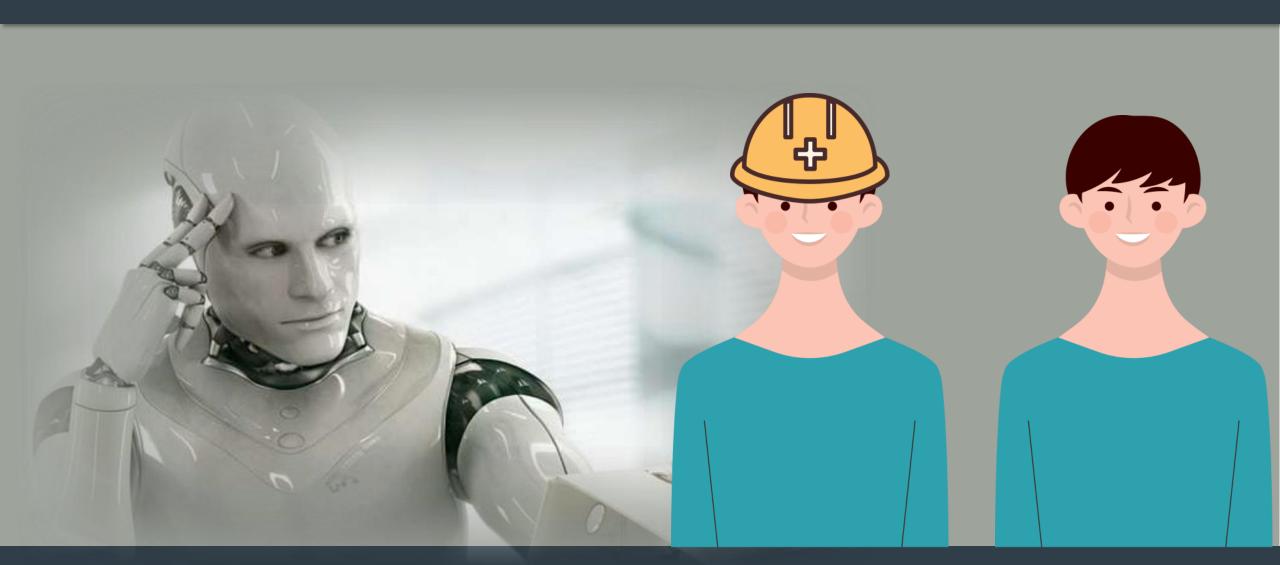
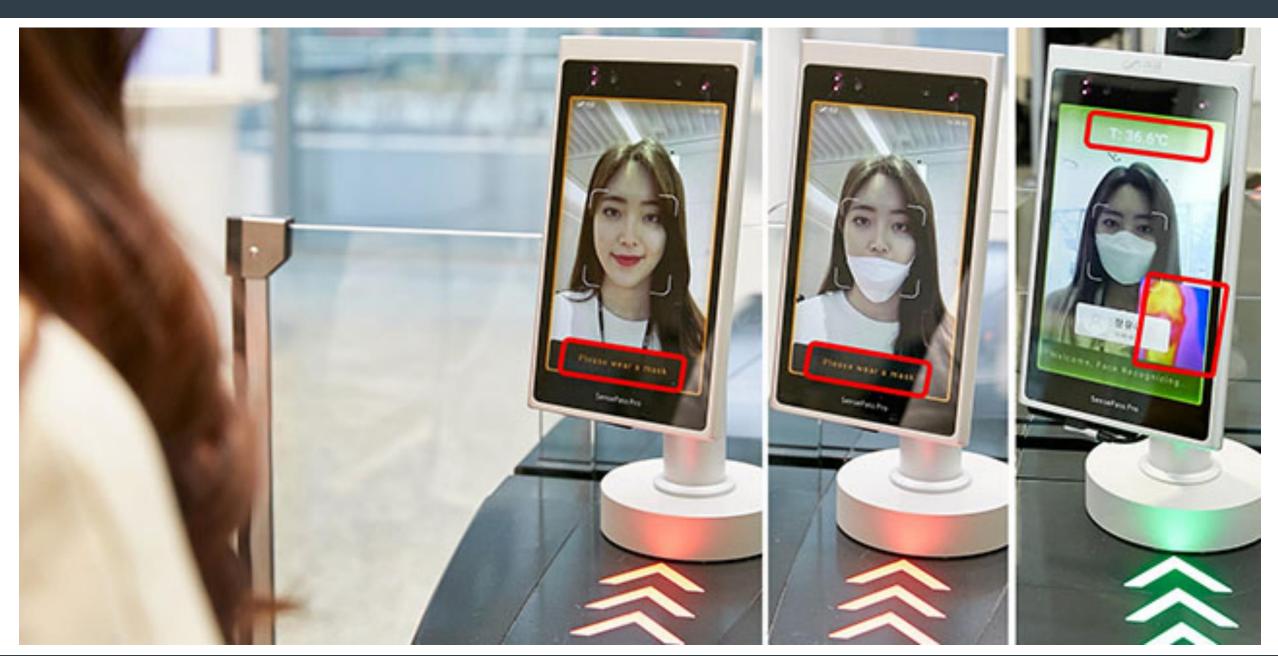
객체 탐지를 활용한 안전모 착용 여부 확인

