Deep Learning : Helmet Detection

딥러닝 오픈소스를 활용한 안전모 착용 감지 시스템

20160402 최성원
20190256 주우성

목차 A table of contents

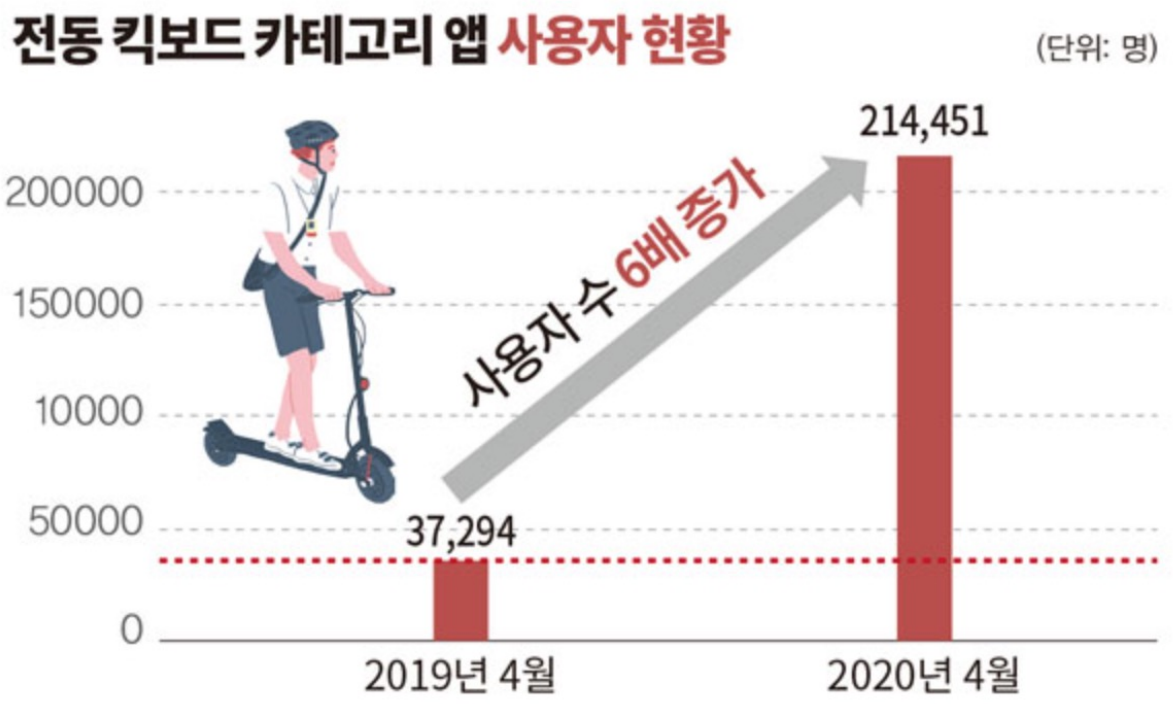
Part 1,

Aim & Background

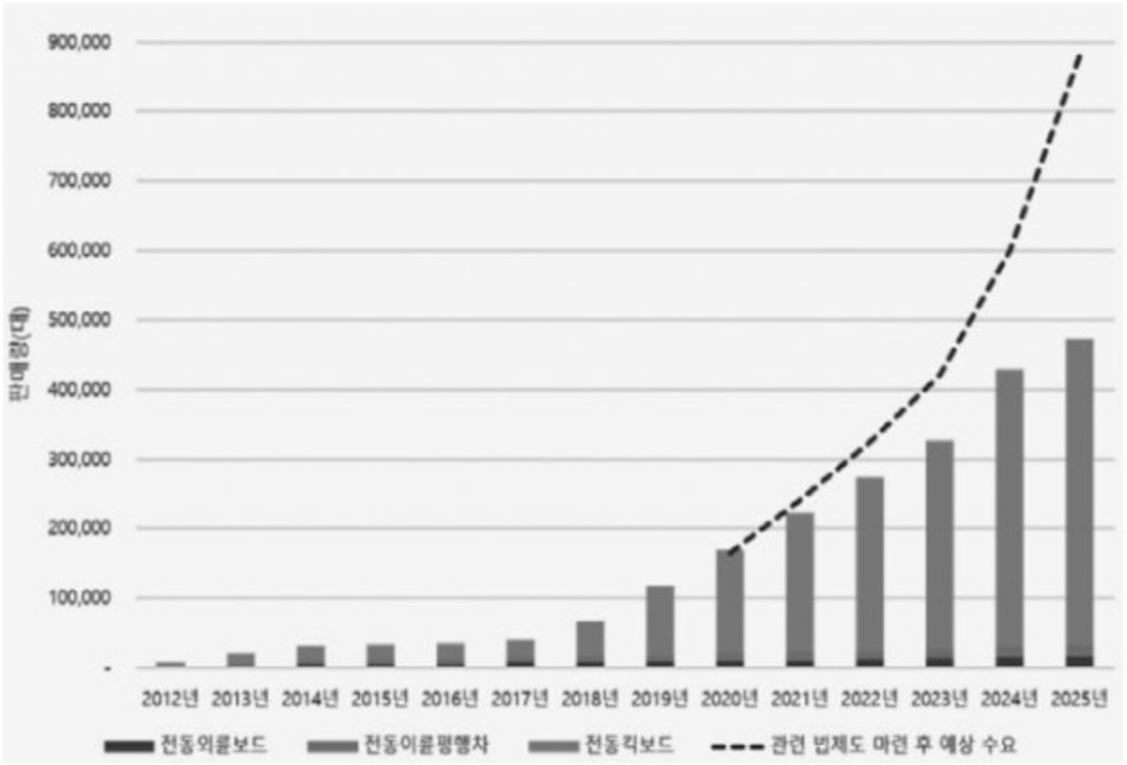
프로젝트 목표와 배경


