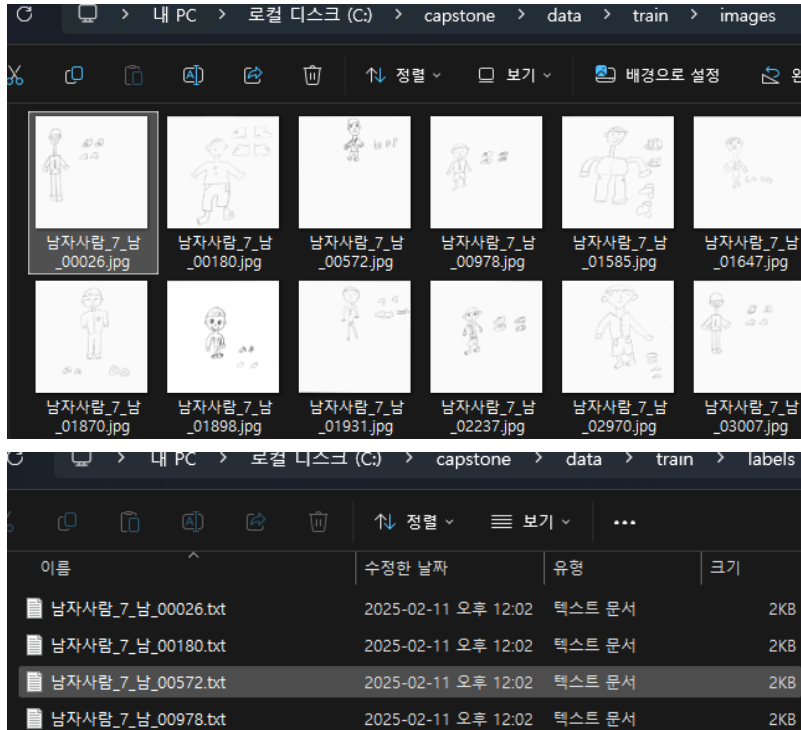


0213 5번까지 진행 내용 정리

1. yolo 모델 학습 완료

1. image data & label data



2. label data를 yolo 형식으로 전처리

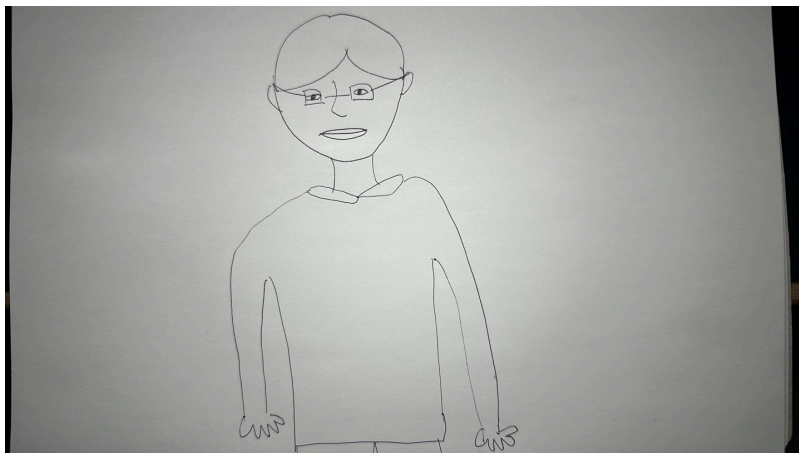
```
1 import json
2 import os
3 import glob
4
5 # 입력 JSON 파일들이 저장된 폴더
6 input_dir = "json"
7
8 # 출력 YOLO 텍스트 파일을 저장할 폴더 (labels 폴더)
9 output_dir = "txt"
10 os.makedirs(output_dir, exist_ok=True)
11
12 # 클래스 번호 매핑 (필요에 따라 수정)
13 class_mapping = {
14     "사람전체": 0,
15     "머리": 1,
16     "얼굴": 2,
17     "눈": 3,
18     "코": 4,
19     "입": 5,
20     "귀": 6,
21     "머리카락": 7,
22     "목": 8,
23     "상체": 9,
24     "팔": 10,
25     "손": 11,
26     "다리": 12,
27     "발": 13,
28     "단추": 14,
29     "주머니": 15,
30     "운동화": 16,
31     "남자구두": 17,
32     "여자구두": 18
33 }
34
```