딥러닝AI개론

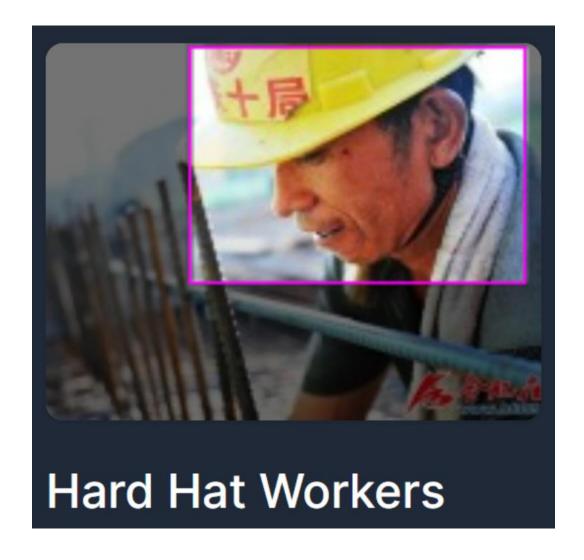
통계데이터사이언스전공 202010936 전소영



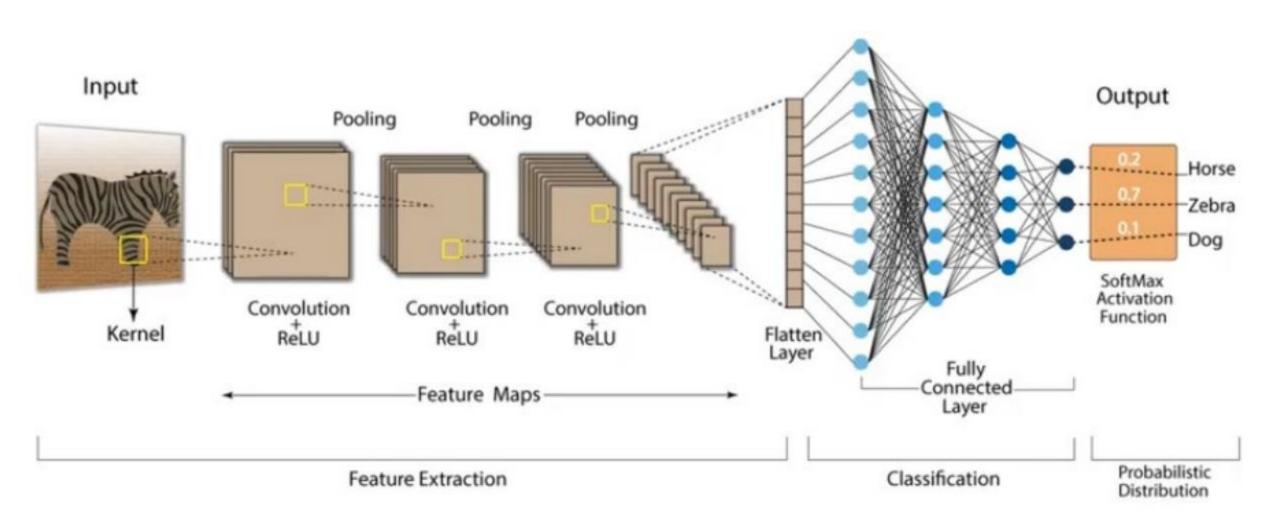


https://www.lgcns.com/pr/news/12676/

roboflow



03 모델 선택 이유

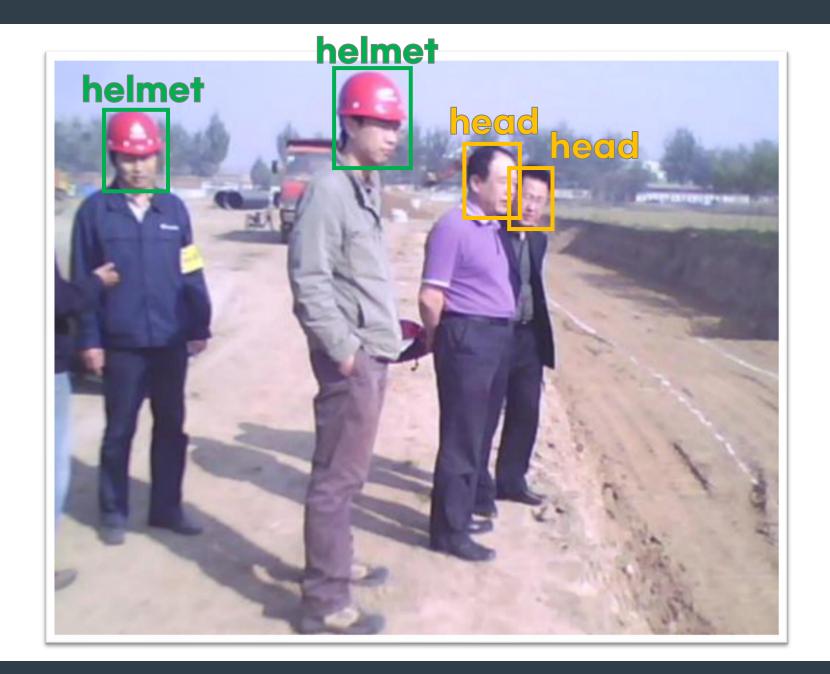


03 모델 선택 이유

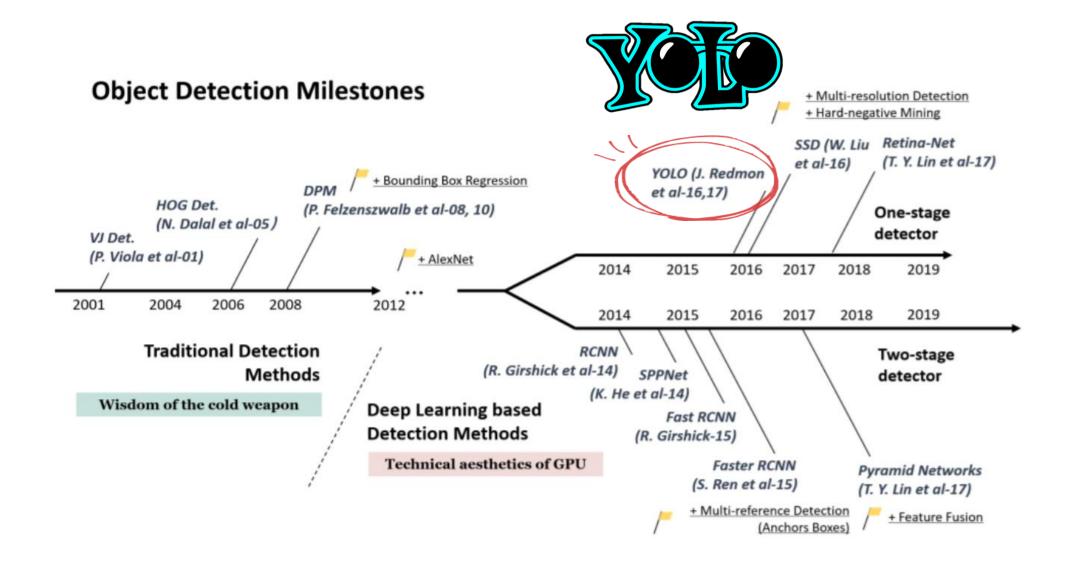




Object Detection 객체 탐지



03 모델 선택 이유





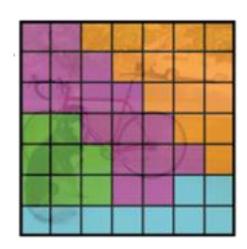
You Only Look Once



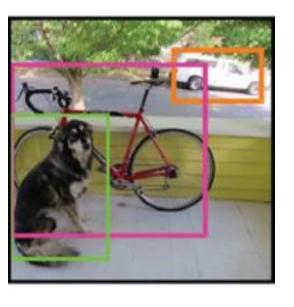








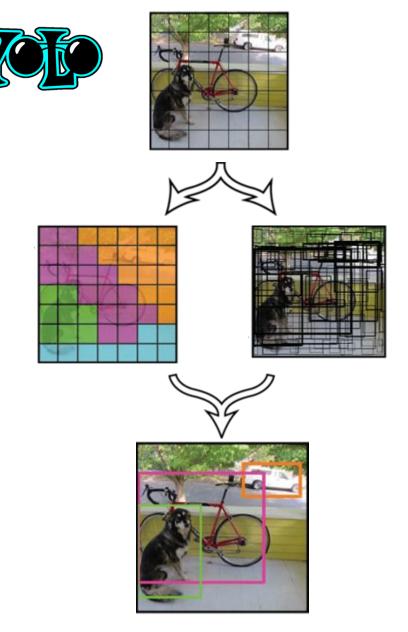




기존 객체 탐지 모델 전처리모델 + 인공신경망 따로 <mark>결합</mark>해 사용 YOLO는 전처리모델 + 인공신경망 을 **통합**한 모델

