

# 객체 인식 파트 - 13

(주)인피닉스 - 강호용 연구원

Infinyx





# 과제 리뷰

---

## # 주어진 데이터 coco 양식을 이용하여 yolo 포맷으로 변경하기

```
import json
import os
import shutil

# yolo dataset create folder
os.makedirs("./glass_dataset/train/images/", exist_ok=True)
os.makedirs("./glass_dataset/train/labels/", exist_ok=True)
os.makedirs("./glass_dataset/val/images/", exist_ok=True)
os.makedirs("./glass_dataset/val/labels/", exist_ok=True)

# image path
image_path = "./test"

# COCO annotation file path
coco_annotation_path = './test/_annotations.coco.json'

# YOLO format annotation save folder
yolo_annotation_folder = './glass_dataset/val/labels/'

# YOLO format image save folder
yolo_image_folder = './glass_dataset/val/images/'

# COCO class names (modify according to your dataset)
coco_classes = ['glass']
yolo_classes = {'glass':0}

# Load COCO annotations
with open(coco_annotation_path, 'r') as f:
    coco_annotations = json.load(f)
```

```
# Convert COCO to YOLO format
for annotation in coco_annotations['annotations']:
    image_id = annotation['image_id']
    image_info = coco_annotations['images'][image_id - 1]

    image_filename = image_info['file_name']
    image_basename = os.path.splitext(image_filename)[0]

    width = image_info['width']
    height = image_info['height']

    category_id = annotation['category_id']
    class_name = coco_classes[category_id - 1]
    yolo_class_name = yolo_classes[class_name]

    bbox = annotation['bbox']
    x, y, bbox_width, bbox_height = bbox
    center_x = (x + bbox_width / 2) / width
    center_y = (y + bbox_height / 2) / height
    norm_width = bbox_width / width
    norm_height = bbox_height / height

    yolo_annotation_path = os.path.join(yolo_annotation_folder, f"{image_basename}.txt")

    with open(yolo_annotation_path, 'a') as f:
        annotation_line = f"{yolo_class_name} {center_x:.6f} {center_y:.6f} {norm_width:.6f} {norm_height:.6f}\n"
        f.write(annotation_line)

# Copy image to YOLO image folder
source_image_path = os.path.join(image_path, image_filename)
destination_image_path = os.path.join(yolo_image_folder, image_filename)
shutil.copy(source_image_path, destination_image_path)

print("Annotations and images converted and saved in YOLO format.")
```

## # 주어진 데이터 coco 양식을 이용하여 yolo 포맷으로 변경하기



VID\_20230723\_101947\_0036\_MP4-3.jpg.rf.76b3ddcff77be824...



VID\_20230723\_101947\_0036\_MP4-3.jpg.rf.142eb7f96ae20587...



VID\_20230723\_101947\_0036\_MP4-3.jpg.rf.7490522aca925615...



VID\_20230723\_101947\_0036\_MP4-4.jpg.rf.02ea11307e9e3411...



VID\_20230723\_101947\_0036\_MP4-4.jpg.rf.945d534064096c1...



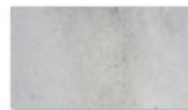
VID\_20230723\_101947\_0036\_MP4-7.jpg.rf.f44ba8fe50af20d8e...



VID\_20230723\_101947\_0036\_MP4-8.jpg.rf.8ee04015aa23df46...



VID\_20230723\_101947\_0036\_MP4-8.jpg.rf.45b715e136679df1...



VID\_20230723\_101947\_0036\_MP4-8.jpg.rf.a6fd65bb2860474...



VID\_20230723\_101947\_0036\_MP4-9.jpg.rf.6bd1364638be239...

VID\_20230723\_101947\_0036\_MP4-3.jpg.rf.76b3ddcff77be824...

VID\_20230723\_101947\_0036\_MP4-3.jpg.rf.142eb7f96ae20587...

VID\_20230723\_101947\_0036\_MP4-3.jpg.rf.7490522aca925615...

VID\_20230723\_101947\_0036\_MP4-4.jpg.rf.02ea11307e9e3411...

VID\_20230723\_101947\_0036\_MP4-4.jpg.rf.945d534064096c1...

VID\_20230723\_101947\_0036\_MP4-4.jpg.rf.945d534064096c1...

VID\_20230723\_101947\_0036\_MP4-6.jpg.rf.8ee04015aa23df46...

VID\_20230723\_101947\_0036\_MP4-6.jpg.rf.8ee04015aa23df46...

VID\_20230723\_101947\_0036\_MP4-6.jpg.rf.8ee04015aa23df46...

VID\_20230723\_101947\_0036\_MP4-7.jpg.rf.f44ba8fe50af20d8e...

VID\_20230723\_101947\_0036\_MP4-7.jpg.rf.f44ba8fe50af20d8e...