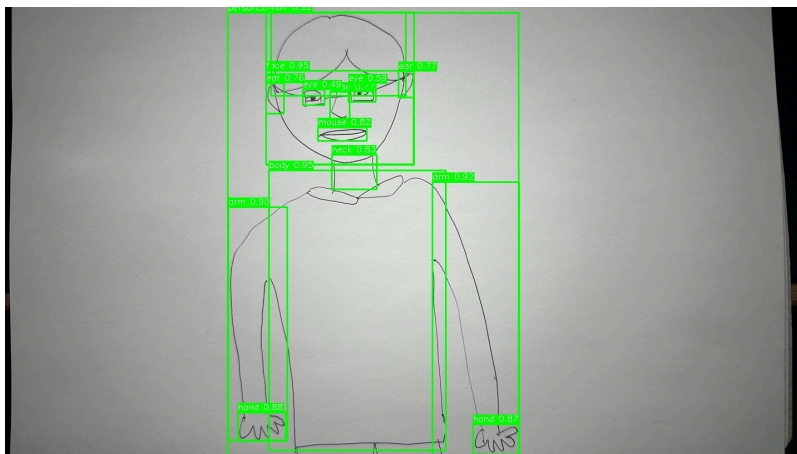
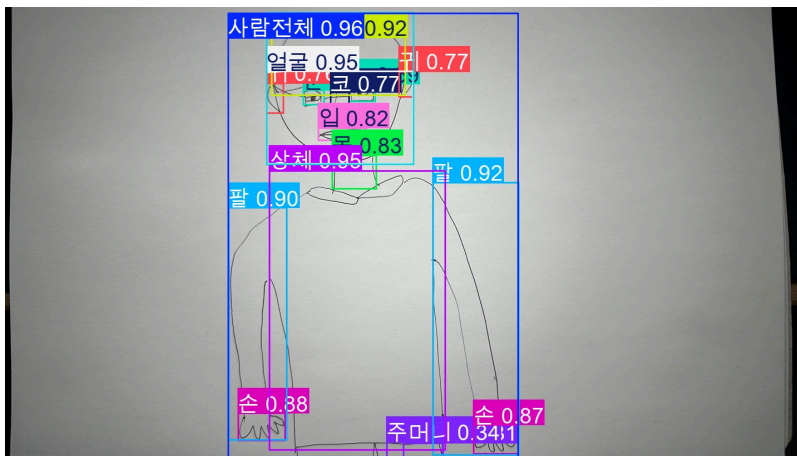
3. 학습 진행

yolo task=detect mode=train data=dataset.yaml model=yolov8n.pt epochs=50 device=0



4. 직접 그린 사람그림을 폰으로 촬영하여 테스트 진행



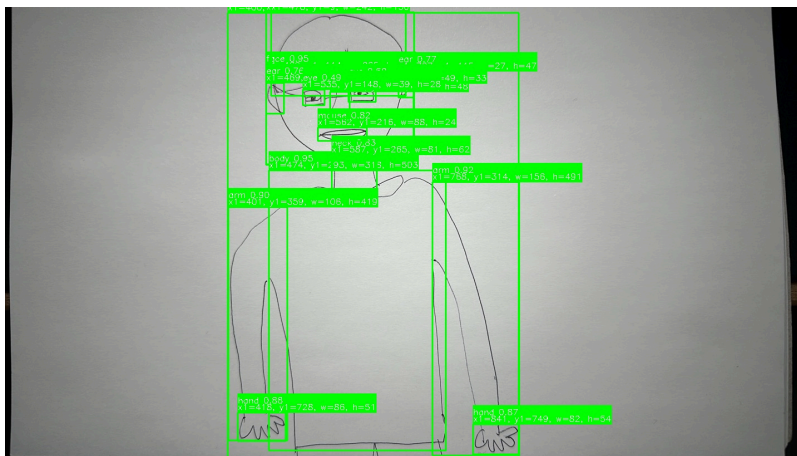


5. 위치, 크기 정보 얻어오기

```

16
17 # 5. 클래스 이름 리스트(클래스 순서대로)
18 class_names = [
19     'person_all', 'head', 'face', 'eye', 'nose', 'mouse', 'ear', 'hair',
20     'neck', 'body', 'arm', 'hand', 'leg', 'foot', 'button', 'pocket', 'sneakers', 'm
21 ]
22
23 # 6. 추론 결과를 이용해 바운딩 박스와 라벨 시각화
24 # results는 이미지당 하나의 결과 객체를 포함하는 리스트
25 for result in results:
26     boxes = result.boxes # 탐지된 모든 객체의 바운딩 박스 정보
27     for box in boxes:
28         # 좌표, 신뢰도, 클래스 정보 추출(tensor -> numpy로 변환)
29         xyxy = box.xyxy.cpu().numpy()[0] # [x1, y1, x2, y2]
30         conf = box.conf.cpu().numpy()[0] # 신뢰도
31         cls_id = int(box.cls.cpu().numpy()[0]) # 클래스 번호
32
33         # 좌표를 정수형으로 변환(OpenCV 그리기 위해)
34         x1, y1, x2, y2 = map(int, xyxy)
35
36         # 클래스 이름 결정 (클래스 번호가 class_names 범위 내에 있을 때)
37         label = class_names[cls_id] if cls_id < len(class_names) else str(cls_id)
38
39         # 바운딩 박스 그리기 (녹색 테두리, 두께 2)
40         cv2.rectangle(image, (x1, y1), (x2, y2), color=(0, 255, 0), thickness=2)
41
42         # 라벨 및 신뢰도 텍스트 생성
43         text = f"{label} {conf:.2f}"
44
45         # 텍스트 배경(선택 사항): 텍스트가 잘 보이도록 배경 박스 그리기
46         (text_width, text_height), baseline = cv2.getTextSize(text, cv2.FONT_HERSHEY
47         cv2.rectangle(image, (x1, y1 - text_height - baseline), (x1 + text_width, y1
48
49         # 텍스트 그리기(하얀색 크기)