기존 실시간 탐지 모델들에 비해 2배 높은 정밀도를 가져 정확도가 높음

속도가 빨라 실시간으로 Object Detection이 가능함

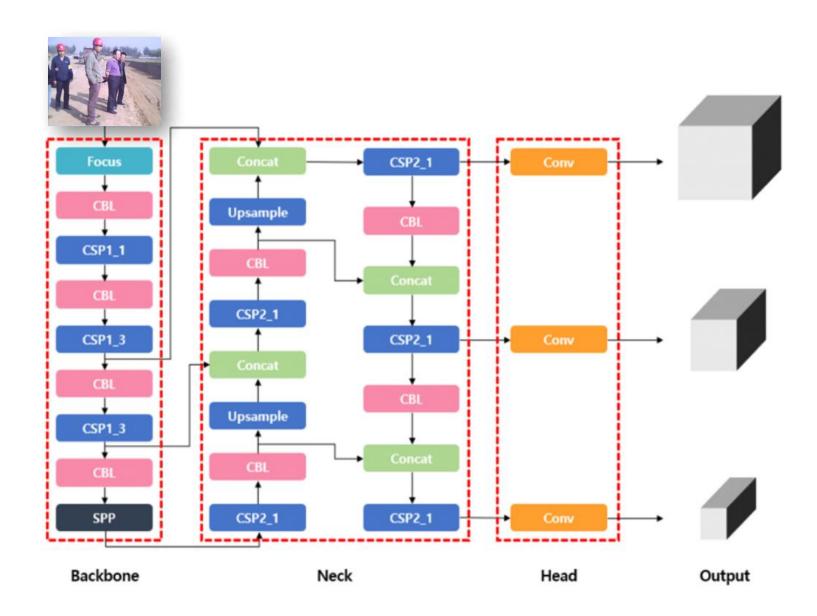


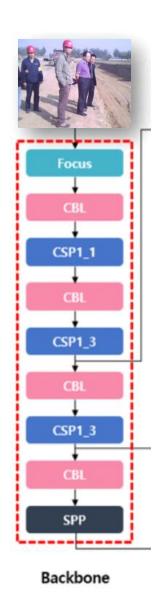
YOLOv5를 사용한 수영모 착용 여부 판별 알고리즘

고광현, 김민철, 이승환, 유지나*, 김재현

아주대학교 전자공학과, *아주대학교 AI 융합네트워크학과

{kokh0128, dktl530, tmdghks7643, *jina1114, jkim}@ajou.ac.kr



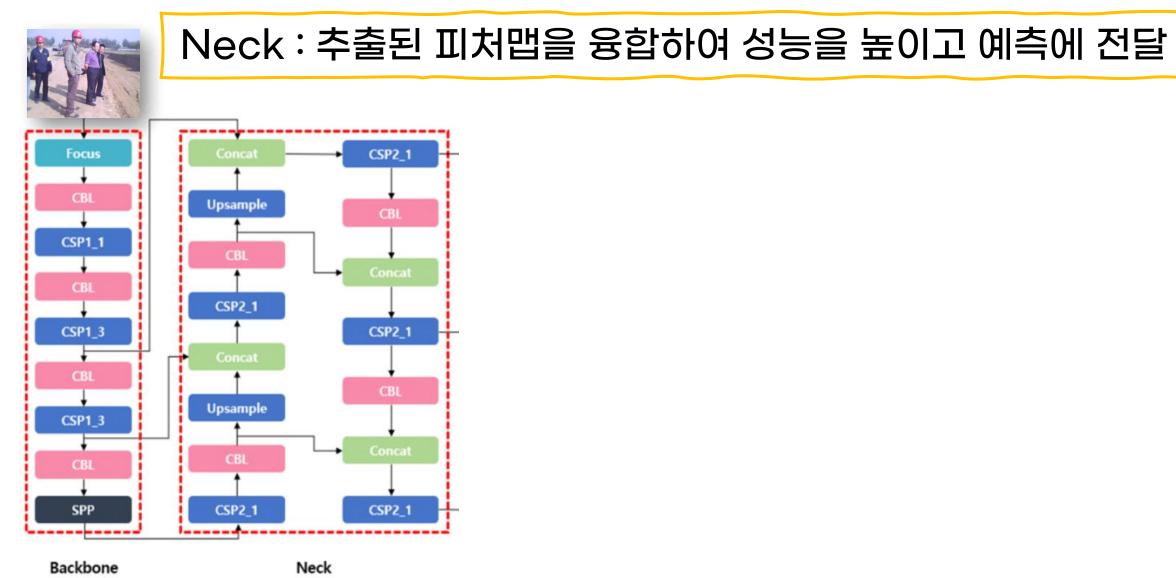


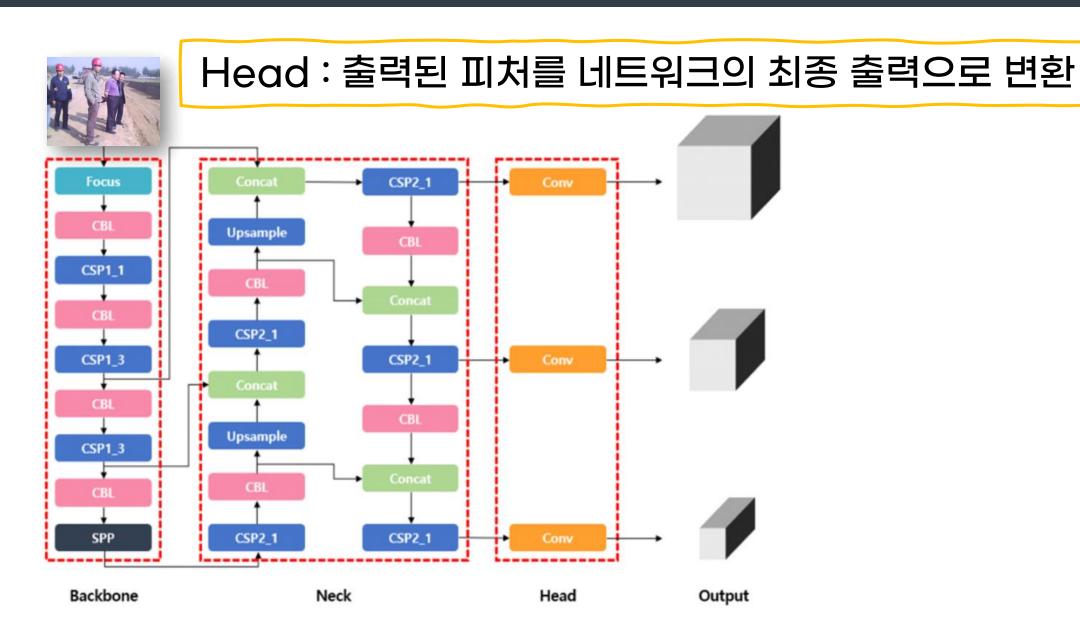
Backbone: 입력 이미지로부터 다양한 피처맵을 추출

CBL: 컨볼루셔널 레이어, 배치 정규화, Leaky Relu의 활성함수로 이루어 진 블록으로 피처를 추출하는데 기본적으로 사용 (Convolution with Batch normalization and Leaky Relu)

