

[프로젝트 #2]

YOLO를 이용한 안전모 검출

2022. 04. 27

도규원, 김현용

충북대학교 산업인공지능학과

공지사항

- 출석관리 철저

- 채팅창에 “20학번 000 출석”으로 1일 2회 기록 : 입장(19시)과 퇴장(21시~)
- 20학번 : 프로젝트(60%) + 주제발표(10%) + **출석(30%)**

항 목		비 율	내 용	비 고
발표 평가 (총 100%)	1차 발표	30%	Project #1. CNN을 이용한 불량검출	조 편성
	2차 발표	40%	Project #2. YOLO를 이용한 안전모 검출	
주제발표 평가		10%	조별 발표	조별 발표 주제 선정
출 석		20%	총 수업시간의 $\frac{3}{4}$ 미달시 F. (총 수업시간 : 60시간, 15시간 이상 미출석시 F)	

→ 5일 이상 결석 시 F 처리

조편성 결과 (4/27 현재)

20 학번 조편성

조	성명	발표여부
20-1	전일우, 이효중, 박성범	○
20-2	임동민, 신정환, 안건호	
20-3	최원희, 손의걸	○
20-4	윤재웅, 김성웅	○
20-5	강윤구, 김병근	○
20-6	박민우	○
20-7	원형일, 장민우	
20-8	고정재, 유용주	○

총 8조 / 17명

21 학번 조편성

조	성명	발표여부
21-1	이용규, 유대건	
21-2	김대훈	○
21-3	최준혁, 이지연	
21-4	김상순, 정수현	
21-5	방창현, 정원용	
21-6	우상진, 김준태	
21-7	봉은정, 김원우	
21-8	이충현, 이지호	
21-9	정준영, 윤범희	

총 9조 / 17명

조별 발표 주제 (4/27 현재)

주차	날짜	대면	발표 주제	발표조
7	4/20	O	Project #2 : YOLO를 이용한 안전모 검출	김현용, 도규원
8	4/27	X	검출과 분할 – Object Detection vs. Segmentation, Mask-RCNN	21-3, 21-4
9	5/4	X	AI특강1(19:00~20:30)	초청강사
10	5/11	X	RCNN계열-R-CNN, fast R-CNN, faster R-CNN의 비교(구조, 성능) 1-Stage detector – YOLO 외 SSD, RetinaNet 등	21-6, 21-9 20-7, 21-1
11	5/18	X	YOLOv5 사용법 - 각종 파라미터 설정, 학습/검증/추론 방법 등 주석 - 레이블링 파일형식(xml, json, yolov5) 변환 코드 평가지표- mAP(예제, 코드) AI특강2(21:00~22:00)	20-2, 21-7 21-5 21-8 초청강사
12	5/25	O	프로젝트 중간점검 : 5분 진행상황 발표 (20학번, 21학번)	모든 조
13	6/1	x	[휴무 대보강] 동영상 강의 시청	윤성철, 이광연
14	6/8	x	AI특강3(19:00~20:00) 프로젝트 최종점검 : 테스트 데이터 공개 → 검출결과 제출	초청강사
15	6/15	O	프로젝트 #2 발표평가	모든 조

Q & A

안전모 데이터 확인

- Kaggle의 안전모 데이터셋

<https://www.kaggle.com/datasets/andrewmvd/hard-hat-detection>

Safety Helmet Detection

Improve work safety by detecting the presence of people and safety helmets.

Data Code (12) Discussion (6) Metadata

About Dataset



+AI 메이커톤 설명자료

팀명 : 투피스

팀장 : 이동현

팀원 : 김석민, 남하연, 김보형

모델 개발 과정

받은 데이터셋의
853개 다시
라벨링하였다.

kaggle에서 head
라벨이 있는
데이터만 추출하여
241개 를
추가하였다.

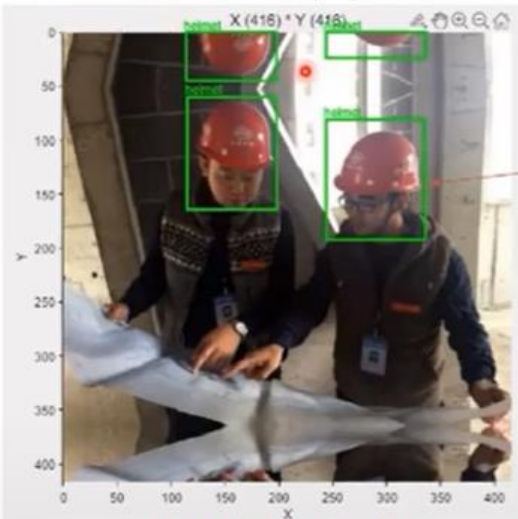
데이터
augmentation을
통해 853개를
3711개로 241개를
1172개로 늘렸다.