Lime
Lime
Lime
Lime
Lime
Lime

1.1 안전모 착용의 중요성

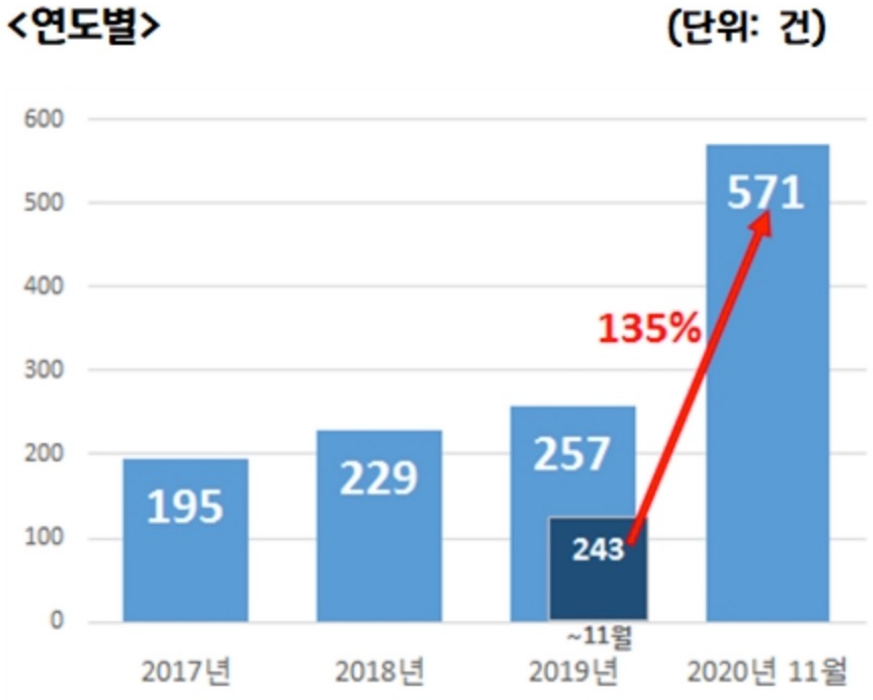


전동 킥보드 사용자 현황.



'퍼스널 모빌리티 현황 및 쟁점사항'

1.1 안전모 착용의 중요성



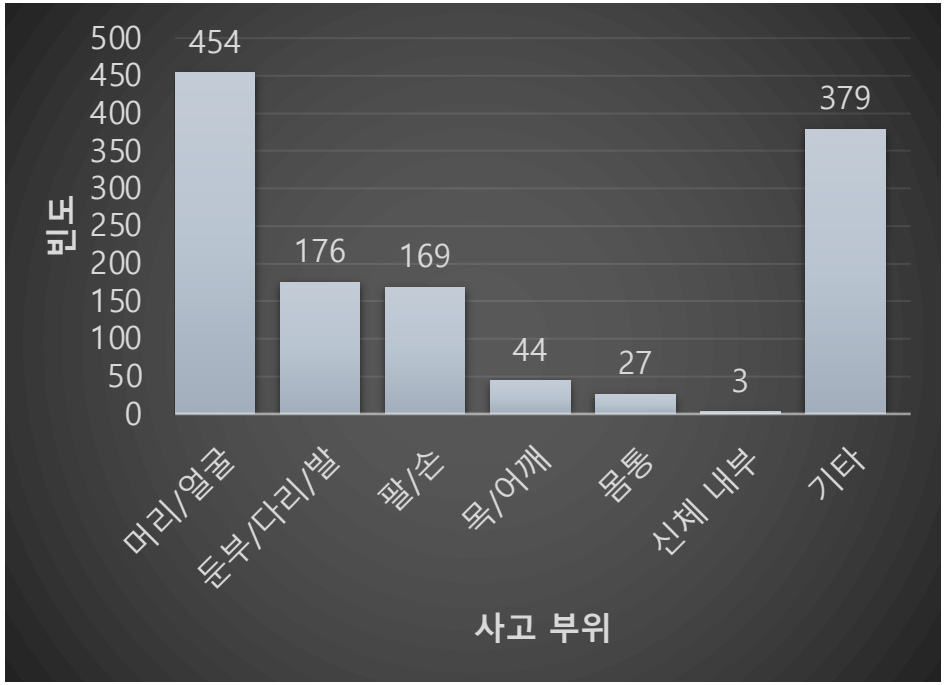
〈 최근 3년 11개월('17~'20.11월)간 전동킥보드 사고 현황 〉

