

[샘플 테스트 - 라즈베리파이]

라즈베리파이 상에서 운용하기 위해 몇 가지 셋업이 필요합니다.

- HW : 27W 충전기, 카메라, hailo, 네트워크 연결
- SW : 가상환경, 관련 라이브러리 설치, Thonny 인터프리터 변경, 샘플 코드 작성

HW 의 경우 이전 카메라 검증 단계에서 모두 세팅을 완료하였으므로 바로 SW 세팅을 진행하였습니다.

1. 가상환경 설치 및 실행 (Pi OS 와 별도로 진행하는 것, 손상 위험 없이 진행가능)

```
pi@raspberrypi:~ $ python3 -m venv --system-site-packages yolo_object
```

=> 가상환경 관련 패키지 설치 및 경로 설정

```
pi@raspberrypi:~ $ source yolo_object/bin/activate
```

=> 가상환경 활성화

```
pi@raspberrypi:~ $ sudo apt update
sudo apt install python3-pip -y
pip install -U pip
```

=> 가상환경은 기존 환경과 별도의 개념이므로 라이브러리 최신화 진행

```
pi@raspberrypi:~ $ pip install ultralytics
```

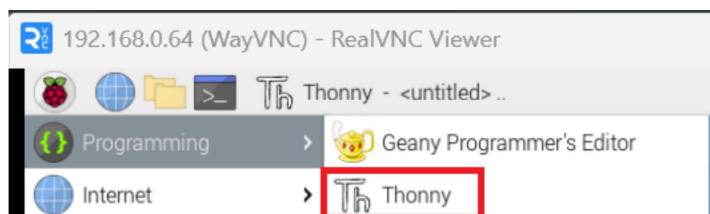
=> 컴퓨터 비전 작업을 지원하는 패키지 설치

```
pi@raspberrypi:~ $ pip install picamera2
```

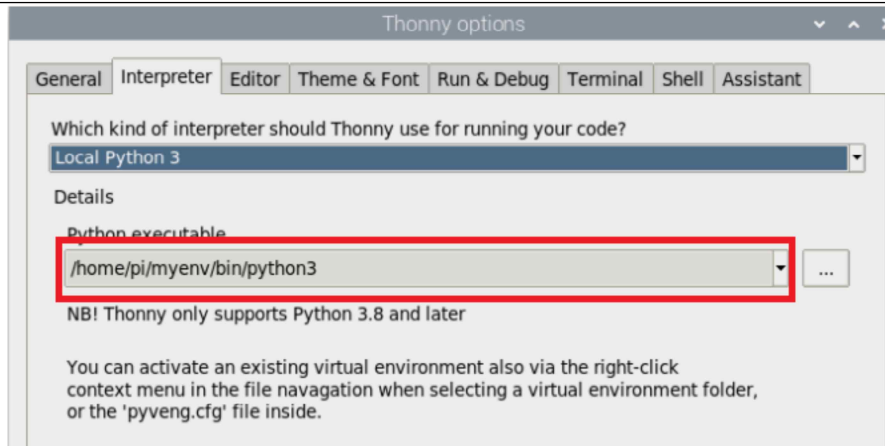
=> Picamera2 모듈 설치

```
pi@raspberrypi:~ $ sudo reboot
```