VID\_20230723\_101947\_0036\_MP4-7.jpg.rf.f44ba8fe50af20d8e...

```
0 0.847290 0.640625 0.020752 0.039931
0 0.657349 0.910807 0.008057 0.016059
0 0.945923 0.764974 0.022705 0.040365
0 0.753052 0.834852 0.019775 0.044705
```

## # CVAT 양식으로 변경하기 그 후 CVAT Tool 적용하여 라벨 체크

```
import os
import glob
import cv2
from ultralytics import YOLO
import xml.etree.ElementTree as ET

model = YOLO('./runs/detect/train/weights/best.pt') # load a custom model
data_path = "./ultralytics/cfg/yolo_dataset/val"
data_path_list = glob.glob(os.path.join(data_path, "images", "*.png"))

tree = ET.ElementTree()
root = ET.Element("annotations")
seen = 0
xml_save_path = "./test.xml"

for path in data_path_list:
    names = model.names
    result = model.predict(path, save=False, imgsz=640, conf=0.5)
    boxes = result[0].boxes
    box = boxes
    box_xyxy = box.xyxy
    cls = box.cls
    image = cv2.imread(path)
    img_height, img_width, _ = image.shape

    file_name = os.path.basename(path)

    # xml fix code
    xml_frame = ET.SubElement(root, tag="image", id="%d" % seen, name=file_name, width="%d" % img_width,
                              height="%d" % img_height)
    #
```

# CVAT 양식으로 변경하기 그 후 CVAT Tool 적용하여 라벨 체크

```
for bbox , cls_number in zip(box_xyxy, cls) :
    cls_number = int(cls_number.item())
    cls_name = names[cls_number]
    x1 = int(bbox[0].item())
    y1 = int(bbox[1].item())
    x2 = int(bbox[2].item())
    y2 = int(bbox[3].item())
    ET.SubElement(xml_frame, tag="box", label=str(cls_name), occluded="0", source="manual", xtl="%.3f" % x1, ytl="%.3f" % y1, xbr="%.3f" % x2, ybr="%.3f" % y2, z_order="0")

seen += 1
tree._setroot(root)
tree.write(xml_save_path, encoding="utf-8")
```



## # CVAT 양식으로 변경하기 그 후 CVAT Tool 적용하여 라벨 체크

```
<annotations>
  <image id="0" name="syn_03522.png" width="1920" height="1040">
    <box label="28" occluded="0" source="manual" xtl="1028.000" ytl="165.000" xbr="1304.000" ybr="443.000" z_order="0"/>
    <box label="8" occluded="0" source="manual" xtl="814.000" ytl="492.000" xbr="1108.000" ybr="791.000" z_order="0"/>
    <box label="5" occluded="0" source="manual" xtl="380.000" ytl="441.000" xbr="666.000" ybr="747.000" z_order="0"/>
  </image>
  <image id="1" name="syn_03714.png" width="1920" height="1040">
    <box label="25" occluded="0" source="manual" xtl="1052.000" ytl="305.000" xbr="1364.000" ybr="629.000" z_order="0"/>
    <box label="29" occluded="0" source="manual" xtl="303.000" ytl="309.000" xbr="542.000" ybr="579.000" z_order="0"/>
  </image>
  <image id="2" name="syn_04367.png" width="1920" height="1040">
    <box label="29" occluded="0" source="manual" xtl="348.000" ytl="475.000" xbr="608.000" ybr="772.000" z_order="0"/>
  </image>
  <image id="3" name="syn_04003.png" width="1920" height="1040">
    <box label="25" occluded="0" source="manual" xtl="757.000" ytl="426.000" xbr="1080.000" ybr="785.000" z_order="0"/>
    <box label="29" occluded="0" source="manual" xtl="355.000" ytl="468.000" xbr="624.000" ybr="778.000" z_order="0"/>
    <box label="17" occluded="0" source="manual" xtl="946.000" ytl="83.000" xbr="1259.000" ybr="399.000" z_order="0"/>
  </image>
  <image id="4" name="syn_04284.png" width="1920" height="1040">
    <box label="7" occluded="0" source="manual" xtl="343.000" ytl="350.000" xbr="629.000" ybr="686.000" z_order="0"/>
    <box label="9" occluded="0" source="manual" xtl="788.000" ytl="501.000" xbr="1085.000" ybr="793.000" z_order="0"/>
  </image>
  <image id="5" name="syn_03621.png" width="1920" height="1040">
    <box label="31" occluded="0" source="manual" xtl="982.000" ytl="231.000" xbr="1249.000" ybr="476.000" z_order="0"/>
    <box label="32" occluded="0" source="manual" xtl="624.000" ytl="198.000" xbr="884.000" ybr="481.000" z_order="0"/>
    <box label="26" occluded="0" source="manual" xtl="256.000" ytl="199.000" xbr="522.000" ybr="514.000" z_order="0"/>
  </image>
  <image id="6" name="syn_05608.png" width="1920" height="1040">
    <box label="12" occluded="0" source="manual" xtl="647.000" ytl="250.000" xbr="916.000" ybr="514.000" z_order="0"/>
    <box label="28" occluded="0" source="manual" xtl="986.000" ytl="181.000" xbr="1265.000" ybr="462.000" z_order="0"/>
    <box label="12" occluded="0" source="manual" xtl="333.000" ytl="483.000" xbr="616.000" ybr="793.000" z_order="0"/>
  </image>
```