두 데이터를 합친
후, train dataset
80%, test dataset
20%로 설정하여
yolov5x, yolov5x6
모델 학습 후 검증

재라벨링

출제문제 설명

- 훈련데이터셋의 이미지-레이블 예
- 문제 취지와 다른 레이블링(2개) 포함



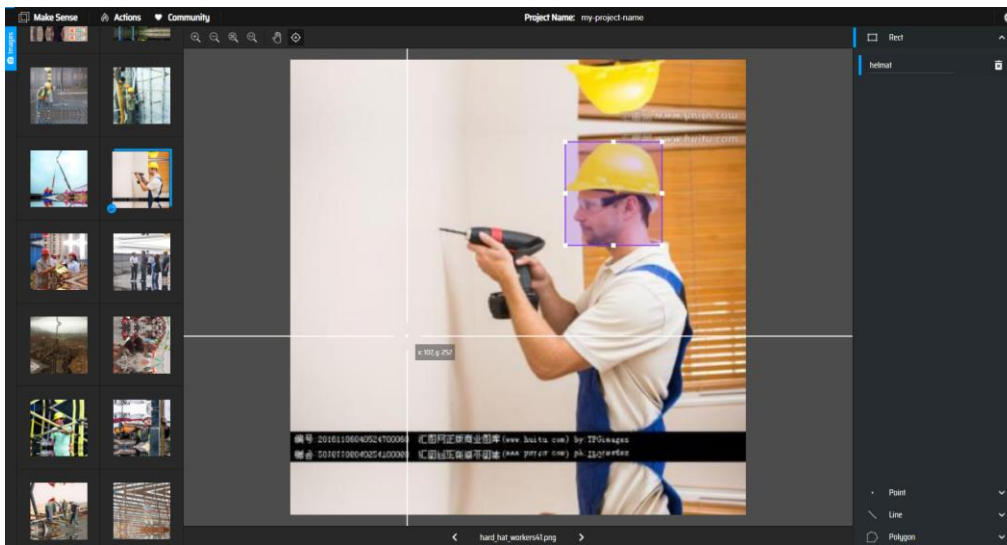
Hard_hat_workers1002.png

```
<annotation>
  <folder>images</folder>
  <filename>hard_hat_workers1002.png</filename>
  <size>
    <width>416</width>
    <height>416</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>helmet</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <occluded>0</occluded>
    <difficult>0</difficult>
    <bndbox>
      <xmin>244</xmin>
      <ymin>81</ymin>
      <xmax>336</xmax>
      <ymax>192</ymax>
    </bndbox>
  </object>
  <object>
    ...
  </object>
  <object>
    ...
  </object>
  <object>
    ...
  </object>
</annotation>
```

Hard_hat_workers1002.xml

- 받은 데이터 셋에는 문제의 취지와 맞지 않는 라벨링이 있기 때문에 그대로 학습을 하는 것은 제대로 학습이 되지 않을 것이라고 판단하였다.

재라벨링



- makesense를 활용하여 취지에 맞게 데이터를 라벨링하였다.
- 모든 데이터를 전부 다시 하지 않고 853개만 라벨링을 하였다.
- <https://www.makesense.ai/>

모델 개발 과정

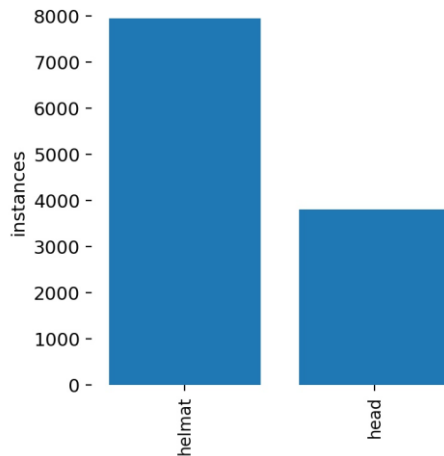
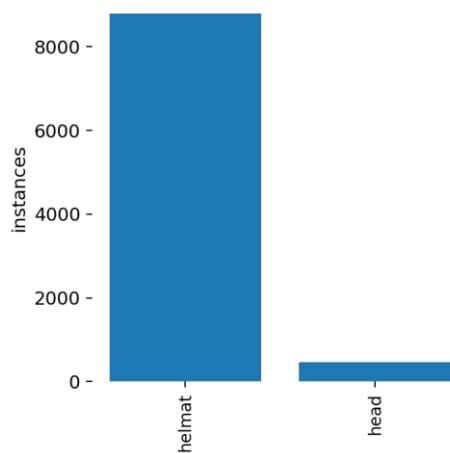
받은 데이터셋의
853개 다시
라벨링하였다.

kaggle에서 head
라벨이 있는
데이터만 추출하여
241개 를
추가하였다.