=> 재부팅 진행



=> 샘플 코드를 실행시킬 “Thonny” 실행



=> “Configuration Interpreter...” 실행 후, 가상환경에서의 인터프리터 경로로 재설정

```

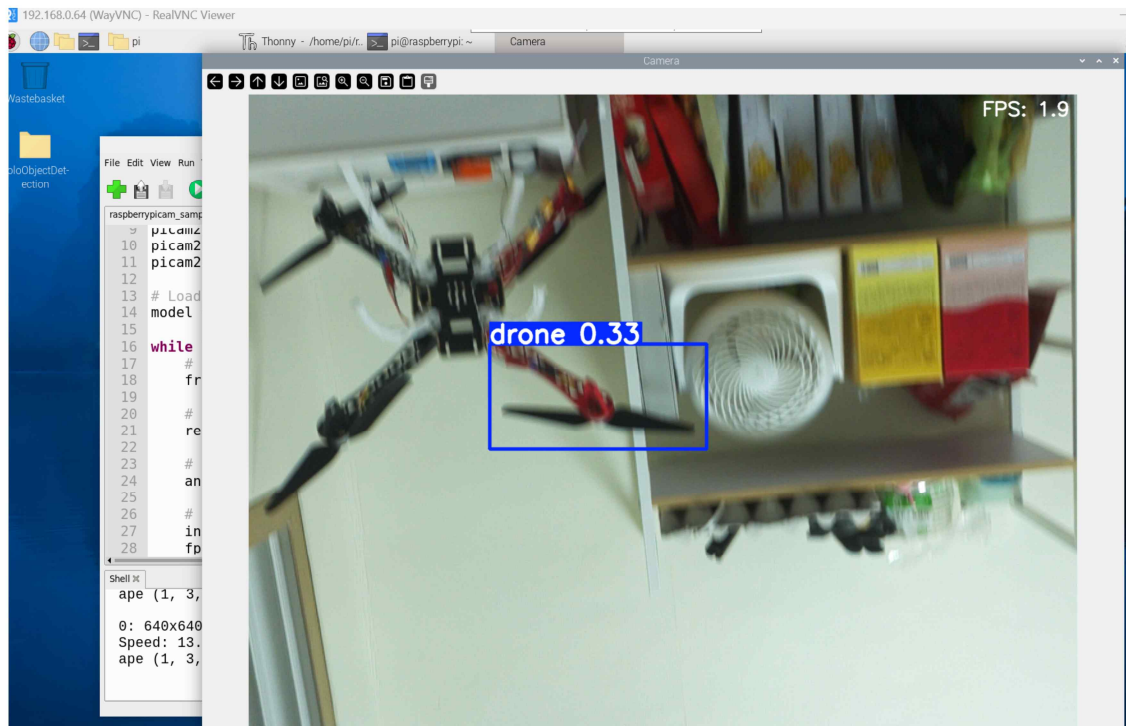
1  import cv2
2  from picamera2 import Picamera2
3  from ultralytics import YOLO
4
5  picam2 = Picamera2()
6  picam2.preview_configuration.main.size = (1280, 1280)
7  picam2.preview_configuration.main.format = "RGB888"
8  picam2.preview_configuration.align()
9  picam2.configure("preview")
10 picam2.start()
11
12 model = YOLO("/home/pi/best100.onnx")
13
14 while True:
15     frame = picam2.capture_array()
16     results = model(frame)
17     annotated_frame = results[0].plot()
18
19     inference_time = results[0].speed['inference']
20     fps = 10000 / inference_time
21     text = f'FPS: {fps:.1f}'
22
23     font = cv2.FONT_HERSHEY_SIMPLEX
24     text_size = cv2.getTextSize(text, font, 1, 2)[0]
25     text_x = annotated_frame.shape[1] - text_size[0] - 10
26     text_y = text_size[1] + 10
27
28     cv2.putText(annotated_frame, text, (text_x, text_y), font, 1, (255, 255, 255), 2, cv2.LINE_AA)
29     cv2.imshow("Camera", annotated_frame)
30
31     if cv2.waitKey(1) == ord("q"):
32         break
33
34 cv2.destroyAllWindows()
35
36

```

=> 샘플 코드 작성 (picam2 라이브러리 활용하여 실행)

만약 라이브러리 설치가 잘 되지 않은 경우, 미처 설치하지 못한 라이브러리가 있을 수 있으므로
가상환경과 Pi os 상의 라이브러리 동기화 작업을 진행해주면 될 것 같습니다.

[실행 화면]



=> 정상적으로 실행되는 모습, 단 샘플 모델이어서 인식율이 저조함

[학습 테스트 진행]

본 학습 테스트는 단순히 epochs 를 많이 두었을 시 과적합 현상이 발현할 수 있으므로

epochs 수를 점점 늘려가며 이에 비례한 인식율을 확인 후 어느정도의 학습량이 적절한지 파악하기 위해서 진행

테스트 과정은 아래와 같습니다.

1. epochs 양 조절 후 학습
2. 산출된 학습 데이터 시각화 자료 생성
3. 라즈베리파이로 파일 전송 후, 샘플 코드 기반 테스트

(setup) 이전 샘플 테스트로 진행하였던 코드 기반으로, 반복적으로 실행하여 학습시킬 수 있는 코드 작성

```
#학습 이어하기용 코드
from ultralytics import YOLO

last_checkpoint = '/content/drive/MyDrive/yolov8_training/drone_detection/weights/last.pt'

model = YOLO('yolov8n.pt')

results = model.train(
    data='/content/drive/MyDrive/dataset/data.yaml',
    epochs=10, # 총 에포크 수 (기존 것 누적 중)
    batch=16,
    imgsz=640,
    name='drone_detection2',
    project='/content/drive/MyDrive/yolov8_training',
    #resume=True # 이전 상태에서 이어서 학습
)
```

