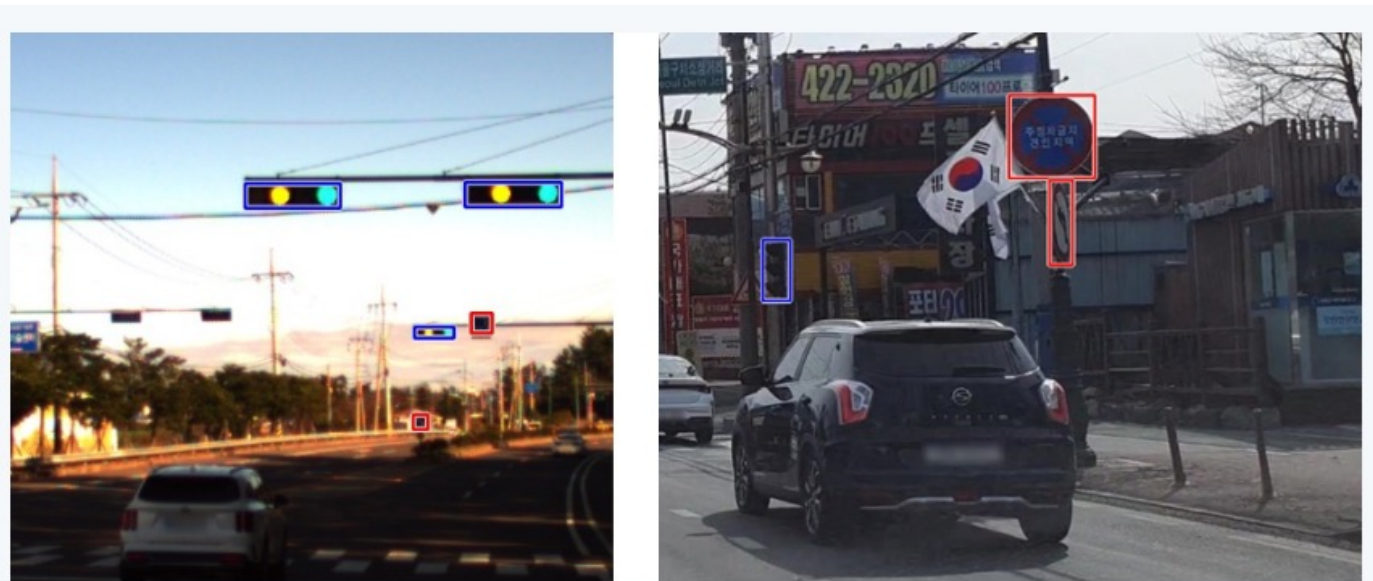


소프트웨어융합최신기술 실습과제



■ 데이터 구성 요소

```
{ "annotation": [ { "color": "red", "shape": "circle", "kind": "normal", "box":  
[1001, 558, 1022, 580], "text": "0", "type": "restriction", "class": "traffic_sign" },  
{ "color": "red", "shape": "circle", "kind": "normal", "box":  
[980, 560, 1001, 582], "text": "0", "type": "restriction", "class": "traffic_sign" } ], "image":  
{ "filename": "i0669692.jpg", "imsize": [1920, 1080] } }
```

- Annotation
 - Box : 박스의 좌표값 (xmin, ymin, xmax, ymax)
 - Class : 박스의 클래스(traffic_lighjt, traffic_sign, traffic_information(사용x))
- Image
 - Filename : 이미지 이름
 - Imsize : 이미지 크기
- 데이터 출처 : <https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&dataSetSn=187>