데이터
augmentation을
통해 853개를
3711개로 241개를
1172개로 늘렸다.

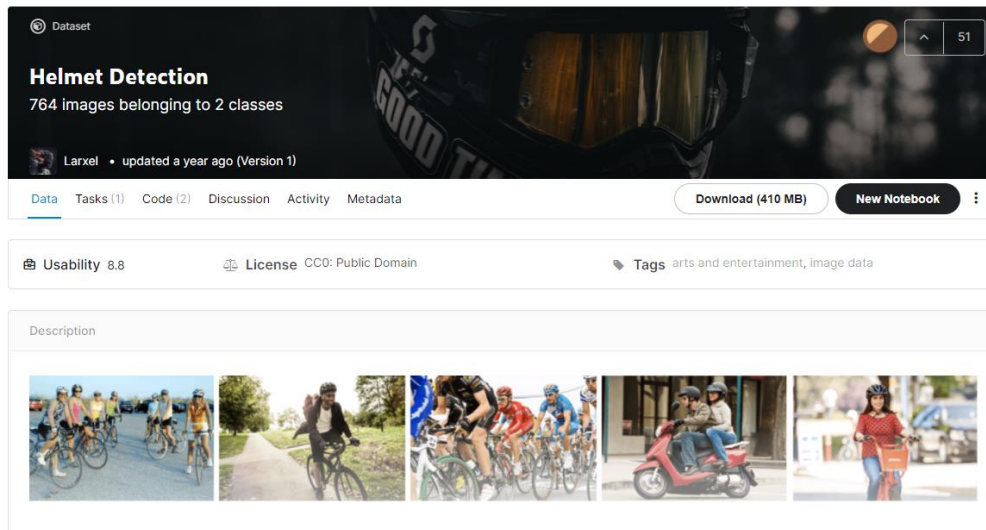
두 데이터를 합친
후, train dataset
80%, test dataset
20%로 설정하여
yolov5x, yolov5x6
모델 학습 후 검증

데이터셋 라벨링 구성



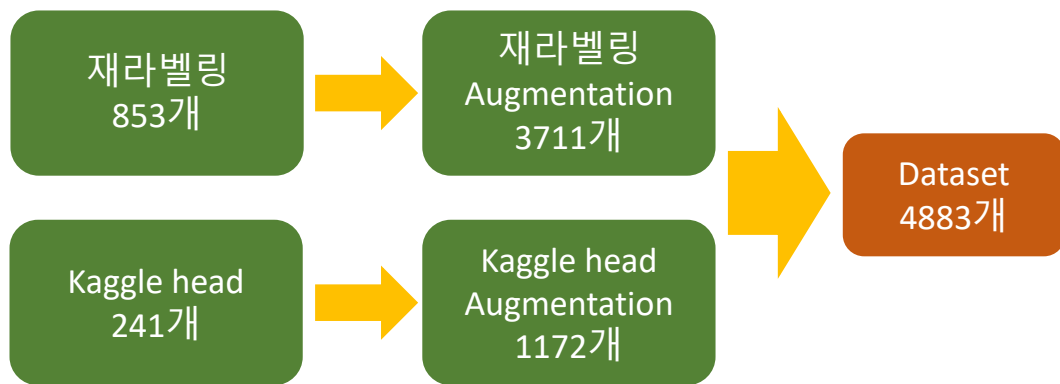
- 왼쪽 차트는 Kaggle에서 받은 head 데이터를 추가하지 않은 데이터의 라벨링 개수를 보여준다. helmet 라벨에 비해 head 라벨이 현저히 적은 것을 알 수 있다.
- 오른쪽 차트는 Kaggle에서 받은 head 데이터를 추가한 데이터의 라벨링 개수를 보여준다. 데이터 불균형을 어느정도 해소한 모습을 볼 수 있다.