(setup) Colab 환경에서의 학습 시, 산출되는 데이터 수집 후 시각화 자료 작업

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size			
1/100	2.08G	1.746	2.573	1.281	7	640:	100%	201/201	[01:18<00:00, 2.58it/s]
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100%	26/26 [00:16<00:00, 1.55it/s]
	all	802	660	0.546	0.732	0.577	0.258		
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size			
2/100	2.4G	1.742	1.608	1.316	13	640:	100%	201/201	[01:14<00:00, 2.69it/s]
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100%	26/26 [00:15<00:00, 1.63it/s]
	all	802	660	0.667	0.674	0.708	0.347		

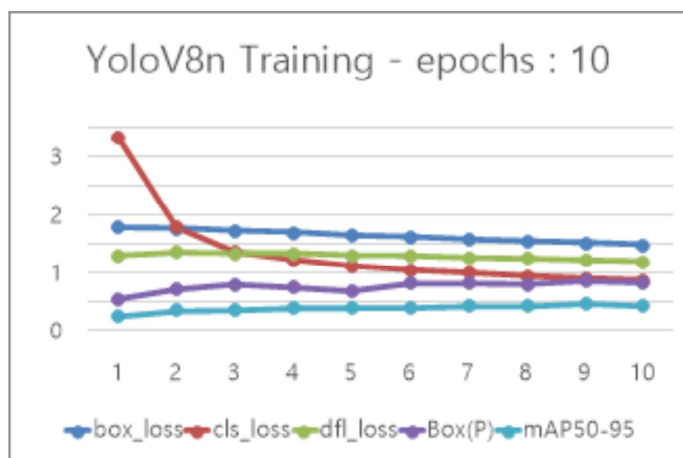
저는 위 데이터 중 (box_loss , cls_loss , dfl_loss , Box(P) , mAP50-95) 들을 선정하여 시각화 하였습니다.

(setup) 이 데이터들을

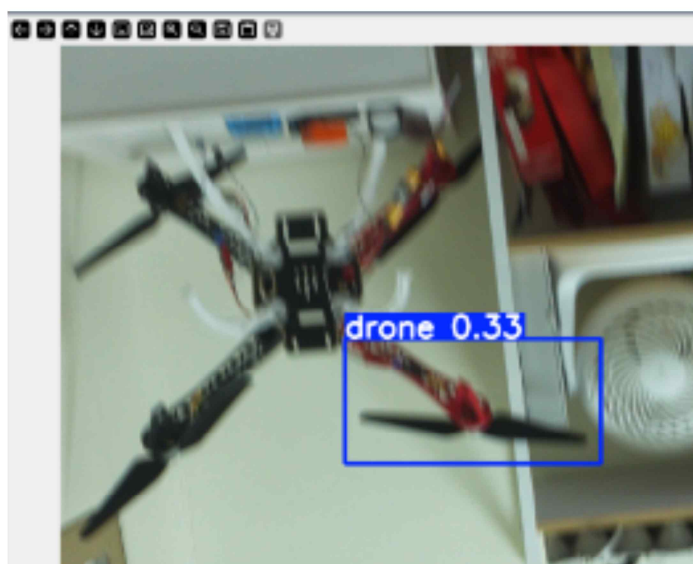
“Capstone_Design\1.PrecisionLandingModule\1.ReasearchNote\ResearchNoteData.xlsx” 의

“Test YoloV8n in RaspberryPi” sheet 에 정리하였습니다.

< 1 차 학습 - epochs 10 / 모델 학습 소요시간 : 10 분 >

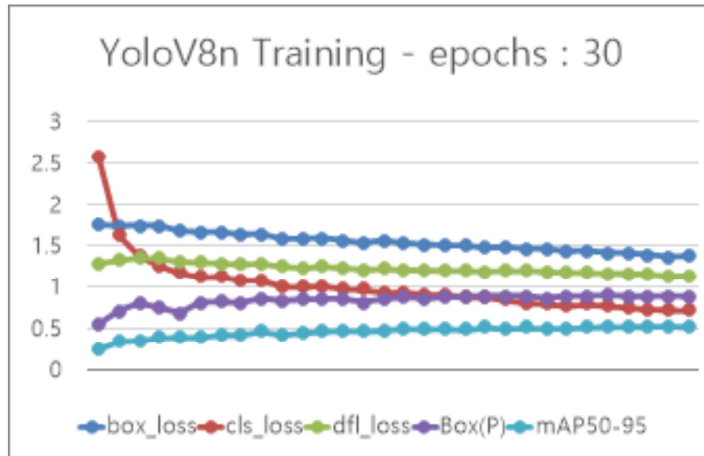


=> 학습 정확도가 급격히 변화하는 모습 (횡수 자체가 적어서 그런듯 합니다)