# yolov8 model을 이용하여 간단하게 track 적용 실습

```
from collections import defaultdict
import cv2
import numpy as np
from ultralytics import YOLO

# Load the YOLOv8 model
model = YOLO('yolov8s.pt')

# Open the video file
video_path = "MOT17-09-SDP-raw.webm"
cap = cv2.VideoCapture(video_path)

# Store the track history
track_history = defaultdict(lambda: [])
```

- 샘플 비디오 파일 읽어서 Track 진행



## # yolov8 model을 이용하여 간단하게 track 적용 실습

```

while cap.isOpened():
    # Read a frame from the video
    success, frame = cap.read()

    if success:
        # Run YOLOv8 tracking on the frame, persisting tracks between frames
        results = model.track(frame, persist=True)

        # Get the boxes and track IDs
        boxes = results[0].boxes.xywh.cpu()
        track_ids = results[0].boxes.id.int().cpu().tolist()

        # Visualize the results on the frame
        annotated_frame = results[0].plot()

        # Plot the tracks
        for box, track_id in zip(boxes, track_ids):
            x, y, w, h = box
            track = track_history[track_id]

            track.append((float(x), float(y))) # x, y center point
            if len(track) > 30: # retain 90 tracks for 90 frames
                track.pop(0)

            # Draw the tracking lines
            points = np.hstack(track).astype(np.int32).reshape((-1, 1, 2))
            cv2.polylines(annotated_frame, pts=[points], isClosed=False, color=(230, 230, 230), thickness=10)

        # Display the annotated frame
        cv2.imshow( winname: "YOLOv8 Tracking", annotated_frame)

        # Break the loop if 'q' is pressed
        if cv2.waitKey(30) & 0xFF == ord("q"):
            break

```

```

else:

```



```

# Break the loop if the end of the video is reached

```

```

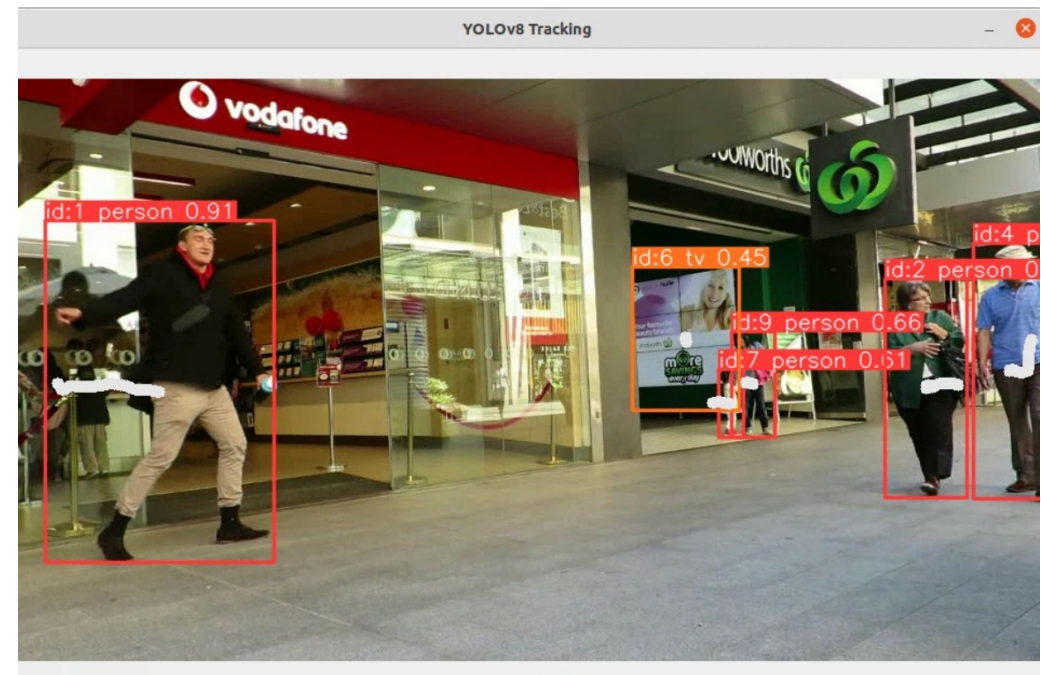
break

```

```

# Release the video capture object and close the display window
cap.release()
cv2.destroyAllWindows()

```



감사합니다.