```



```
image 1/1 C:\Capstone\yolo\test.jpg: 384x640 1 사람전체, 1 머리, 1 얼굴, 2 눈s, 1 코, 1 입, 2 귀s, 1 머리카락, 1 목, 1 상체, 2 팔s, 2 손s, 22.5ms
Speed: 3.8ms preprocess, 22.5ms Inference, 49.6ms postprocess per Image at shape (1, 3, 384, 640)
클래스: person_all, 신뢰도: 0.96
좌표: x1=400, y1=10, x2=923, y2=810
크기: w=523, h=800

클래스: head, 신뢰도: 0.96
좌표: x1=469, y1=0, x2=735, y2=283
크기: w=266, h=274

클래스: face, 신뢰도: 0.95
좌표: x1=469, y1=114, x2=734, y2=282
크기: w=265, h=168

클래스: body, 신뢰도: 0.95
좌표: x1=474, y1=293, x2=792, y2=796
크기: w=318, h=503

클래스: hair, 신뢰도: 0.92
좌표: x1=478, y1=0, x2=720, y2=159
크기: w=242, h=150

클래스: arm, 신뢰도: 0.92
좌표: x1=768, y1=314, x2=924, y2=885
크기: w=156, h=491

클래스: arm, 신뢰도: 0.99
좌표: x1=481, y1=359, x2=587, y2=778
크기: w=106, h=419

클래스: hand, 신뢰도: 0.88
좌표: x1=418, y1=728, x2=504, y2=779
크기: w=86, h=51

클래스: hand, 신뢰도: 0.87
좌표: x1=841, y1=749, x2=923, y2=883
크기: w=82, h=54

클래스: neck, 신뢰도: 0.83
좌표: x1=587, y1=265, x2=668, y2=327
크기: w=81, h=62

클래스: mouse, 신뢰도: 0.82
좌표: x1=562, y1=216, x2=658, y2=240
크기: w=88, h=24

클래스: ear, 신뢰도: 0.77
좌표: x1=787, y1=115, x2=734, y2=162
크기: w=27, h=47

클래스: nose, 신뢰도: 0.77
좌표: x1=584, y1=152, x2=619, y2=200
크기: w=35, h=48

클래스: ear, 신뢰도: 0.76
좌표: x1=469, y1=136, x2=501, y2=191
크기: w=32, h=55

클래스: eye, 신뢰도: 0.59
좌표: x1=617, y1=137, x2=666, y2=170
크기: w=49, h=33

클래스: eye, 신뢰도: 0.49
좌표: x1=535, y1=148, x2=574, y2=176
크기: w=39, h=28
```

2. 개선할 점

- [어깨, 턱]은 라벨 데이터로 따로 분류하지 않고 있어서 학습하지 않았음(그 외 부위는 모두 정상 학습됨)
 - 그 외 부위: 사람전체, 머리, 얼굴, 눈, 코, 입, 귀, 머리카락, 목, 상체, 팔, 손, ...
 - 이 부분은 생략하거나 따로 학습시킬 방법을 찾아야 할 듯.

3. 앞으로 계획

- 일차적으로는 파악된 위치, 크기로 [형식, 내용적 분석]에서 바로 크기가 작다, 크다 판단할 수 있고 위치가 가장자리인지, 중심인지 등 바로 적용할 수 있는 것에 먼저 적용해서 점수를 채점할 수 있는 곳에 바로 적용 할 예정. + 머리 눈 코 입 귀 머리카락 목 팔 손 등의 생략 여부
- 그 후에 적용이 안되는 정보를 얻기 위해서 나머지 부분에서 cv를 활용한 정보 추출을 진행할 예정.
 - cv 적용 방법 연구 -> 6번 진행

- 위를 통합한 점수 채점 및 시각화 -> 7,8번 진행