19′ – 20′ 수요자가 급증하며

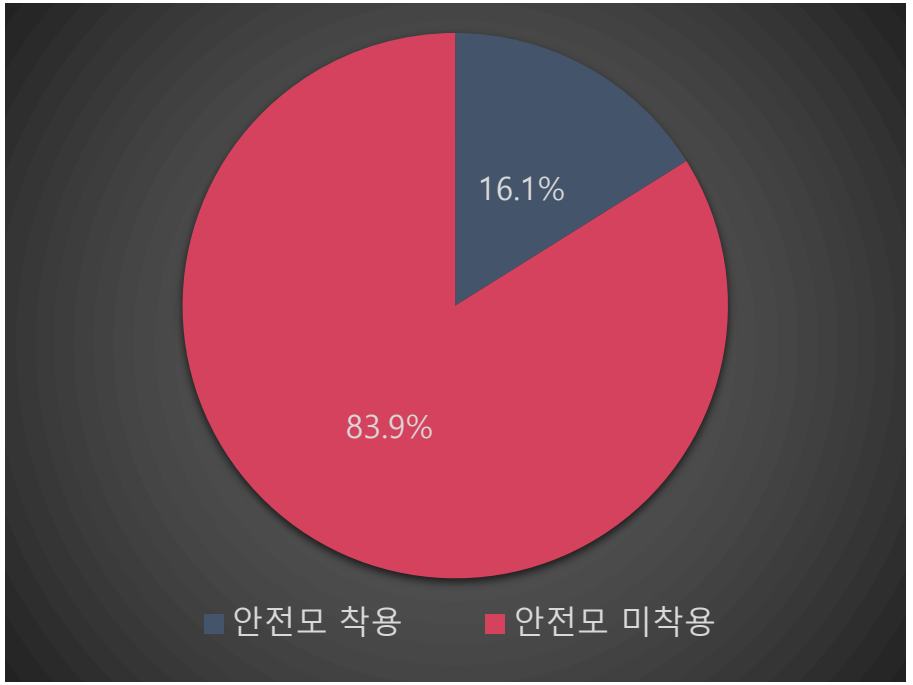
사고율이 135% 증가

1.1 안전모 착용의 중요성

사고에 따른 위해부위별 빈도



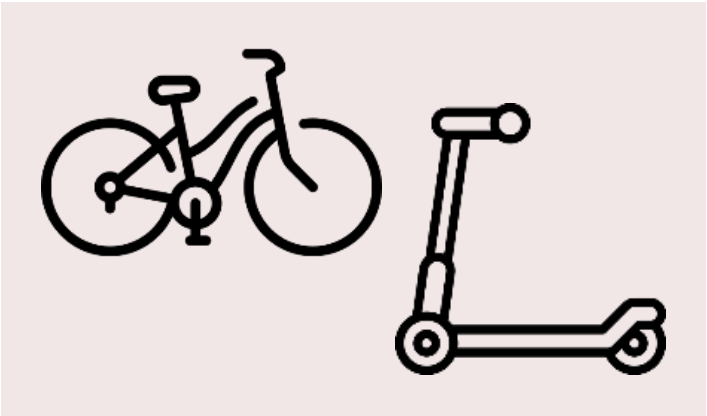
안전모 착용 준수율(~5/26)