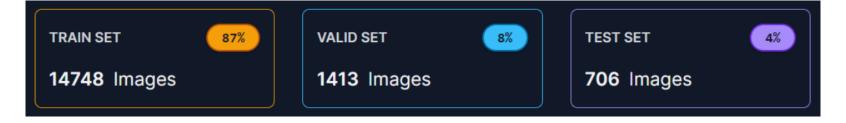
CSP : 피처맵의 일부에서만 연산을 수행하고 나머지는 부분과 통합하여 연산량을 줄인다

SPP : 피처맵을 다양한 크기의 필터로 풀링한 후 합쳐 성능을 향상시킴





05 데이터 설명





05 데이터 설명

Preprocessing Auto-Orient: Applied

Resize: Stretch to 416x416

Modify Classes: 0 remapped, 1 dropped

Augmentations Outputs per training example: 3

Flip: Horizontal

Crop: 0% Minimum Zoom, 20% Maximum Zoom

Rotation: Between -10° and +10°

Grayscale: Apply to 10% of images

Hue: Between -20° and +20°

Saturation: Between -25% and +25%

Brightness: Between -20% and +20%

Exposure: Between -20% and +20%

Blur: Up to 1px

Cutout: 6 boxes with 3% size each



```
# yolov5 git 다운로드
!git clone <u>https://github.com/ultralytics/yolov5</u>
%cd yolov5
!git reset --hard fbe67e465375231474a2ad80a4389efc77ecff99
```

```
Cloning into 'yolov5'...
remote: Enumerating objects: 16013, done.
remote: Counting objects: 100% (46/46), done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 16013 (delta 25), reused 23 (delta 13), pack-reused 15967
Receiving objects: 100% (16013/16013), 14.66 MiB | 24.18 MiB/s, done.
Resolving deltas: 100% (10987/10987), done.
/content/yolov5
HEAD is now at fbe67e4 Fix `OMP_NUM_THREADS=1` for macOS (#8624)
```

```
#,패키지, 유틸, 모델, 기본가중치 다운로드
!pip install -qr requirements.txt
import torch
from IPython.display import Image, clear_output
from utils.downloads import attempt_download
```

- ▼ myolov5
 - data
 - models
 - utils
 - CONTRIBUTING.md
 - LICENSE
 - README.md
 - detect.py
 - export.py
 - hubconf.py
 - requirements.txt
 - setup.cfg
 - 🖺 train.py
 - tutorial.ipynb
 - val.py



```
#roboflow 에서 데이터셋 다운로드
%cd /content/yolov5

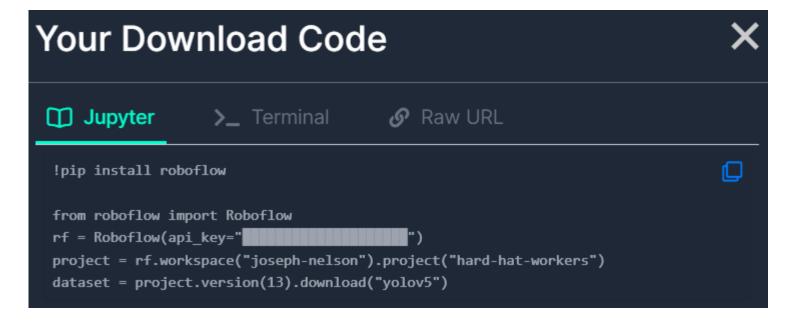
!pip install roboflow

from roboflow import Roboflow

rf = Roboflow(api_key="5WLMzX4BhjFxpVulLz1o")

project = rf.workspace("joseph-nelson").project("hard-hat-workers")

dataset = project.version(13).download("yolov5")
```





#학습시키기

%%time

%cd /content/yolov5/

!python train.py --img 416 --batch 32 --epochs 22 --data .<u>/Hard-Hat-Worker</u> --cfg .<u>/models/yolov5s.yaml</u> --weights '<u>/content/drive/MyDrive/yolov5/yolov</u> --project <u>/content/drive/MyDrive</u>/ --name yolov5s_results1

- yolov5
 - data 🖿
 - models
 - utils 🖿
 - CONTRIBUTING.md
 - LICENSE
 - README.md
 - detect.py
 - export.py
 - hubconf.py
 - requirements.txt
 - setup.cfg
 - train.py
 - tutorial.ipynb
 - val.py



#학습시키기 %%time

%cd /content/yolov5/

!python train.py -- img 416 --batch 32 --epochs 22 --data .<u>/Hard-Hat-Worker</u> --cfg .<u>/models/yolov5s.yaml</u> --weights '<u>/content/drive/MyDrive/yolov5/yolov</u> --project <u>/content/drive/MyDrive/</u> --name yolov5s_results1

- yolov5
 - data 🖿
 - models
 - utils
 - CONTRIBUTING.md
 - LICENSE
 - README.md
 - detect.py
 - export.py
 - hubconf.py
 - requirements.txt
 - setup.cfg
 - train.py
 - tutorial.ipynb
 - val.py



#학습시키기

%%time

%cd /content/yolov5/

!python train.py --img 416 --batch 32 --epochs 22 --data .<u>/Hard-Hat-Workers-13/data.yaml</u> # --cfg .<u>/models/yolov5s.yaml</u> --weights '<u>/content/drive/MyDrive/yolov5/yolov5s_results2/weights/best.pt</u>' # --project <u>/content/drive/MyDrive</u>/ --name yolov5s_results1

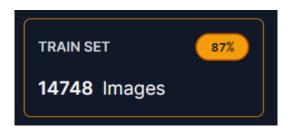




```
#학습시키기
%%time
%cd /content/yolov5/
!python train.py --img 416 --batch 32 --epochs 22 --data ./Hard-Hat-Workers-13/data.yaml ₩
--cfg ./models/yolov5s.yaml --weights '/content/drive/MyDrive/yolov5/yolov5s_results2/weights/best.pt' ₩
--project /content/drive/MyDrive/ --name yolov5s_results1
```