Kaggle Head 라벨 데이터 추가



- 다음의 Kaggle 데이터 셋을 받아서 head label이 있는 데이터만 추출하여 추가하였다.
- <https://www.kaggle.com/andrewmvd/helmet-detection>

데이터 augmentation



- 데이터를 증폭시키기 위해서 코드를 구현하여 augmentation을 수행하였다.
- 방법은 반전, 회전, 스케일링, 이동을 사용하였다.
- <https://blog.paperspace.com/data-augmentation-for-bounding-boxes/>

데이터 augmentation



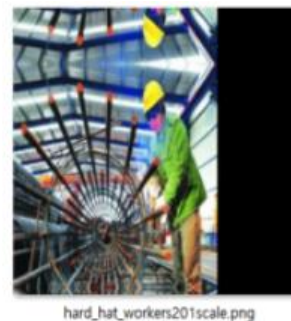
원본



반전



회전

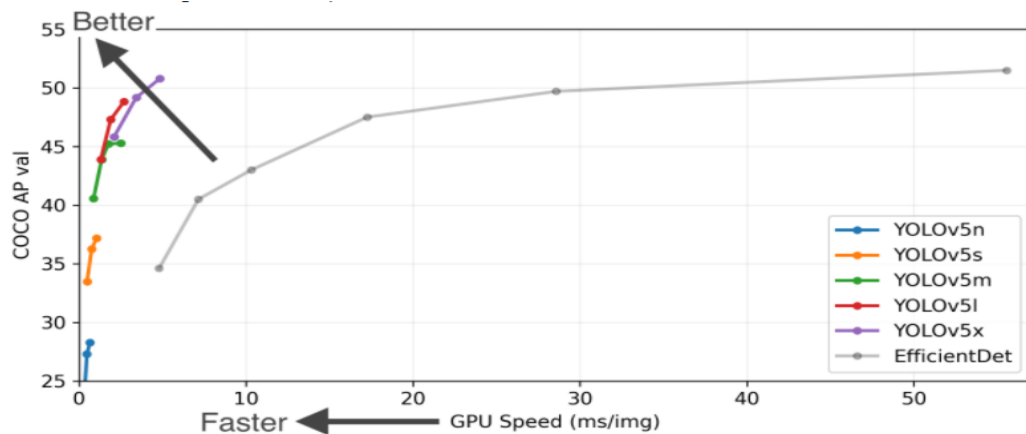


스케일링



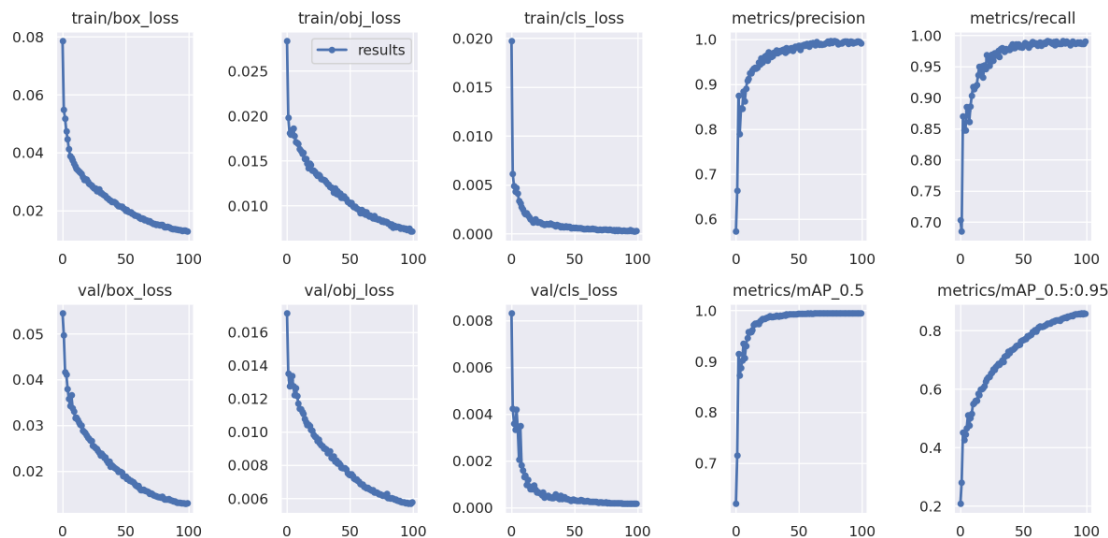
이동

사용한 모델



- YOLOv5x 모델을 사용하였다.
- COCO dataset에 대해 pretrained 모델을 사용하여 학습하였다.

학습 결과



- YOLOv5x 모델은 mAP0.5 기준으로 최고 0.99484, mAP0.5:0.95 기준으로 최고 0.85961의 결과가 나왔다.
- 추론 시 YOLOv5x를 활용하고 TTA 기법을 활용하여 추론하였다.