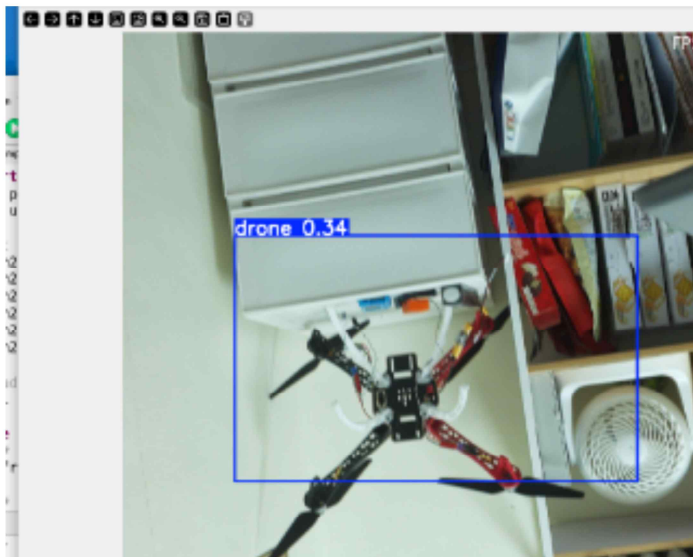


=> 실제 인식 사진

< 2 차 학습 - epochs 30 / 모델 학습 소요시간 : 40 분 >

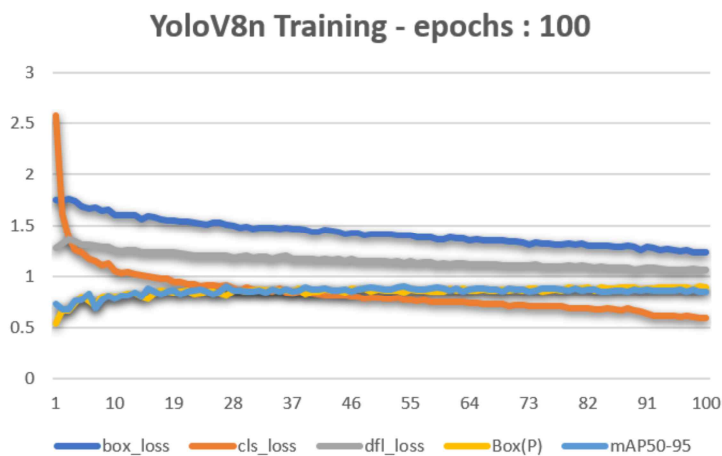


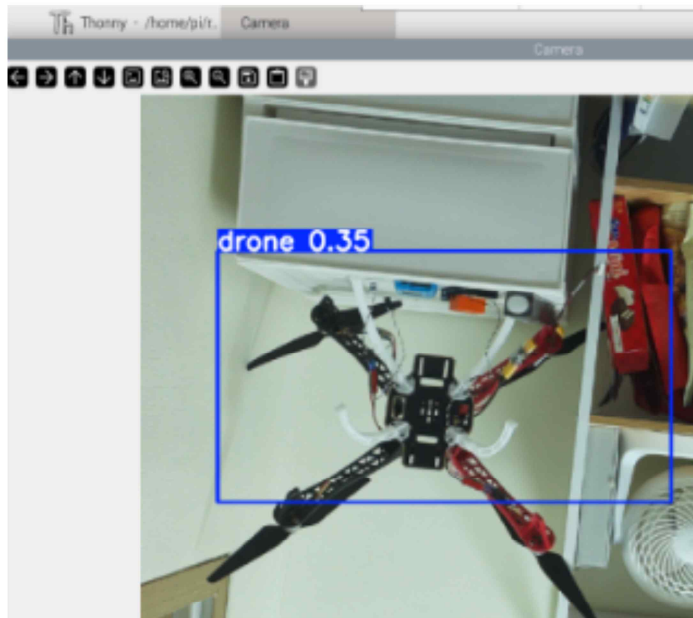
=> 횟수가 늘어갈 수록 인식율이 점점 좋아지긴 하나 누적 될 수록 변화량이 작은 것이 보임



=> 바운더리 박스와 전체적인 인식율이 많이 올라간 모습

< 3 차 학습 - epochs 100 / 모델 학습 소요시간 : 4 시간 >





=> 10 -> 30 에 비해 30 -> 100 은 큰 변화량이 없어 보임.