1.2

목표 / 핵심내용

1

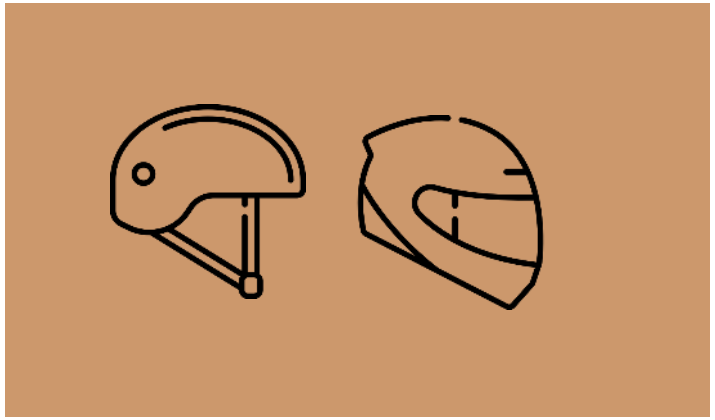


개인형 이동장치 대상

2

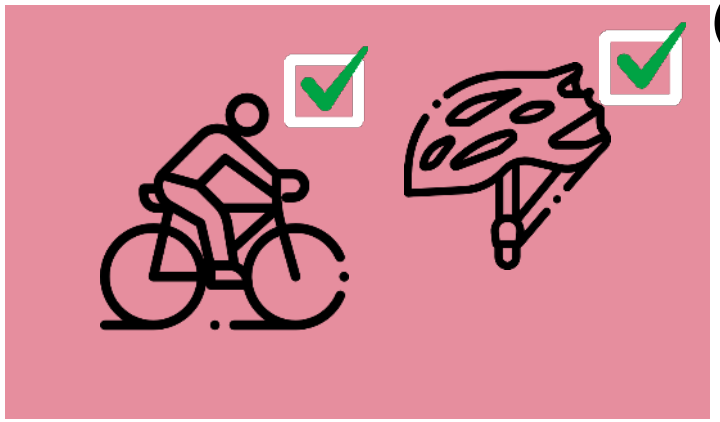
운전자 탑승

3



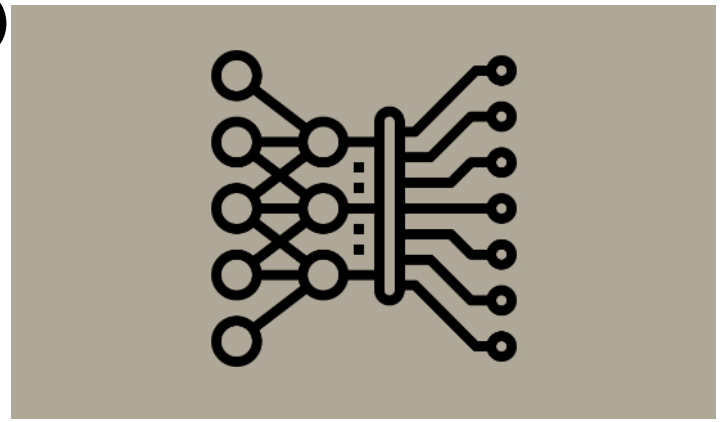
안전모 착용 확인

4



운전자와 안전모를 식별

5



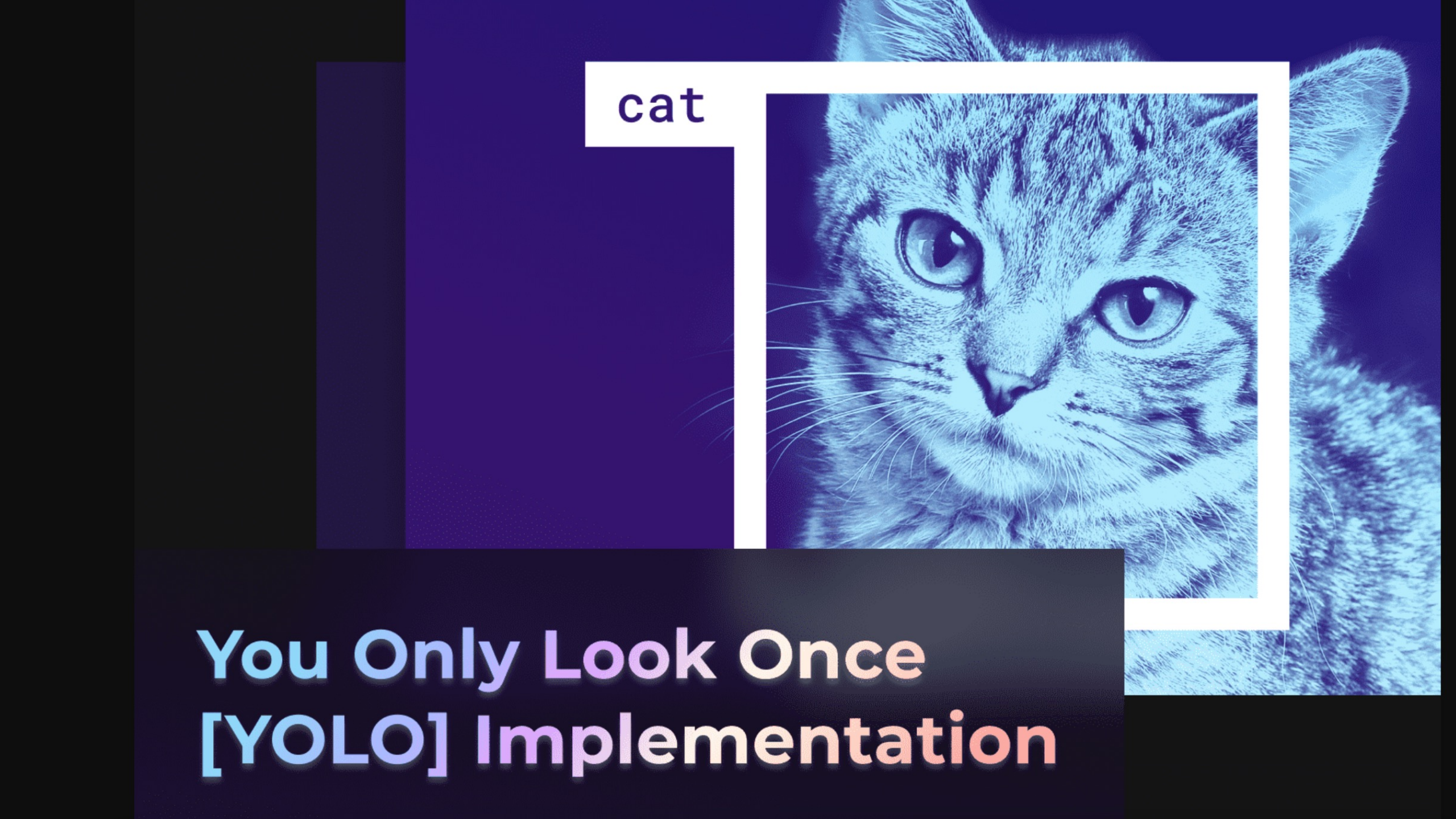
학습된 데이터셋

6

경고음 출력

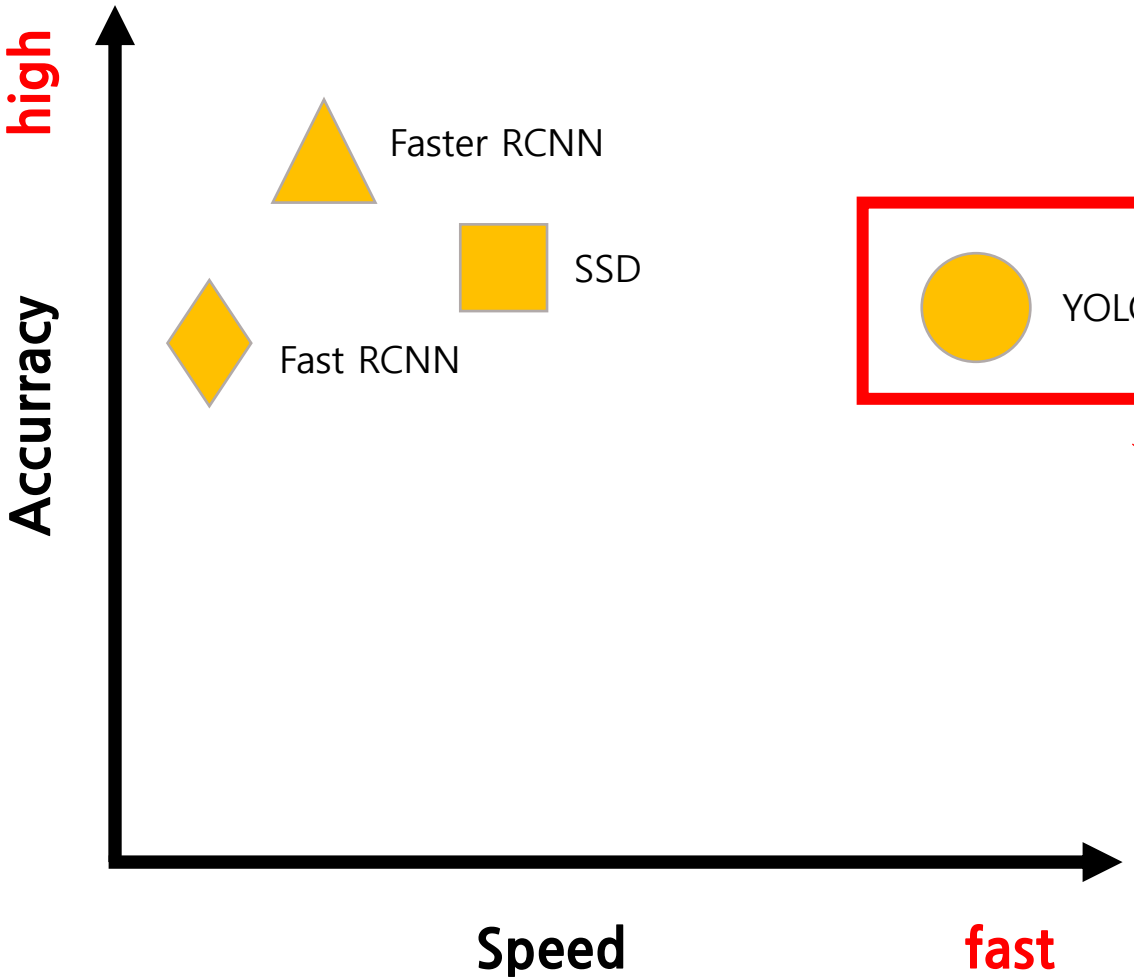
Part 2,

Contents



You Only Look Once [YOLO] Implementation

2.1 Yolo Detection



Accuracy : normal
Speed : fast

2.2

Bounding image

2.2

Bounding image

2.2

Bounding image

