

프로젝트 #2 결과 발표

2022. 6. 15

충북대학교 산업인공지능학과

[21-1조] 2021254001 이용규

2021254013 유대건

수행방법 및 기여도

수행방법

- 프로젝트 주제로 제시한 Kaggle 에 Data 를 활용.
- 업무의 비중을 고려하여 데이터 가공 및 전처리 파트와 코딩 및 자료작성 파트로 진행.
- 각 파트 작업 후 수시로 연락 및 대면으로 통합 작업.

업무분장 및 기여도

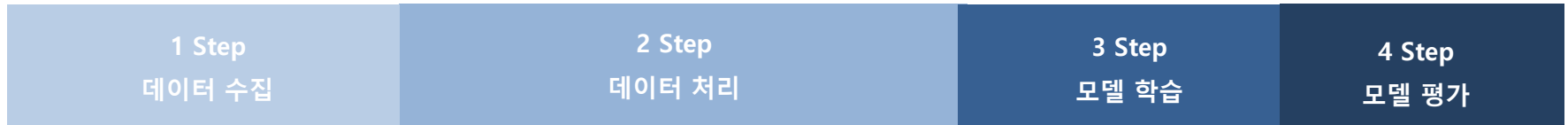
이름	비중	수행내용			비고
		프로젝트#1	프로젝트#2	주제발표	
이용규	50%	<ul style="list-style-type: none">• 데이터 증량• 주제발표	<ul style="list-style-type: none">• 레이블링• 코딩/학습	<ul style="list-style-type: none">• 실제구현(사례)	
유대건	50%	<ul style="list-style-type: none">• 코딩/학습• 결과발표	<ul style="list-style-type: none">• 데이터 수집• 레이블링	<ul style="list-style-type: none">• 자료작성 및 발표	