Batch size : 1회 트레이닝(훈련)에 사용할 관측치의 수

- 클수록 빠르지만 성능 저하
- 작을수록 느리지만 성능 향상



$$14748 \div 32 = 460.875 \approx 461$$



```
#학습시키기
%%time
%cd /content/yolov5/
!python train.py --img 416 --batch 32 --epochs 22 --data ./Hard-Hat-Workers-13/data.yaml #
--cfg ./models/yolov5s.yaml --weights '/content/drive/MyDrive/yolov5/yolov5s_results2/weights/best.pt' #
--project /content/drive/MyDrive/ --name yolov5s_results1
```

Epochs : 전체 데이터에 대해 학습을 반복하는 횟수



```
#학습시키기
```

%%time

%cd /content/yolov5/

!python train.py --img 416 --batch 32 --epochs 22 --data ./Hard-Hat-Workers-13/data.yaml # --cfg ./models/yolov5s.yaml --weights '/content/drive/MyDrive/yolov5/yolov5s_results2/weights/best.pt' # --project /content/drive/MyDrive/ --name yolov5s_results1

- ▼ myolov5
 - ▼ Hard-Hat-Workers-13
 - ▶ **test**
 - ▶ train
 - valid
 - README.dataset.txt
 - README.roboflow.txt
 - data.yaml

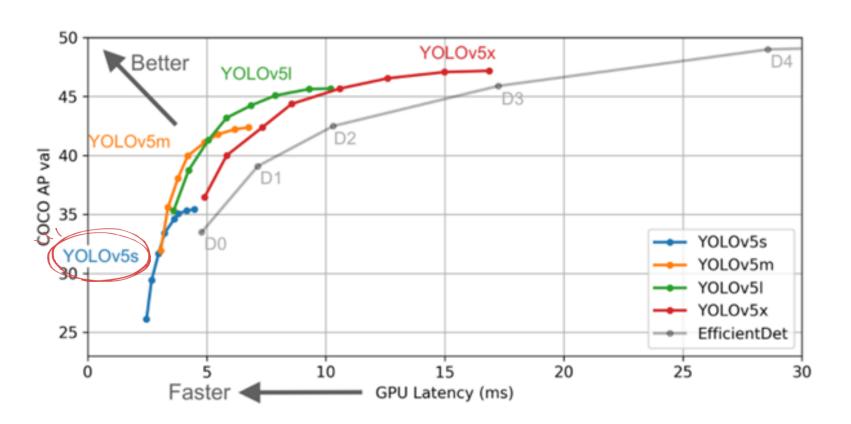
- 1 names:
- 2 head
- 3 helmet
- 4 nc: 2
- train: Hard-Hat-Workers-13/train/images
- 6 val: Hard-Hat-Workers-13/valid/images



#학습시키기 %%time %cd <u>/content/yolov5</u>/ !python train.py --img 416 --batch 32 --epochs 22 --data ./Hard-Hat-

--cfg .<u>/models/yolov5s.yaml</u>--weights '<u>/content/drive/MyDrive/yolov5</u>

--project /content/drive/MyDrive/ --name yolov5s_results1





- ▶ Hard-Hat-Workers-13
- data 🖿
- ▼ models
 - hub 🖿
 - init__.py
 - common.py
 - experimental.py

<u>)t</u>'₩

- tf.py
- yolo.py
- yolov5l.yaml
- yolov5m.yaml
- yolov5n.yaml
- yolov5s.yaml
- yolov5x.yaml



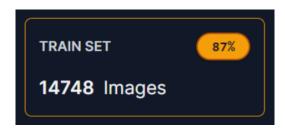
```
#학습시키기
%%time
%cd /content/yolov5/
!python train.py --img 416 --batch 32 --epochs 22 --data ./Hard-Hat-Workers-13/data.yaml #
--cfg ./models/yolov5s.yaml --weights ' ' #
--project /content/drive/MyDrive/ --name yolov5s_results1
```



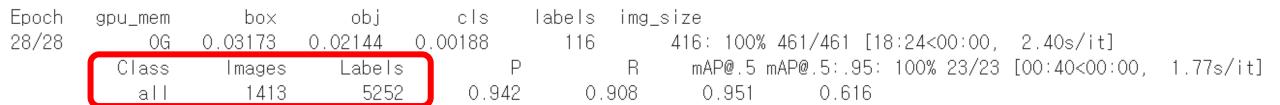
```
#학습시키기
%%time
%cd /content/yolov5/
!python train.py --img 416 --batch 32 --epochs 22 --data ./Hard-Hat-Workers-13/data.yaml \#
--cfg ./models/yolov5s.yaml --weights '/content/drive/MyDrive/yolov5/yolov5s_results2/weights/best.pt' \#
--project /content/drive/MyDrive/ --name yolov5s_results1
 /Content/drive/MyDrive/yolov5s_results1
                                                           내 드라이브 > yolov5s_results1 ▼
🔼 드라이브
                         Q 드라이브에서 검색
                                                                 weights
                                                                 hyp.yaml
                          내 드라이브 ▼
 十 신규
                                                                 opt.yaml
                                              수정 날짜
                           유형
                                    사람
▶ ▲ 내 드라이브
                                                                 labels correlogram.jpg
▶ [ 컴퓨터
                             yolov5s_results1
                                                                 labels.jpg
                                                                 events.out.tfevents.1698117115.ab8d677ea5e9.6679.0
                                                                 results.csv
```

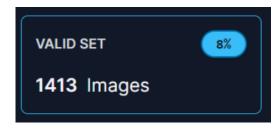
Epoch 28/28

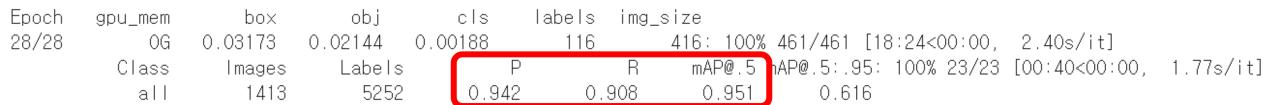
gpu_mem	box	obj	cls	labels	img_size				
OG	0.03173	0.02144	0.00188	116	416: 100)% 461/461 [18:24<00:00,	2.40s/it]	
Class	Images	Labels	Р		R mAP@.5	5 mAP@.5:.95	: 100% 23/23	[00:40<00:00]	1.
all	1413	5252	0.942	0.9	08 0.95 ⁻	0.616			