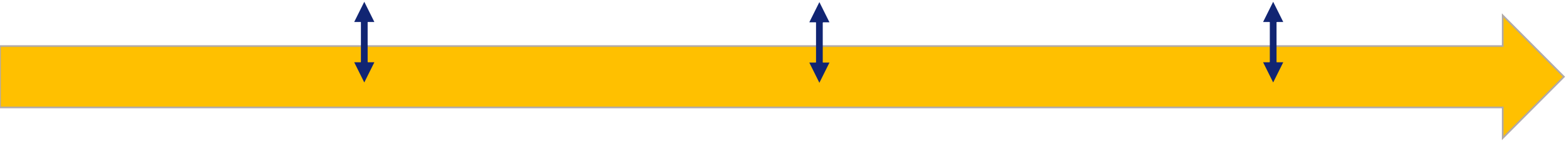
2.3

Detection

Image data
전처리

Without helmet
With helmet

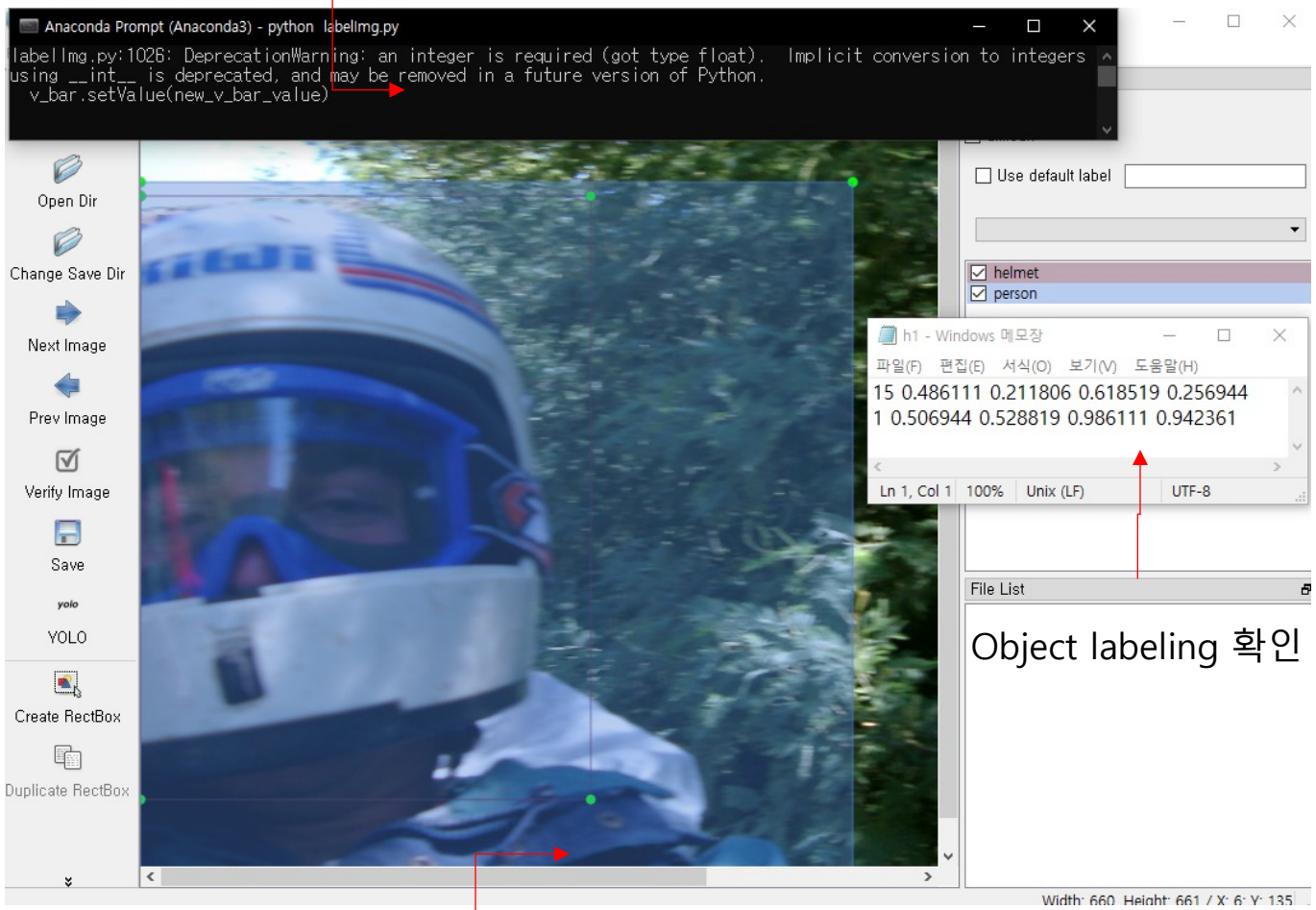
Image detected
Video detected
Realtime detected



Data - 전처리

Image data 수집

Python labelimg.py 실행



Create RectBox로 Object의 영역을 Bounding

Data labeling

2.3

Train – 데이터 학습

data.yaml

Detection.pptx | ! data.yaml | README.md M

lab1 > src > ! data.yaml

```yaml
train: ../train/images
val: ../valid/images

nc: 2
names: ['With Helmet', 'Without Helmet']
```