■ 데이터 전처리

- Mmdetection을 사용하기 위해선 PascalVOC 또는 COCO 데이터셋 포맷을 활용 해야 한다. COCO 데이터셋은 각각 Train, Validation 한개의 json 파일로 학습이 되며 본 과제에 제공되는 label은 여러개의 json 파일로 제공기 때문에 전처리 과정이 필요함.
- 사용할 포맷 COCO
 - 구성요소
 - Images : image의 id
 - Categories : category 의 id
 - Annotations : image와 category를 매칭하고 box값을 제공

```
{
  "images": [
    {
      "file_name": "i0303827.jpg",
      "width": 1936,
      "height": 1464,
      "id": 0
    },
    {
      "file_name": "i0254392.jpg",
      "width": 1936,
      "height": 1464,
      "id": 1
    }
  ],
  Images 예시
```

```
"categories": [
  {
    "id": 0,
    "name": "traffic_lighjt"
  },
  {
    "id": 1,
    "name": "traffic_sign"
  }
],
Category 예시
```

```
"annotations": [
  {
    "segmentation": [
    ],
    "image_id": 0,
    "bbox": [
      540,
      519,
      86,
      66
    ],
    "category_id": 1,
    "id": 0,
    "area": 5675,
    "iscrowd": 0
  },
  Annotations 예시
```

■ 데이터 전처리

- Annotations
 - Segmentation : segmenation 좌표값 본 과제에선 빈 리스트
 - Image_id : 앞에서 설정한 image_id
 - bbox : 순서대로 X, Y, Width, Height 의 좌표값
 - X : xmin 값
 - Y : ymin 값
 - Width : xmax - xmin 값
 - Height : ymax - ymin 값
 - category_id : 앞에서 설정한 category_id
 - area : Width * Height 값
 - iscrowd : 0으로 고정



학습용 사진

```
"annotations": [  
  {  
    "segmentation": [  
    ],  
    "image_id": 0,  
    "bbox": [  
      540,  
      519,  
      86,  
      66  
    ],  
    "category_id": 1,  
    "id": 0,  
    "area": 5675,  
    "iscrowd": 0  
  },  
  Annotations 예시
```

■ 과제

1. 주어진 데이터를 전처리 과정을 통하여 COCO 데이터 포맷으로 변환.(20점)
2. 전처리를 진행한 데이터를 토대로 MMDetection을 활용하여 학습을 진행하여 성능을 보이시오. (Pretrained 사용 x).(20점)
3. Pretrained model을 가져와서 학습을 진행하여 성능을 보이고 결과에 대한 의견을 서술하시오. (Pretrained 사용).(10점)
 - 제출시 유의사항
 - 하나의 PDF 문서로 제출, 각 문제의 코드 반드시 포함
 - 2, 3번 문제는 아래와 같은 MMDetection 에서 제공하는 성능표 첨부

```
Average Precision (AP) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.149
Average Precision (AP) @[ IoU=0.50 | area= all | maxDets=1000 ] = 0.312
Average Precision (AP) @[ IoU=0.75 | area= all | maxDets=1000 ] = 0.101
Average Precision (AP) @[ IoU=0.50:0.95 | area= small | maxDets=1000 ] = 0.046
Average Precision (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=1000 ] = 0.429
Average Precision (AP) @[ IoU=0.50:0.95 | area= large | maxDets=1000 ] = 0.800
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.213
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets=300 ] = 0.213
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets=1000 ] = 0.213
Average Recall (AR) @[ IoU=0.50:0.95 | area= small | maxDets=1000 ] = 0.093
Average Recall (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=1000 ] = 0.519
Average Recall (AR) @[ IoU=0.50:0.95 | area= large | maxDets=1000 ] = 0.800

2022-11-07 05:57:45,440 - mmdet - INFO - Exp name: yolo_sea_runtime.py
2022-11-07 05:57:45,440 - mmdet - INFO - Epoch(val) [66][65] bbox_mAP: 0.1490,
bbox_mAP_50: 0.3120, bbox_mAP_75: 0.1010, bbox_mAP_s: 0.0460, bbox_mAP_m: 0.4290,
bbox_mAP_l: 0.8000, bbox_mAP_copypaste: 0.149 0.312 0.101 0.046 0.429 0.800
```