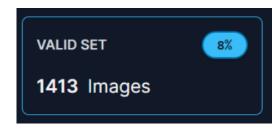


$$14748 \div 32 = 460.875 \approx 461$$

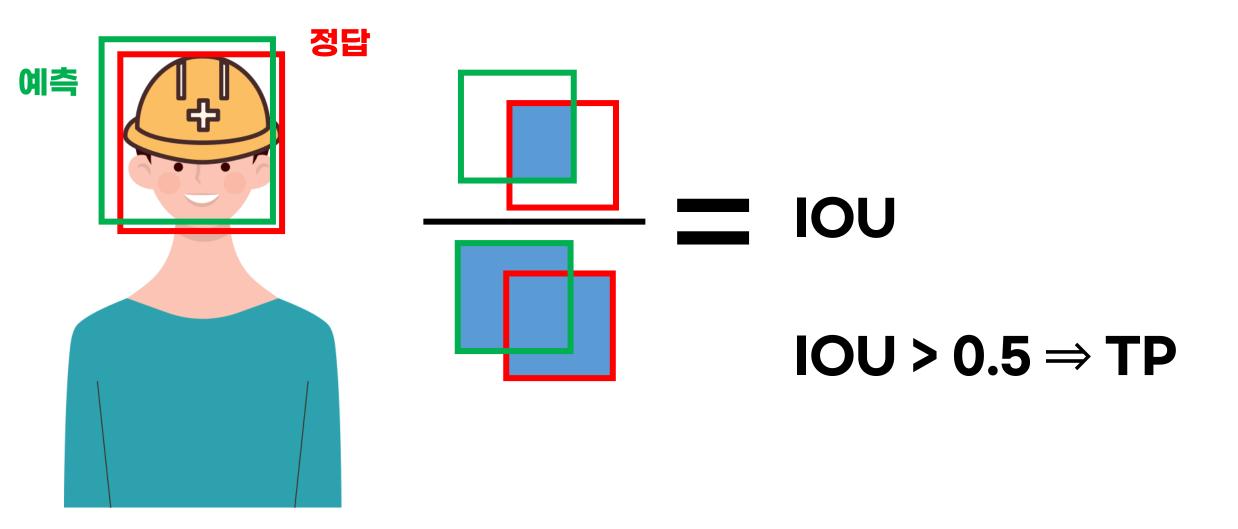


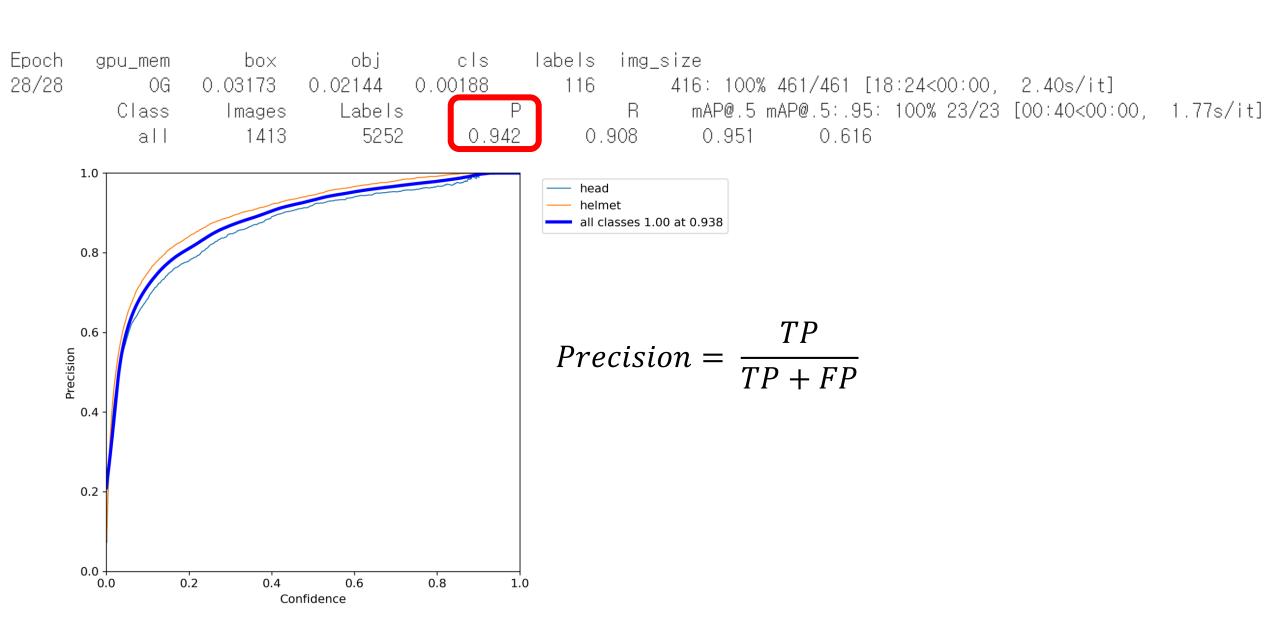






07-1 성능 평가 지표





0.0

0.2

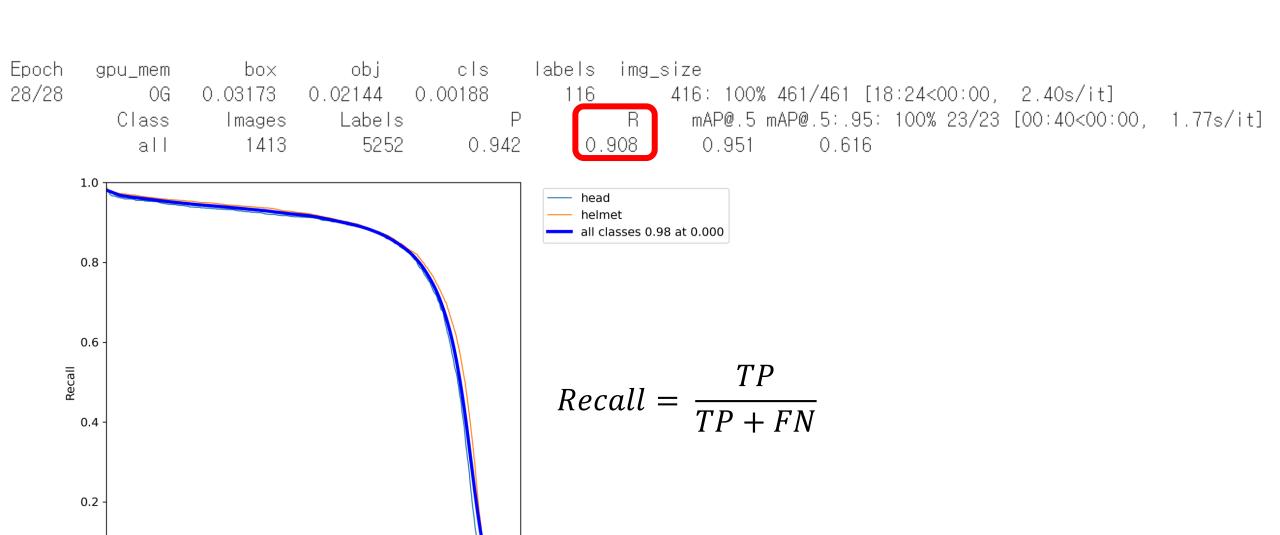
0.4

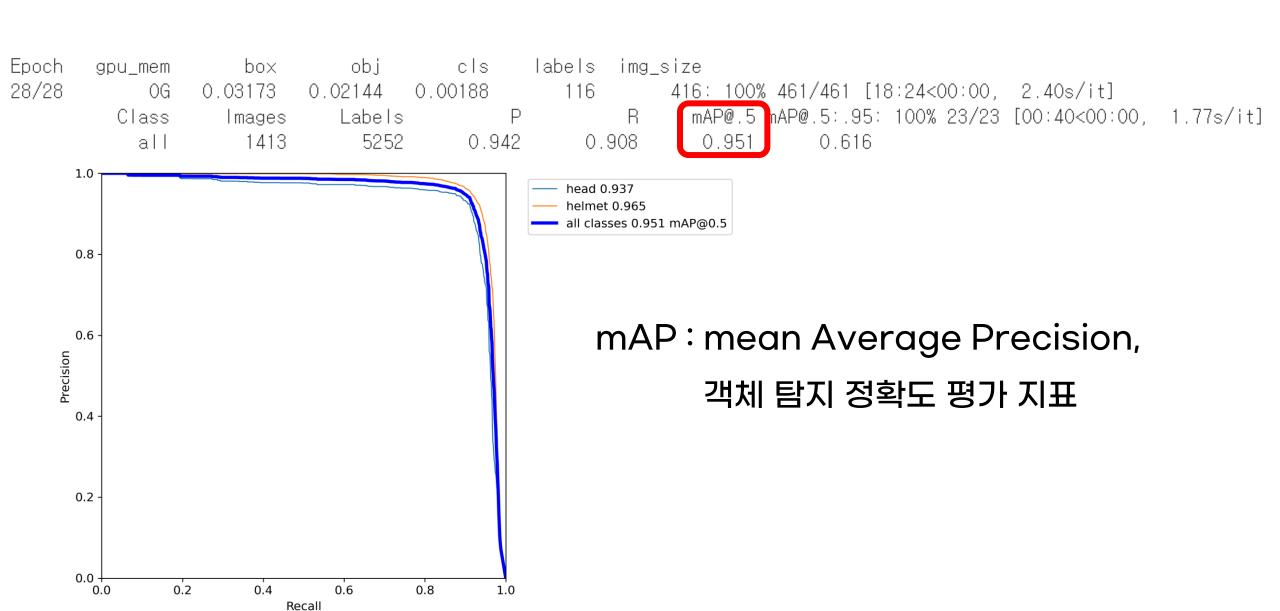
Confidence

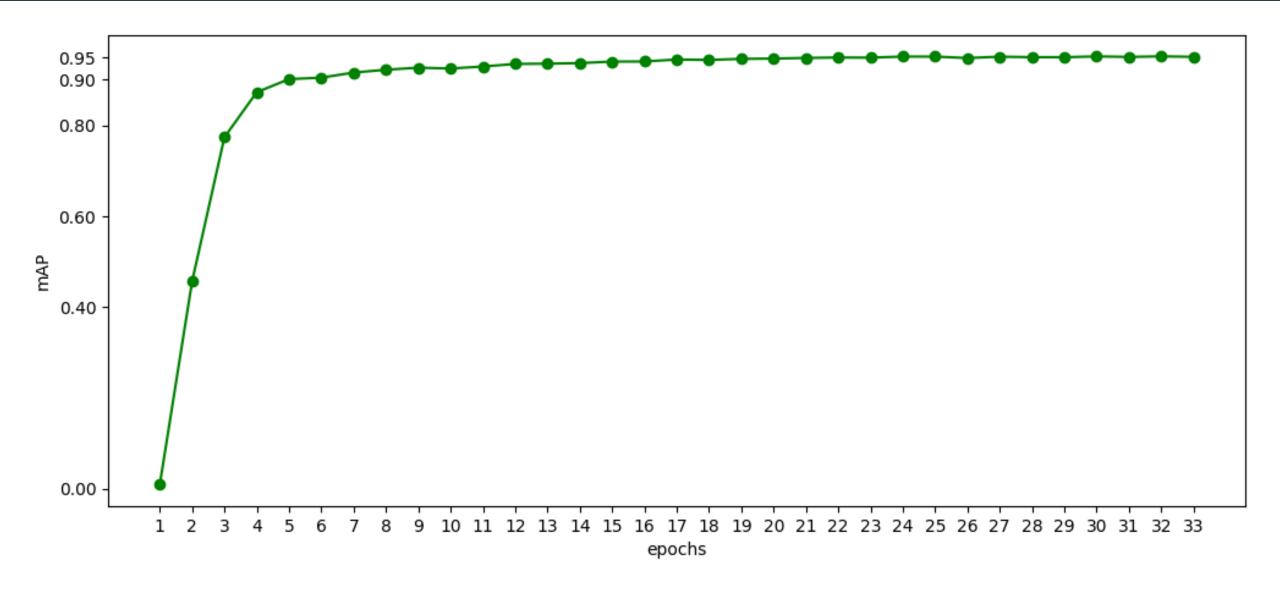
0.6

0.8

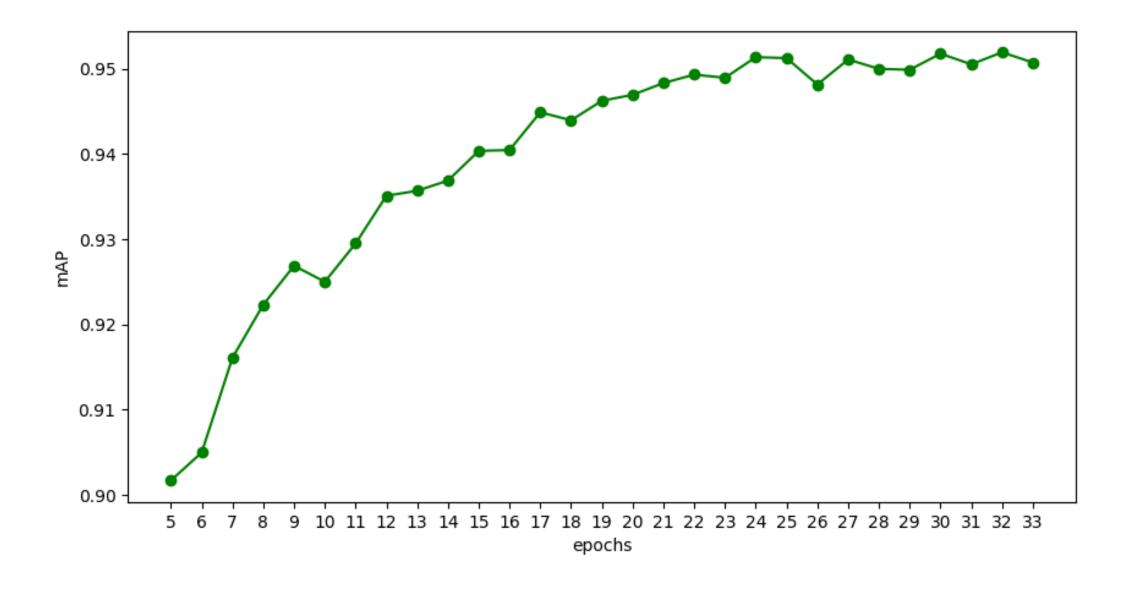
1.0



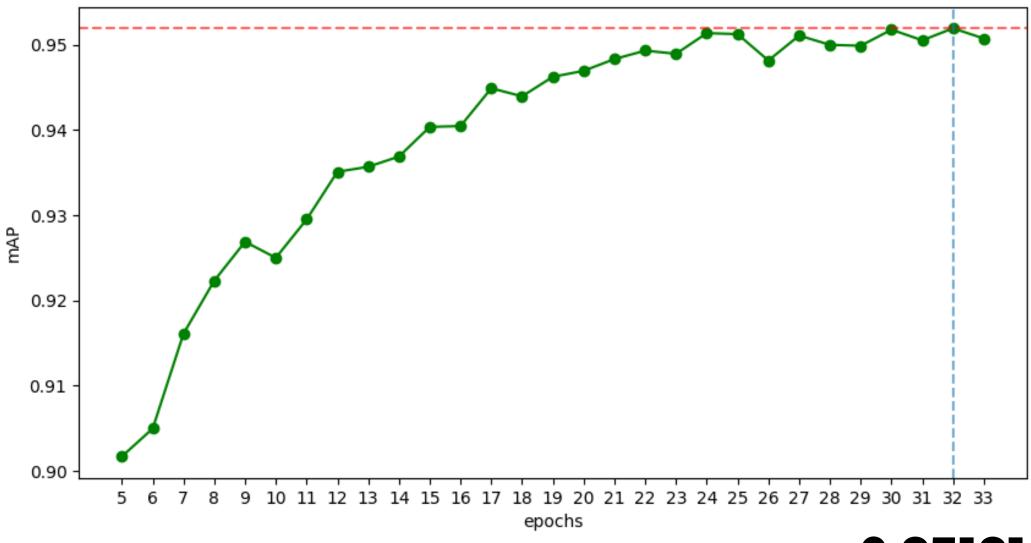




07 학습 결과



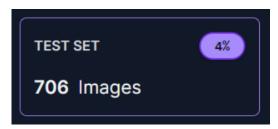
07 학습 결과



0.95191



!python /content/yolov5/val.py --task test --data /content/yolov5/Hard-Hat-Workers-13/data.yaml # --weights /content/drive/MyDrive/yolov5s_results2/weights/best.pt --img 416 --save-txt --save-conf





!python <u>/content/yolov5/val.py</u> --task test --data <u>/content/yolov5/Hard-Hat-Workers-13/data.yaml</u> # --weights <u>/content/drive/MyDrive/yolov5s_results2/weights/best.pt</u> --img 416 --save-txt --save-conf

val: data=/content/yolov5/Hard-Hat-Workers-13/data.yaml, weights=['/content/drive/MyDrive/yolov5s_r YOLOv5