- 비중은 총합이 100%일 것

모델 개발 프로세스

우수성/차별성

- 데이터 수집
- 레이블링



데이터수집

주석변환

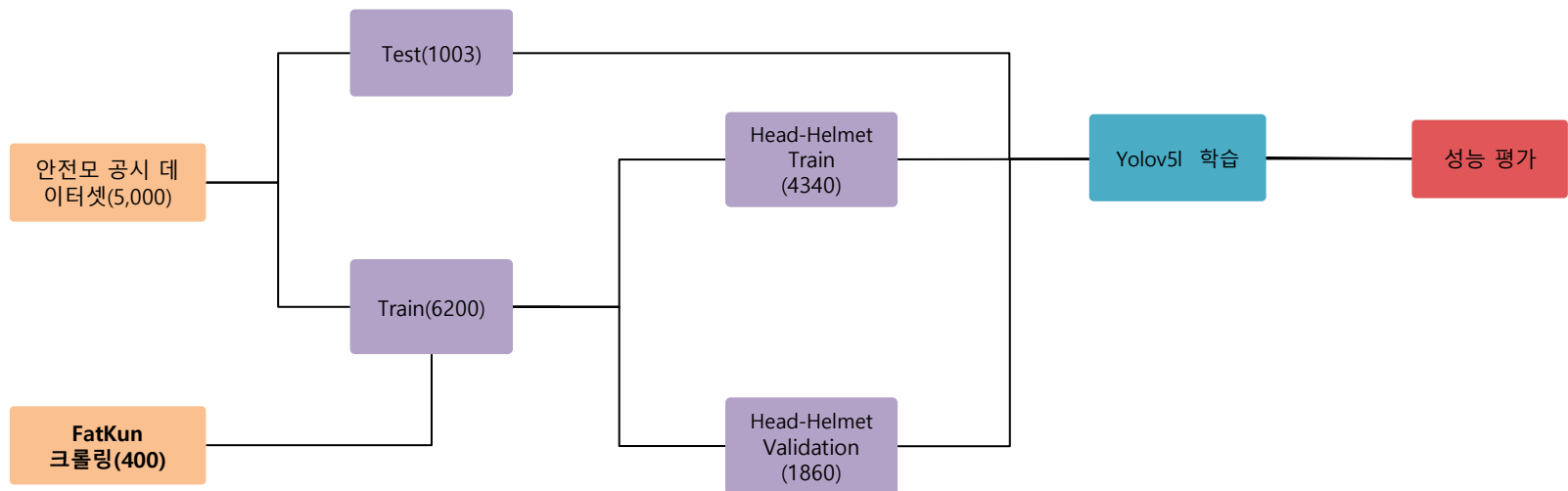
레이블링

데이터 증량

모델복잡도

인자