Dataset 경로

확인할 class의 개수와 이름

test

training

{image.jpg, label.txt}

validate

{image.jpg, label.txt}

Roboflow_Train.ipynb

Roboflow_Train_YOLOv5.ipynb M

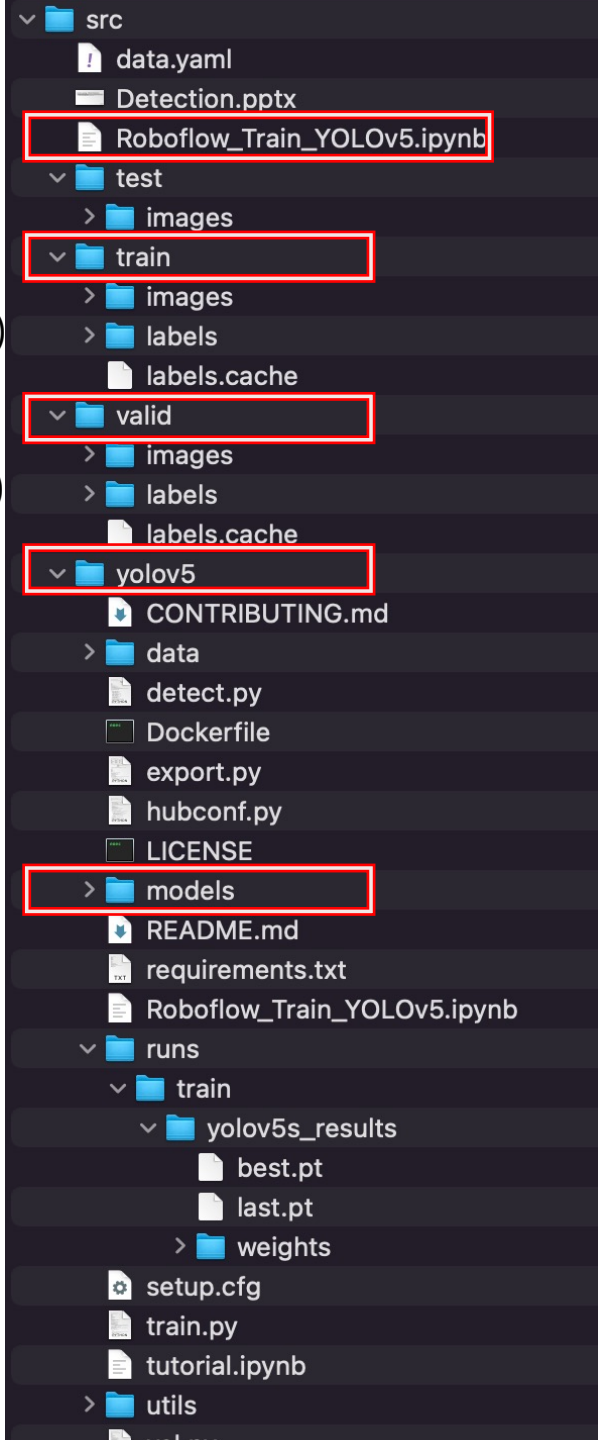
or > Next, we'll fire off training! > # train yolov5s on custom data for 100 epochs# time its perfor

+ Code + Markdown | Run All | Clear Outputs of All Cells | Outline ···

```
# train yolov5s on custom data for 100 epochs
# time its performance
%%time
%cd /content/yolov5/
!python train.py --img 416 --batch 16 --epochs 100 --data '../data.yaml' --cfg
./models/custom_yolov5s.yaml --weights '' --name yolov5s_results  --cache
```

N -> S -> M -> L -> X

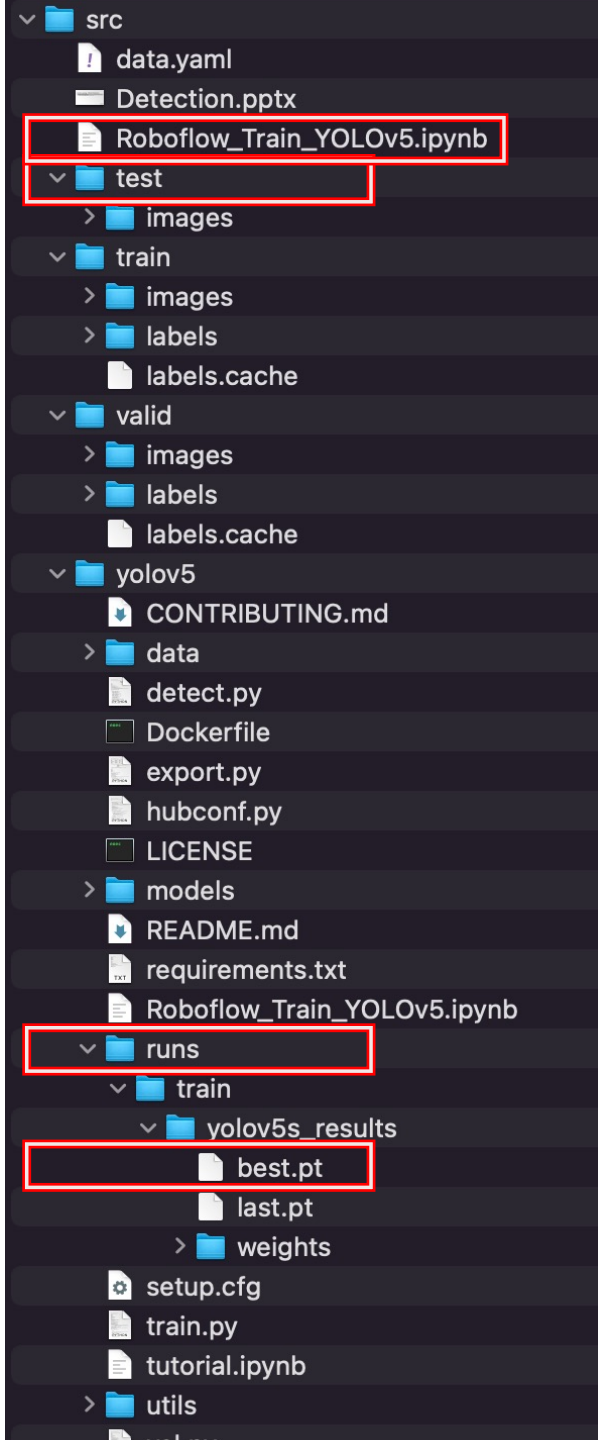
Trained Pytorch



2.3

Train - Result

```
# print out an augmented training example
print("GROUND TRUTH AUGMENTED TRAINING DATA:")
Image(filename='/content/yolov5/runs/train/yolov5s_results/train_batch0.jpg', width=900)
```