✓ v6.1-306-gfbe67e4 Python-3.10.12 torch-2.1.0+cu118 CPU

Fusing layers...

YOLOv5s summary: 213 layers, 7015519 parameters, 0 gradients, 15.8 GFLOPs

test: Scanning '/content/yolov5/Hard-Hat-Workers-13/test/labels' images and labels...706 found, 0 m

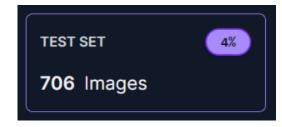
test: New cache created: /content/yolov5/Hard-Hat-Workers-13/test/labels.cache

Class	Images	Labels	Р	R	mAP@.5 m/	P@.5:.95:	100% 23/23
all	706	2641	0.94	0.913	0.959	0.622	
head	706	726	0.938	0.915	0.952	0.627	
helmet	706	1915	0.942	0.912	0.967	0.618	

Speed: 0.2ms pre-process, 33.1ms inference, 0.8ms NMS per image at shape (32, 3, 416, 416)

Results saved to runs/val/exp4

706 labels saved to runs/val/exp4/labels





!python <u>/content/yolov5/detect.py</u> --img 416 --conf 0.5 --source <u>/content/11.jpg</u> # --weights <u>/content/drive/MyDrive/yolov5/yolov5s_results2/weights/best.pt</u>