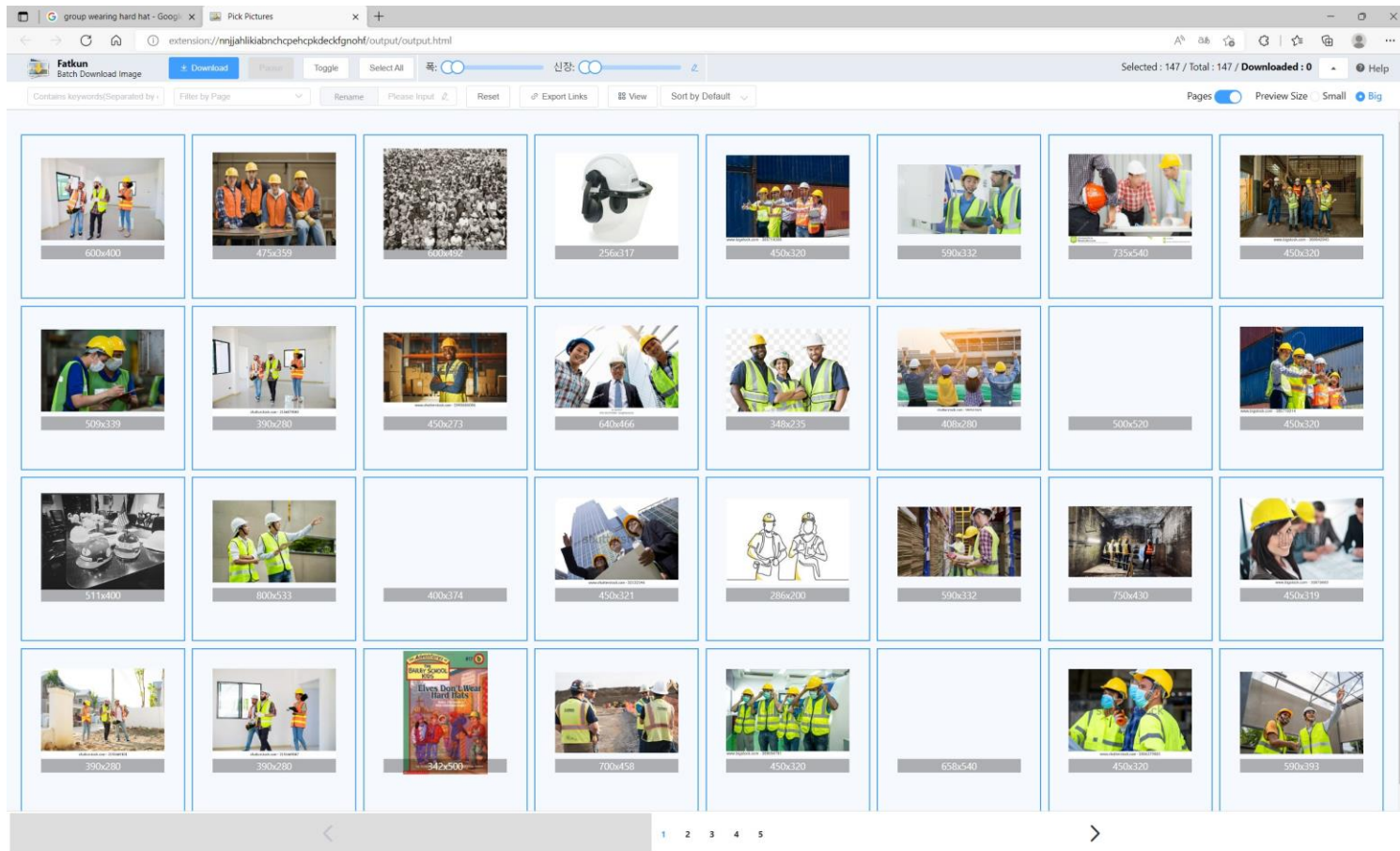
모델 평가



데이터셋 구성

Data 추가/augmentation

- Octoparse, Cyot WebCopy, Scraper, VisualScraper, **Fatkun**
- 추가 400 장 => 증량 1200 장