2.3

Test – Detecting

image Resize

train모델의 사이즈와 동일하게 test이미지의 크기를 맞춰줌

```
# image size
import glob
import cv2
path = glob.glob("../test/images/*.jpg")

for img in path:
    src = cv2.imread(img)
    dst = cv2.resize(src, dsize=(416, 416), interpolation=cv2.INTER_LINEAR)
    cv2.imwrite(img, dst)
print("finish")
```

test

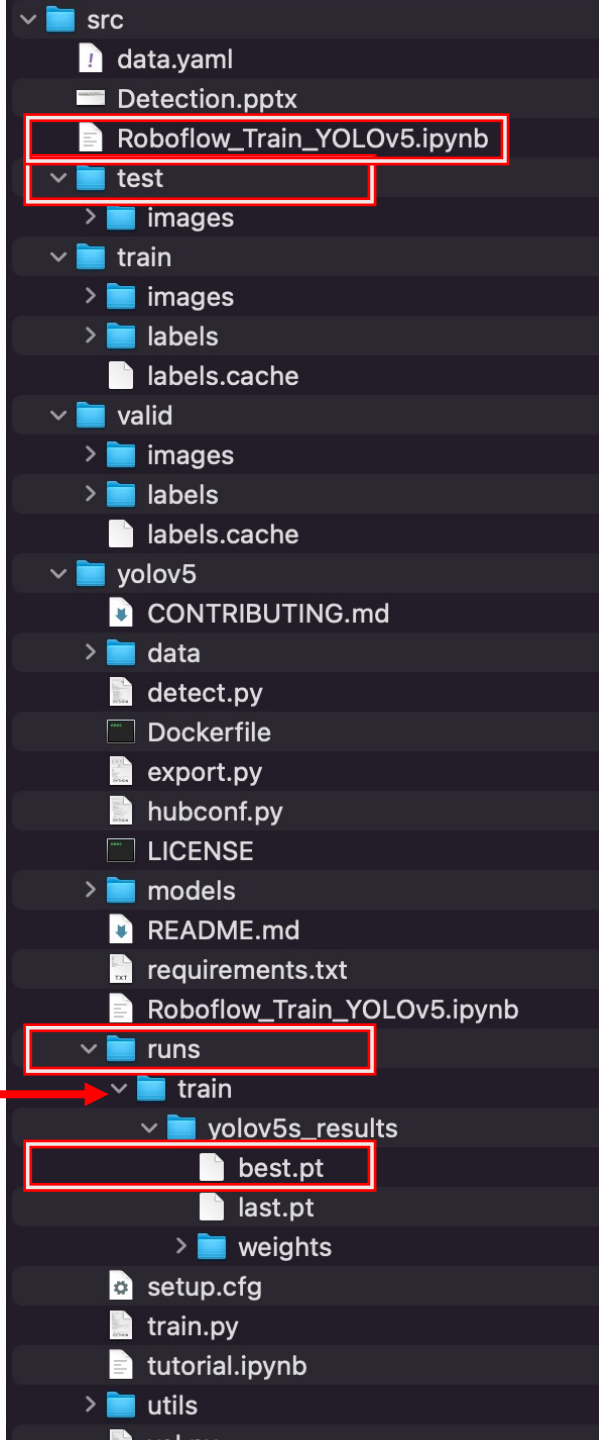
{image.jpg}

../test/images 존재하는 이미지 Detection

```
# when we ran this, we saw .007 second inference time. That is 140 FPS on a TESLA P100!
# use the best weights!
%cd /content/yolov5/
!python detect.py --weights runs/train/yolov5s_results/weights/best.pt --img 416 --conf 0.3 --source ../test/images
```

Detect Test image

Trained Pytorch



2.3

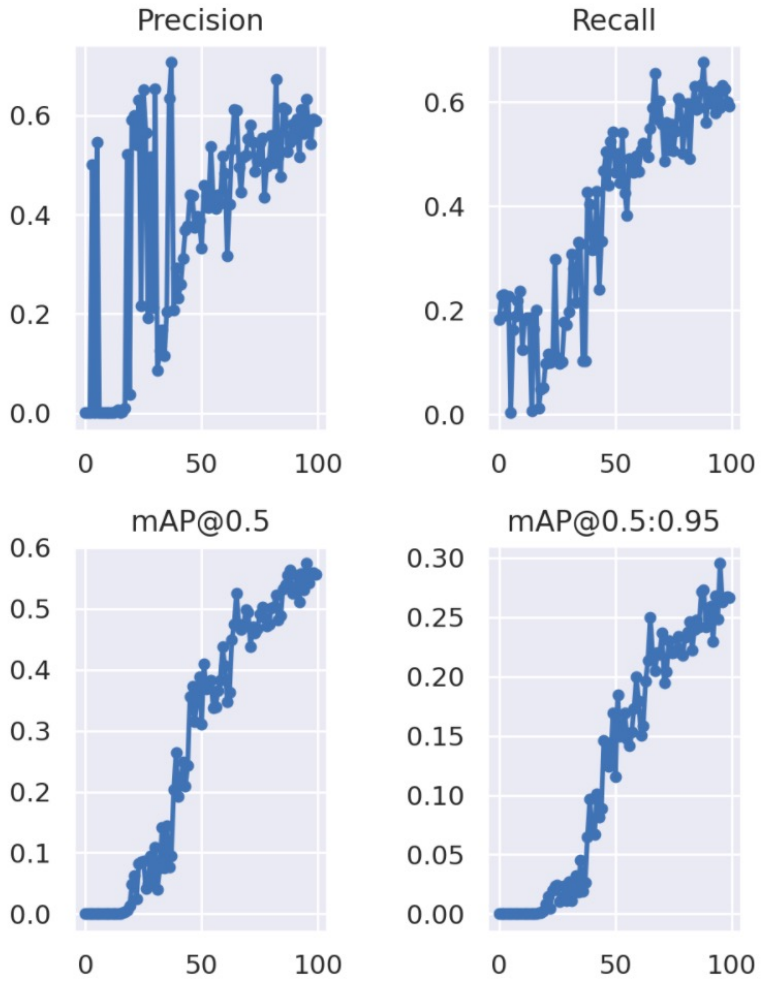
Test – Result

```
#display inference on ALL test images
#this looks much better with longer training above

import glob
from IPython.display import Image, display

for imageName in glob.glob('/content/yolov5/runs/detect/exp/*.jpg'): #assuming JPG
    display(Image(filename=imageName))
    print("\n")
```