!python <u>/content/yolov5/detect.py</u> --img 416 --conf 0.5 --source <u>/content/11.jpg</u> # --weights <u>/content/drive/MyDrive/yolov5/yolov5s_results2/weights/best.pt</u>

Results saved to runs/detect/exp

yolov5
data
models
runs
detect
exp

display(Image(filename='/content/yolov5/runs/detect/exp/11.jpg'))



































09 결론 및 보완점





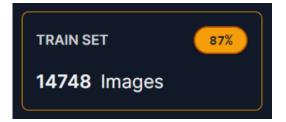


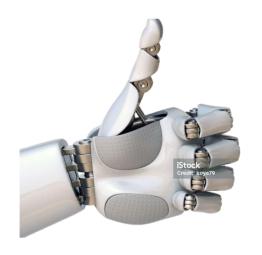












09 결론 및 보완점

난간 없는 현장, 안전모 안쓴 작업자··· 사망사고 60%는 '기본'을 안지킨 대가

2023.01.02 머니투데이

공유 전기자전거 이용자 99% 안전모 미착용

2023.03.09 소비자를 위한 신문

전동킥보드 사고 환자 85% '헬멧 미착용'

2022.06.13 메디칼업저버

10 참고자료

- 이미지 출처 : ppt 슬라이드 우측 하단 링크주소
- 장석원, 백재민, 강민석. 2021; 「유도무기 영상의 회전 객체 탐지를 위한 분류 및 회귀방법의 통합적 접근」.
 [한국전자파학회논문지], 32(12), 1099-1107
- 고광현, 김민철, 이승환, 유지나, 김재현. 2022;「YOLOv5를 사용한 수영모 착용 여부 판별 알고리즘」.
 [한국통신학회 학술대회논문집], 2022(11), 910-911
- 김도희, 김명호. 2023; 「객체 탐지 성능 향상을 위한 생성형 인공지능 기반 데이터 증강 기법 연구」. [한국컴퓨터정보학회 학술발표논문집], 31(2), 51-54
- 김준용. 2023; 「객체탐지모델 YOLO의 버전별 특성 비교 연구」. [한국컴퓨터정보학회 학술발표 논문집], 31(2), 75-78

감사합니다