데이터셋 구성

Data 추가/augmentation

- Image 이름 일괄 변환

```
In [2]: import os

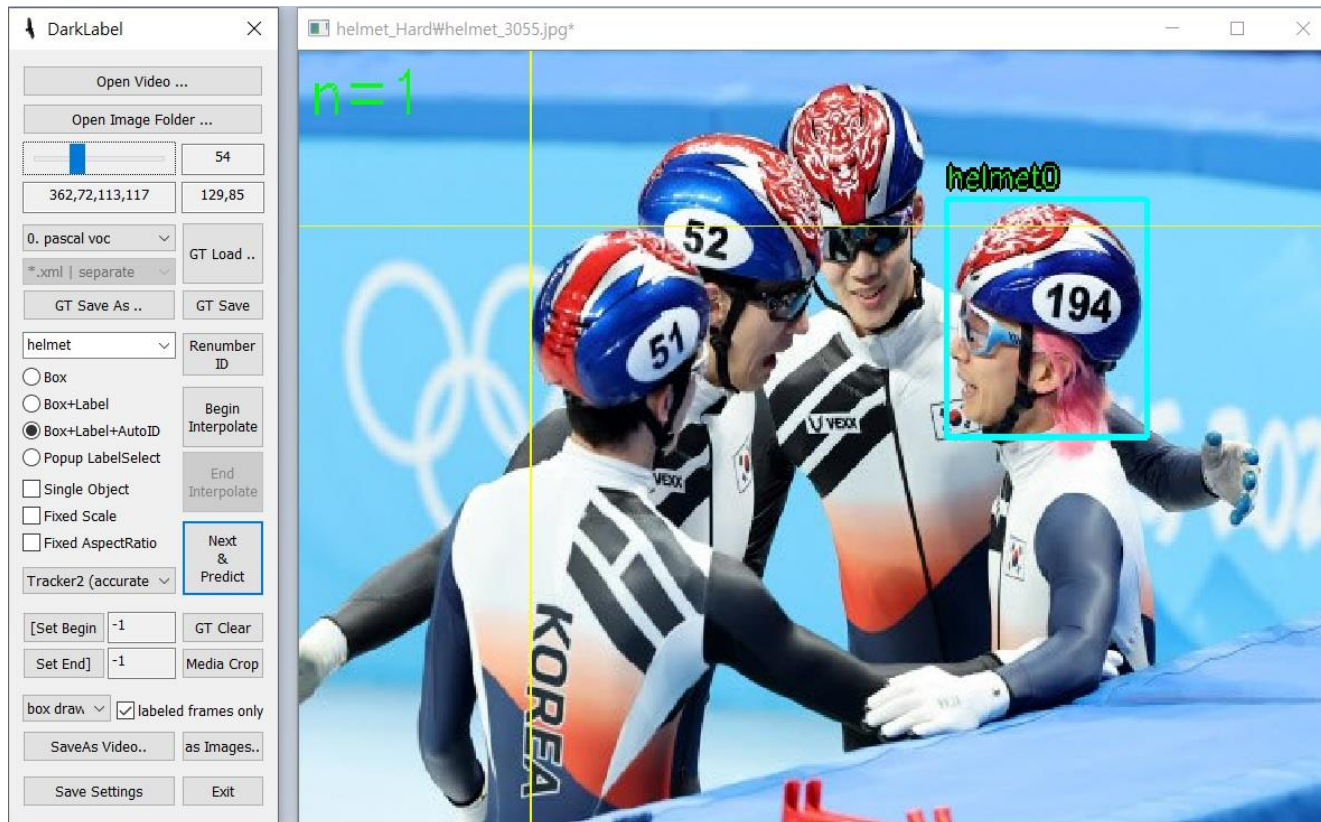
file_path = 'D:\\WC\\BNU\\WC\\CapstoneProject\\WC\\YOLO\\WC\\archive\\WC\\images\\WC'
file_names = os.listdir(file_path)

i = 3301
for name in file_names:
    src = os.path.join(file_path, name)
    dst = 'helmet_' + str(i) + '.jpg'
    dst = os.path.join(file_path, dst)
    os.rename(src, dst)
    i += 1
```

