
pose_deploy_linevec_faster_4_stages.prototxt	PROTOTXT 파일	32KB
pose_iter_160000.caffemodel	CAFFEModel 파일	201,124KB

```
----- Downloading body pose (COCO and MPI), face and hand models -----
Body (BODY_25)
--2021-02-03 10:35:40-- http://posefs1.perception.cs.cmu.edu/OpenPose/models/pose/body_25/pose_iter_584000.caffemodel
Resolving posefs1.perception.cs.cmu.edu (posefs1.perception.cs.cmu.edu)... 128.2.176.37
Connecting to posefs1.perception.cs.cmu.edu (posefs1.perception.cs.cmu.edu)|128.2.176.37|:80...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 104715850 (100M) [text/plain]
Saving to: 'pose/body_25/pose_iter_584000.caffemodel'

pose_iter_584000.caffem 100%[=====] 99.86M 2.93MB/s in 57s

2021-02-03 10:36:37 (1.76 MB/s) - 'pose/body_25/pose_iter_584000.caffemodel' saved [104715850/104715850]

Body (COCO)
--2021-02-03 10:36:37-- http://posefs1.perception.cs.cmu.edu/OpenPose/models/pose/coco/pose_iter_440000.caffemodel
Resolving posefs1.perception.cs.cmu.edu (posefs1.perception.cs.cmu.edu)... 128.2.176.37
Connecting to posefs1.perception.cs.cmu.edu (posefs1.perception.cs.cmu.edu)|128.2.176.37|:80...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 209274056 (200M) [text/plain]
Saving to: 'pose/coco/pose_iter_440000.caffemodel'

pose_iter_440000.caffem 49%[=====] 198.05M 1.59MB/s eta 69s
```

다른 모델을 사용 가능

45

예제 : 딥러닝을 이용한 자세 인식(OpenPose)

```
# MPI에서 각 파트 번호, 선으로 연결될 POSE_PAIRS
BODY_PARTS = {"Head": 0, "Neck": 1, "RShoulder": 2, "RElbow": 3, "RWrist": 4, "LShoulder": 5,
              "LElbow": 6, "LWrist": 7, "RHip": 8, "RKnee": 9, "RAnkle": 10, "LHip": 11, "LKnee": 12,
              "LAnkle": 13, "Chest": 14, "Background": 15}

POSE_PAIRS = [{"Head", "Neck"}, {"Neck", "RShoulder"}, {"RShoulder", "RElbow"}, {"RElbow", "RWrist"},
              {"Neck", "LShoulder"}, {"LShoulder", "LElbow"}, {"LElbow", "LWrist"}, {"Neck", "Chest"},
              {"Chest", "RHip"}, {"RHip", "RKnee"}, {"RKnee", "RAnkle"}, {"Chest", "LHip"},
              {"LHip", "LKnee"}, {"LKnee", "LAnkle"}]

# 각 파일 path
protoFile = "pose_deploy_linevec_faster_4_stages.prototxt"
weightsFile = "pose_iter_160000.caffemodel"
net = cv2.dnn.readNetFromCaffe(protoFile, weightsFile)

# 이미지 읽어오기
image = cv2.imread("full-body3.jpg")
imageHeight, imageWidth, _ = image.shape

# network에 넣기위해 전처리
inpBlob = cv2.dnn.blobFromImage(image, 1.0 / 255, (imageWidth, imageHeight), (0, 0, 0), swapRB=False,
crop=False)

# network에 넣어주기
net.setInput(inpBlob)

# 결과 받아오기
output = net.forward()
```

46

예제 : 딥러닝을 이용한 자세 인식(OpenPose)

```
# output.shape[0] = 이미지 ID, [1] = 출력 맵의 높이, [2] = 너비
H = output.shape[2]
W = output.shape[3]
print("이미지 ID : ", len(output[0]), ", H : ", output.shape[2], ", W : ", output.shape[3]) # 이미지 ID

# 키포인트 검출시 이미지에 그려줌
points = []
for i in range(0, 15):
    probMap = output[0, i, :, :] # 해당 신체부위 신뢰도 얻음.
    minVal, prob, minLoc, point = cv2.minMaxLoc(probMap) # global 최대값 찾기

    # 원래 이미지에 맞게 점 위치 변경
    x = (imageWidth * point[0]) / W
    y = (imageHeight * point[1]) / H

    # 키포인트 검출한 결과가 0.1보다 크면(검출한곳이 위 BODY_PARTS랑 맞는 부위면) points에 추가, 검출했는데 부위가 없으면 None으로
    if prob > 0.1:
        cv2.circle(image, (int(x), int(y)), 3, (0, 255, 255), thickness=-1, lineType=cv2.FILLED)
        cv2.putText(image, "{}".format(i), (int(x), int(y)), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 1,
lineType=cv2.LINE_AA)
        points.append((int(x), int(y)))
    else:
        points.append(None)

cv2.imshow("Output-Keypoints", image)
cv2.waitKey(0)
```

47

예제 : 딥러닝을 이용한 자세 인식(OpenPose)

```
# 이미지 복사
imageCopy = image

# 각 POSE_PAIRS별로 선 그려줌 (머리 - 목, 목 - 왼쪽어깨, ...)
for pair in POSE_PAIRS:
    partA = pair[0] # Head
    partA = BODY_PARTS[partA] # 0
    partB = pair[1] # Neck
    partB = BODY_PARTS[partB] # 1

    if points[partA] and points[partB]:
        cv2.line(imageCopy, points[partA], points[partB], (0, 255, 0), 2) # partA와 partB 연결

cv2.imshow("Output-Keypoints", imageCopy)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

48