Bounding된 test image 표시

Detecting Result

Performence



Precision : 검출 결과의 정밀도

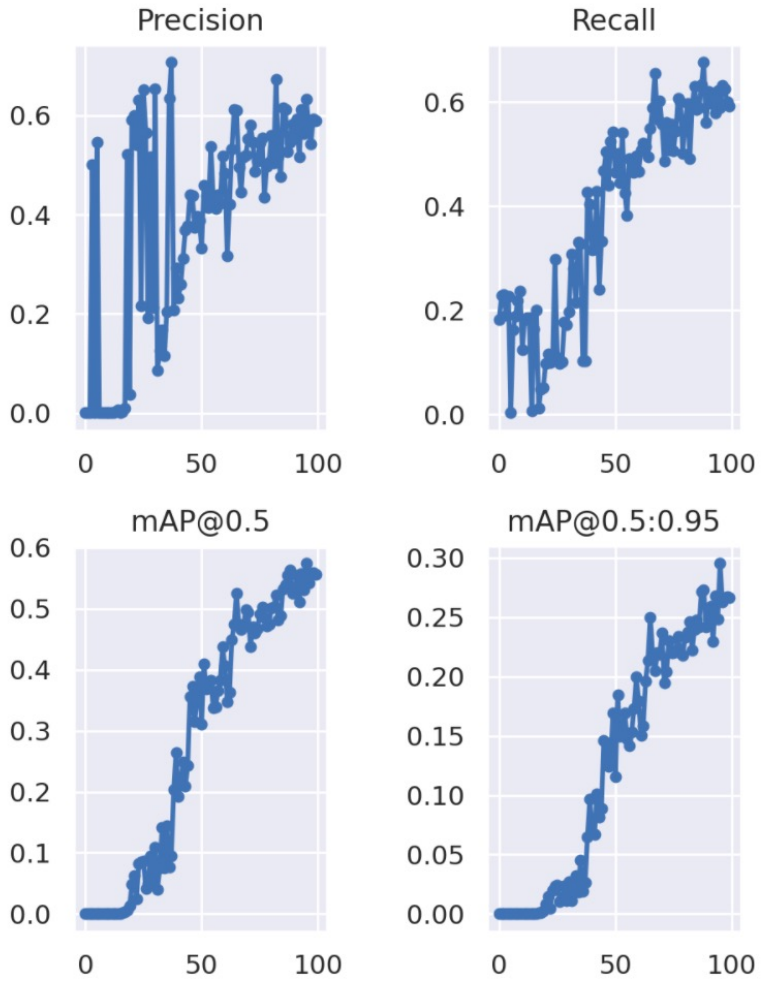
Recall : Detection 재현율

Mean Average Precision : AP의 평균

epoches를 반복 할 수록 정확한 결과를 얻을 수 있음

2.3

Performence

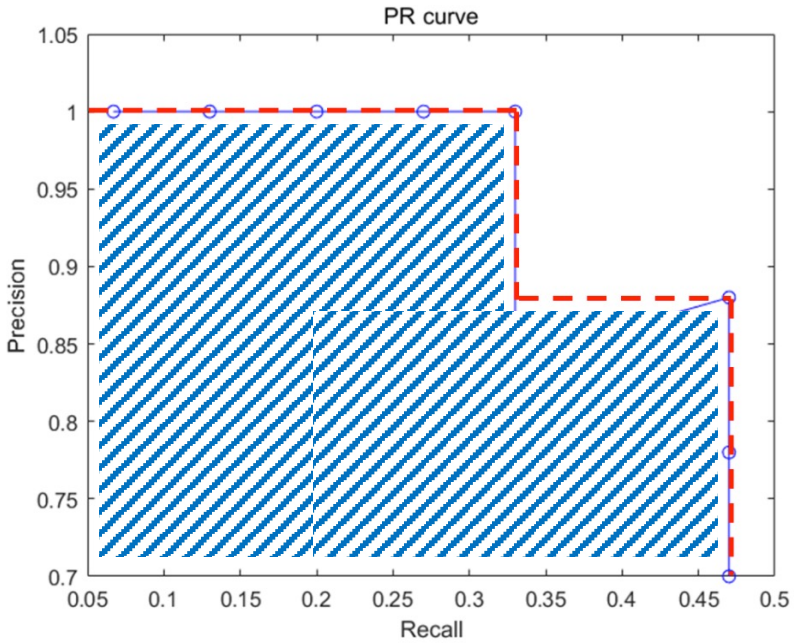


epoches

Precision : 검출 결과의 정밀도

Recall : Detection 재현율

Precision-Recall 곡선



Mean Average Precision : AP의 평균

epoches를 반복 할 수록 정확한 결과를 얻을 수 있음

Part 3,

Appendix

개선방안 및 향후 계획

3.1

문제점 및 개선 방안

mAP수치가 다른 모델에 비해 다소 낮음

GPU사양이 높은 머신에서 정확도가 보다 높은 YOLOv5m, l, x 모델을 사용

학습 데이터를 더욱 다양하게 적용하여 여러가지 상황에서도 Detecting 할 수 있도록 함

Data Augmentation 기법으로 학습 데이터에 인위적인 노이즈를 주어 데이터 수를 늘림

3.2

향후 계획

STEP 1		STEP 2		STEP 3		STEP 4
Nvidia Jetson에 적용. (Embedded)	> >	Without Helmet이 Detecting 될 경우 경고음이 울리도록 설계	> >	적용한 모델을 전동 킥보드의 손잡이 부분에 장착 할 수 있도록 함	> >	헬멧 뿐만 아니라 주변 객체 인식이 가능한(자동차, 또는 보행자) 모델을 구상해봄

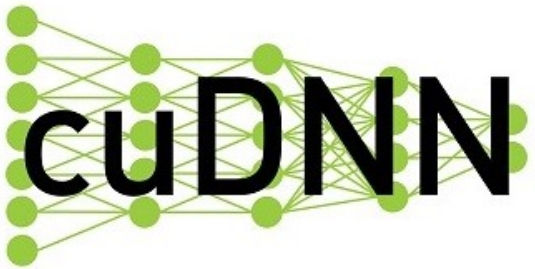
Part 4,

Reference

참고자료

4.1

Required



Reference

https://www.flaticon.com/
http://www.econovill.com/news/articleView.html?idxno=406417
https://www.donga.com/news/It/article/all/20210923/109363192/1
https://github.com/ultralytics/yolov5.git
https://github.com/tzutalin/labelImg.git
https://lapina.tistory.com/98
https://www.kaggle.com/
https://roboflow.com/
https://pytorch.org/

<< 역할분담 >>

주우성, 최성원 : 사례 분석 및 문제 인식
주우성, 최성원 : 데이터 수집 및 전처리
주우성 : 데이터셋 학습
최성원 : 학습된 데이터를 통해 Detect

감사합니다