데이터셋 구성

Data 레이블링

- RoboFlow, YoloLabel, Labellmg, **DarkLabel**



데이터셋 구성

Data 추가/augmentation

- Label Data 파일 변환.

```
import numpy as np
import pandas as pd
import os
from tqdm import tqdm
import xml.etree.ElementTree as et
```

```
os.makedirs('../archive/labels', exist_ok=True)
unique_label_names = set()
```

```
# 라벨 불러오기
labels = []
label_list = os.listdir('../archive/annotations/')
for i in tqdm(range(len(label_list))):
    label = []
    f = open(os.path.join('../archive/labels', 'hard_hat_workers' + str(i) + '.txt'))
    xtree = et.parse('../archive/annotations/hard_hat_workers' + str(i) + '.xml')
    size = xtree.find('size')
    for e in xtree.findall('object'):
        name = e.find('name').text
        boundary = e.find('bndbox')
        xmin = int(boundary.find('xmin').text)
        ymin = int(boundary.find('ymin').text)
        xmax = int(boundary.find('xmax').text)
        ymax = int(boundary.find('ymax').text)
        label.append({
            'name': name,
            'boundary': {
                'xmin': xmin,
                'ymin': ymin,
                'xmax': xmax,
                'ymax': ymax
            }
        })
    unique_label_names.add(name)
```

```
label_name = 0 if name == 'head' else \
    1 if name == 'helmet' else \
    2 if name == 'person' else 3
width = int(size.find('width').text)
height = int(size.find('height').text)
xcenter = (xmax + xmin) / width / 2
ycenter = (ymax + ymin) / height / 2
xlen = (xmax - xmin) / width
ylen = (ymax - ymin) / height

content = "{name} {xcenter} {ycenter} {xlen} {ylen}\n".format(
    name=label_name,
    xcenter=xcenter,
    ycenter=ycenter,
    xlen=xlen,
    ylen=ylen)
f.write(content)
f.close()
labels.append(label)
```

모델 학습

딥러닝 학습 환경

H/W

- GPU : NVIDIA GeForce RTX 3070 Laptop GPU(8G)
- Memory : 32G
- CPU : Intel I7-10870H CPU @ 2.20GHz

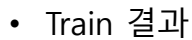
OS

- Linux : Ubuntu 20.04.4 LTS

Train Parameter

```
> python train.py --img 640  
    --batch 8  
    --epochs 200  
    --data ~/dataset/helmet.yaml  
    --cfg ./models/yolov5l.yaml  
    --weights yolov5l.pt  
    --name helmet
```


학습 출력 결과



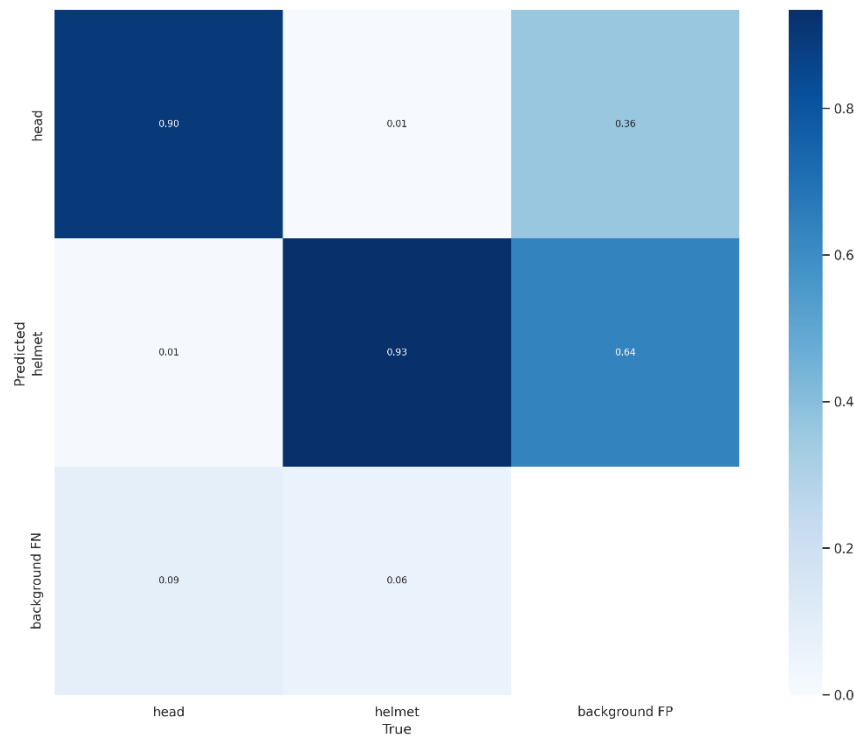
The figure displays a 4x4 grid of 16 images, each showing a different person or group of people wearing various types of headgear. Red bounding boxes are drawn around the headgear, and labels are provided for each, indicating the detected headgear type and its confidence score. The labels are as follows:

- Row 1:
 - Image 1: helmet 0.9, head 1.0, head 0.9
 - Image 2: head 0.9, head 0.9, head 0.9, head 0.9, head 0.9
 - Image 3: helmet 0.7, head 0.7, head 0.5
 - Image 4: helmet 0.5, head 1.0, head 0.5, head 0.5
- Row 2:
 - Image 5: helmet 0.3
 - Image 6: helmet 0.4, head 0.8, head 1.0
 - Image 7: head 0.9, helmet 1.0, helmet 1.0, helmet 1.0
 - Image 8: helmet 1.0, helmet 1.0, helmet 0.9, helmet 0.9
- Row 3:
 - Image 9: head 1.0, head 1.0, head 1.0
 - Image 10: head 0.8, helmet 0.8, head 1.0, helmet 0.8
 - Image 11: helmet 0.3, helmet 0.9, helmet 0.9, helmet 1.0
 - Image 12: helmet 1.0, helmet 1.0, helmet 1.0, helmet 1.0
- Row 4:
 - Image 13: head 1.0, head 0.4
 - Image 14: head 1.0, head 1.0, head 0.3, head 0.3
 - Image 15: head 1.0, head 0.9
 - Image 16: head 1.0, head 0.3, head 0.3

결과 및 토의

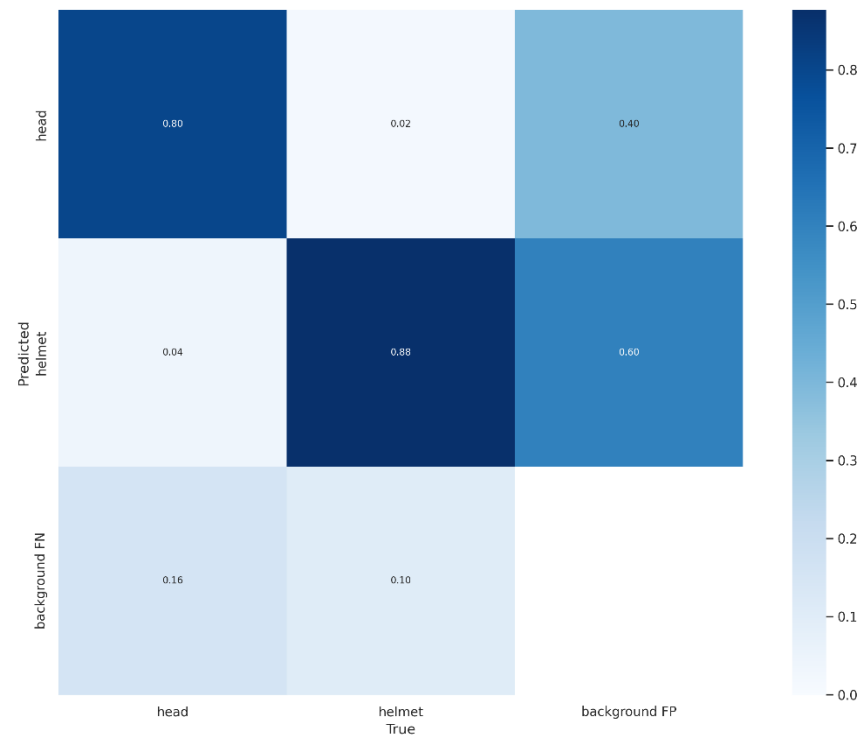
검증 결과

- train:val = 7:3



- 데이터 증량 전

Head : 0.9
Helmet : 0.93

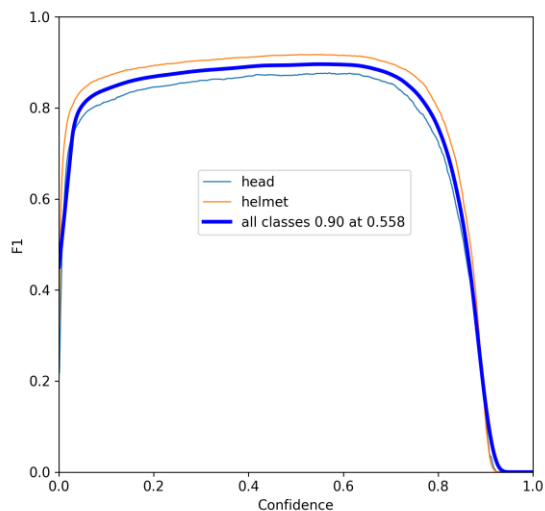


- 데이터 증량 후

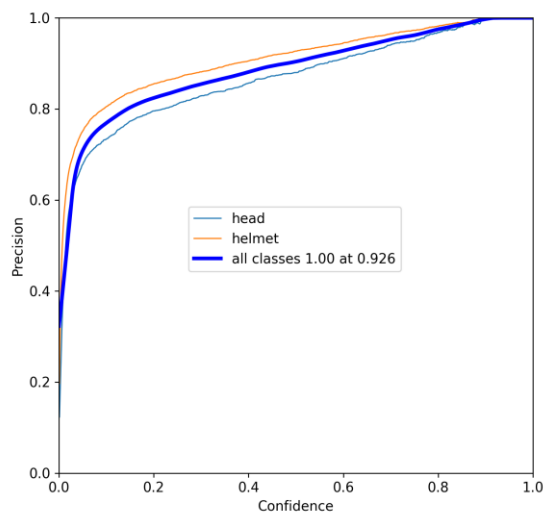
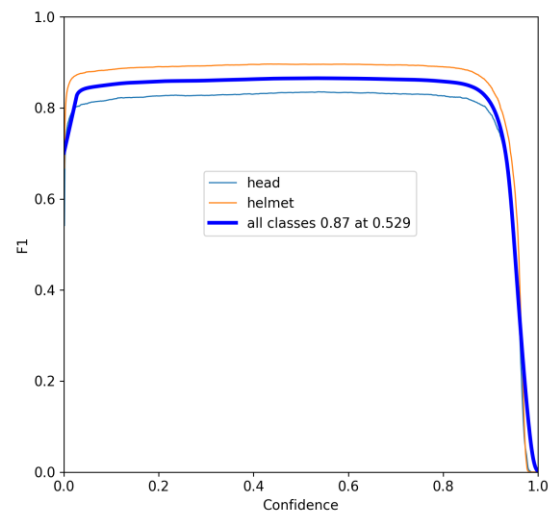
Head : 0.80
Helmet : 0.88

결과 및 토의

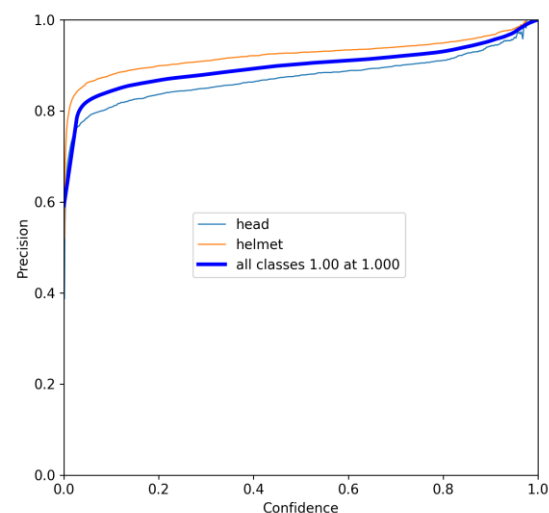
검증 결과



- F1 Curve 데이터 증량 전후

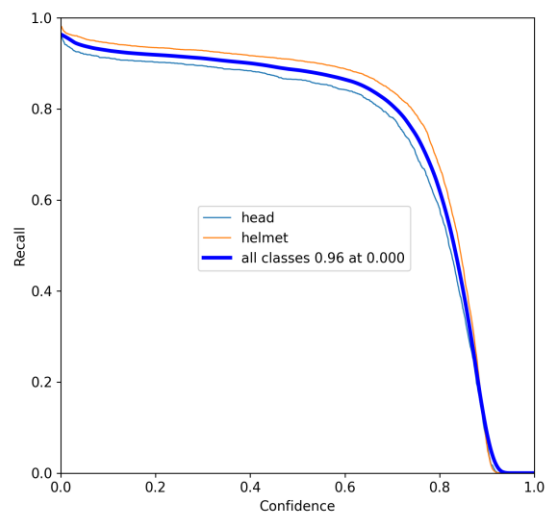


- P Curve 데이터 증량 전후

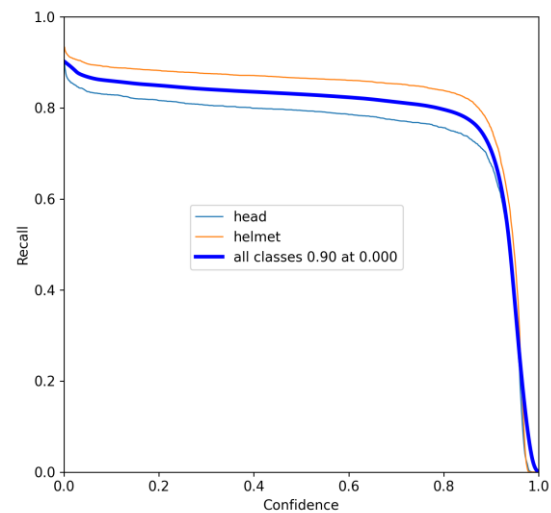


결과 및 토의

검증 결과



- R Curve 데이터 증량 전후

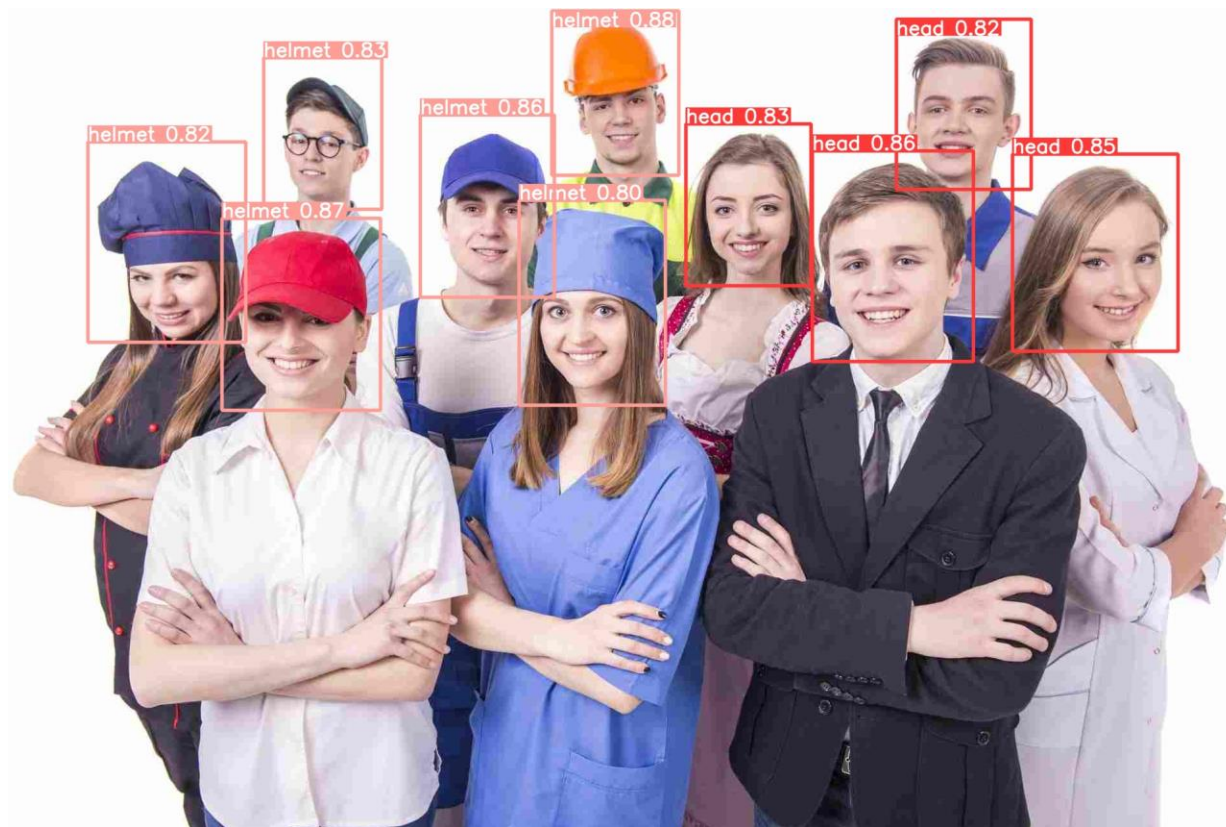


결과 및 토의 (학습 데이터 추가 전)

테스트 결과

Head : 0.84

Helmet : 0.84



오분류 사례 및 개선점

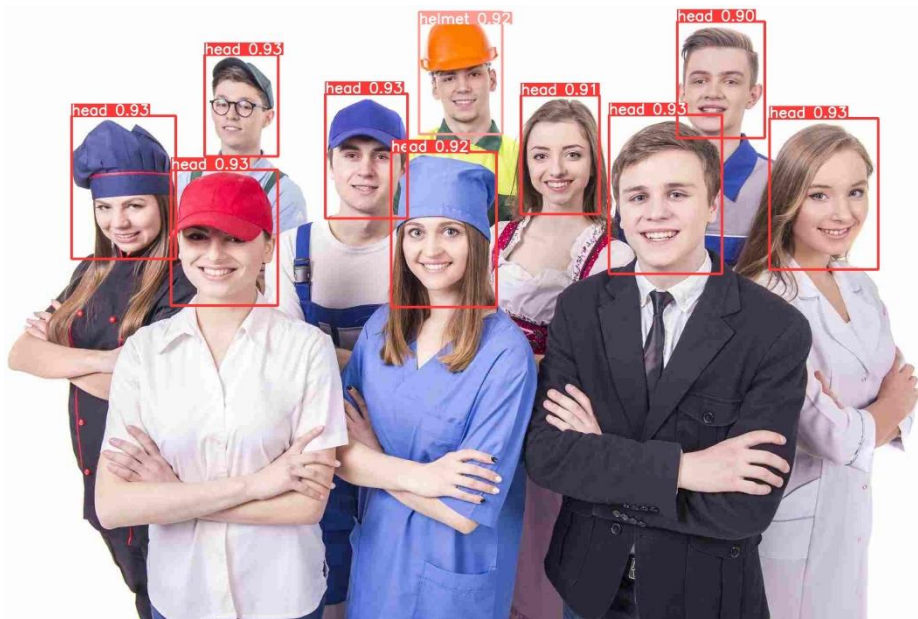
- Cap, Hat, 두건 등을 착용한 사람에 대해 Helmet 으로 인식.
- Cap, Hat, 두건 등에 대해 Head 로 학습 데이터 추가 및 Labeling 작업.

결과 및 토의 (학습 데이터 추가 후)

테스트 결과

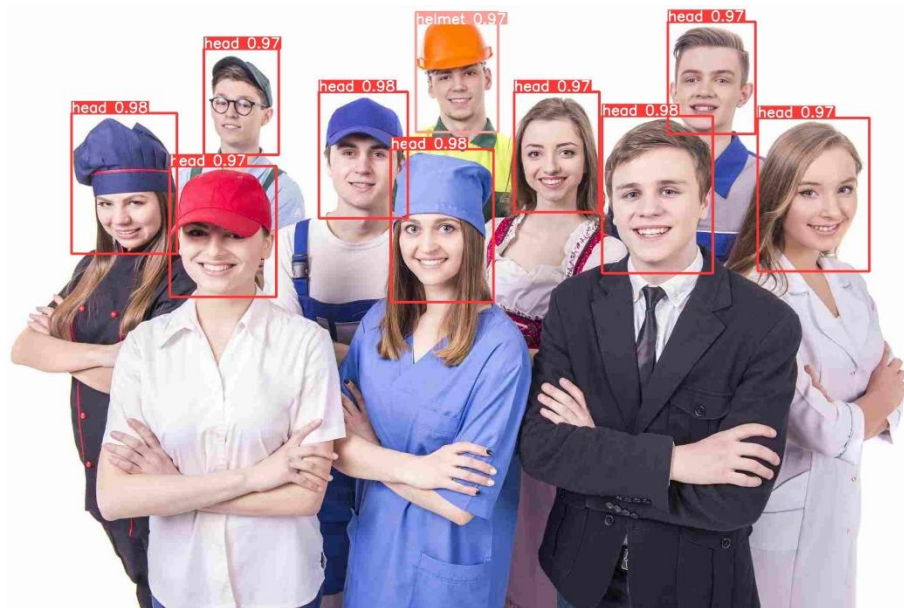
학습데이터 추가 후

Head : 0.92
Helmet : 0.92



학습데이터 추가 후

Head : 0.97
Helmet : 0.97



결과 및 토의

토의 및 개선점

- 검출 객체에 대한 Labeling 정확도에 따라 검출률 상승.
- 다양한 학습 데이터 수집 및 학습을 통하여 오인식률 저하
- 검출 객체와 유사한 데이터 증량으로 정확도 상승.
- 프로그램 작성에 대한 이해도 보다 효율적인 학습 방법에 대한 이해도 필요.
- 프로그램 사용 방법에 대한 가이드를 사전 제시할 경우 다양한 방법 시도 가능.
- 후처리 및 다른 객체인식 알고리즘을 연계한 검출률 향상

감사합니다