



YOLOv8 학습

YOLOv8 코드 자료

YOLOv8 학습 프로세스

- [1] Custom data
- [2] data 를 colab 으로 다운로드
- [3] yaml 파일 만들기
- [4] install YOLOv8
- [5] Train model
- [6] Prediction

실습

- [1] Custom data 구축
- [2] 데이터를 Colab 으로 다운로드
- [3] 커스텀 데이터에 맞는 YAML 파일 만들기
- [4] YOLOv8 라이브러리 설치
- [5] YOLOv8 커스텀 데이터 학습하기
- [6] 학습된 데이터로 테스트 이미지 예측

YOLOv8 코드 자료

<https://github.com/ultralytics/ultralytics>

YOLOv8 학습 프로세스

- 1) Custom data 준비
- 2) data 를 colab 으로 다운로드
- 3) yaml 파일 만들기
- 4) yolo 모델 학습
- 5) 예측

[1] Custom data

- 이미지와 어노테이션으로 이루어진 데이터를 준비한다.

- 커스텀 데이터는 주로 Roboflow 에서 제공하는 데이터를 이용하거나 직접 labeling 시킨 데이터를 구축해야한다.
- 이때, 이미지 데이터와 확장자를 제와한 파일 이름은 동일해야하며, 어노테이션 파일 즉 정답 파일의 확장자는 반드시 .txt 여야 함.



- 이미지 데이터는 bounding box 가 표시된 segmentation image
- 정답 데이터는 제일먼저 segmentation 하는 클래스의 종류가 무엇인지 나타내는 숫자 (class number), 이후의 소숫점으로 이루어진 조합은 segmentation 을 나타내는 위치 정보가 있는 파일임.

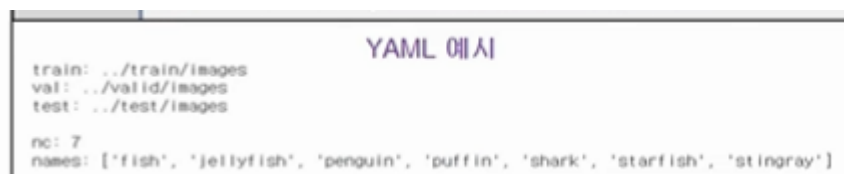
[2] data 를 colab 으로 다운로드

[3] yaml 파일 만들기

YOLOv8 으로 Custom Data 를 학습하기 위해서는 YAML 파일 필요

아래의 내용을 반드시 포함해야 함

- 이미지와 정답이 저장되어 있는 디렉토리 정보 → 학습 시켜야하는 데이터의 디렉토리 정보
- segmentation 하고 싶은 클래스 종류(몇개인지)와 대응되는 각각의 이름 → 클래스의 개수 정보, 클래스에 대응되는 name



[4] install YOLOv8

- `pip install ultralytics`
- yolov8 실행에 필요한 라이브러리 설치 및 dependency 체크

[5] Train model

train 을 통해 학습 시키기

- 사전학습모델 yolov8n-seg.pt 로드
- mydata.yaml 참조하여 학습(파인튜닝)

Train model	
<pre>from ultralytics import YOLO model = YOLO('yolov8n-seg.pt') model.train(data='mydata.yaml', epochs=10)</pre>	<pre># 사전학습모델 yolov8n-seg.pt 로드 # mydata.yaml 참조하여 학습(파인튜닝)</pre>

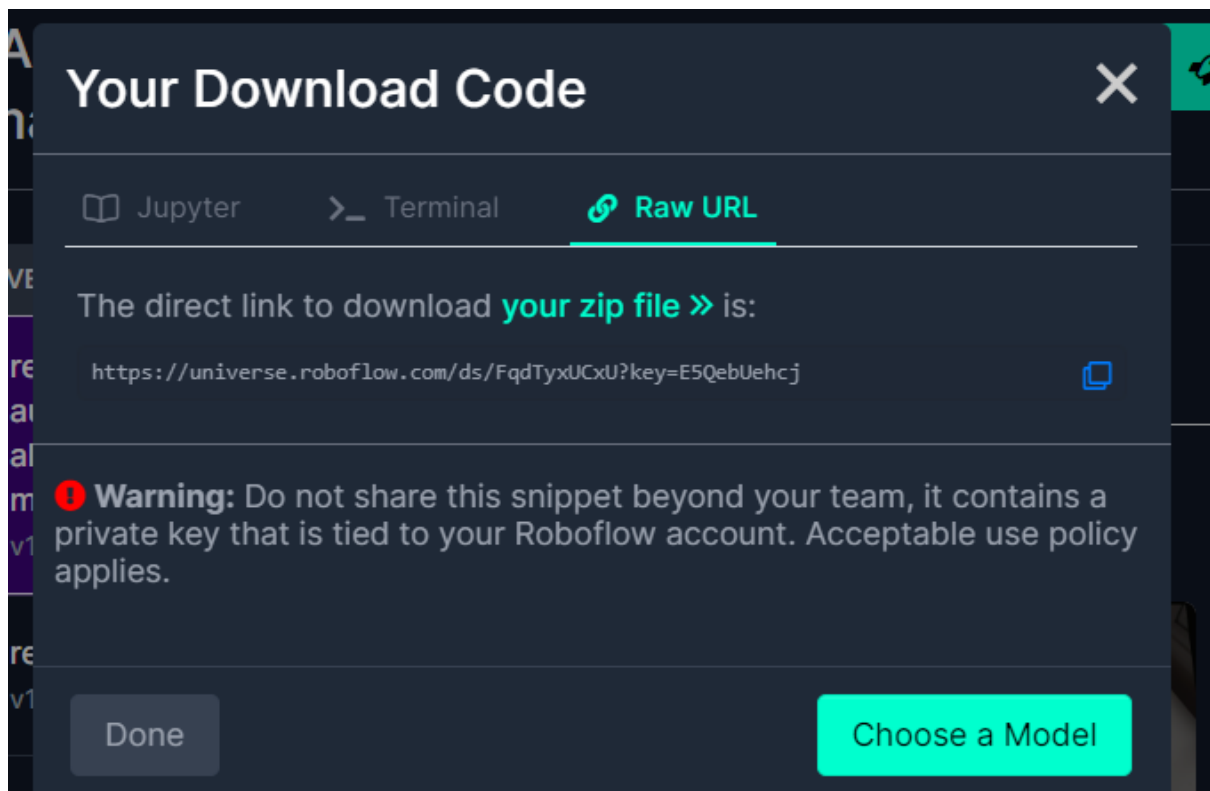
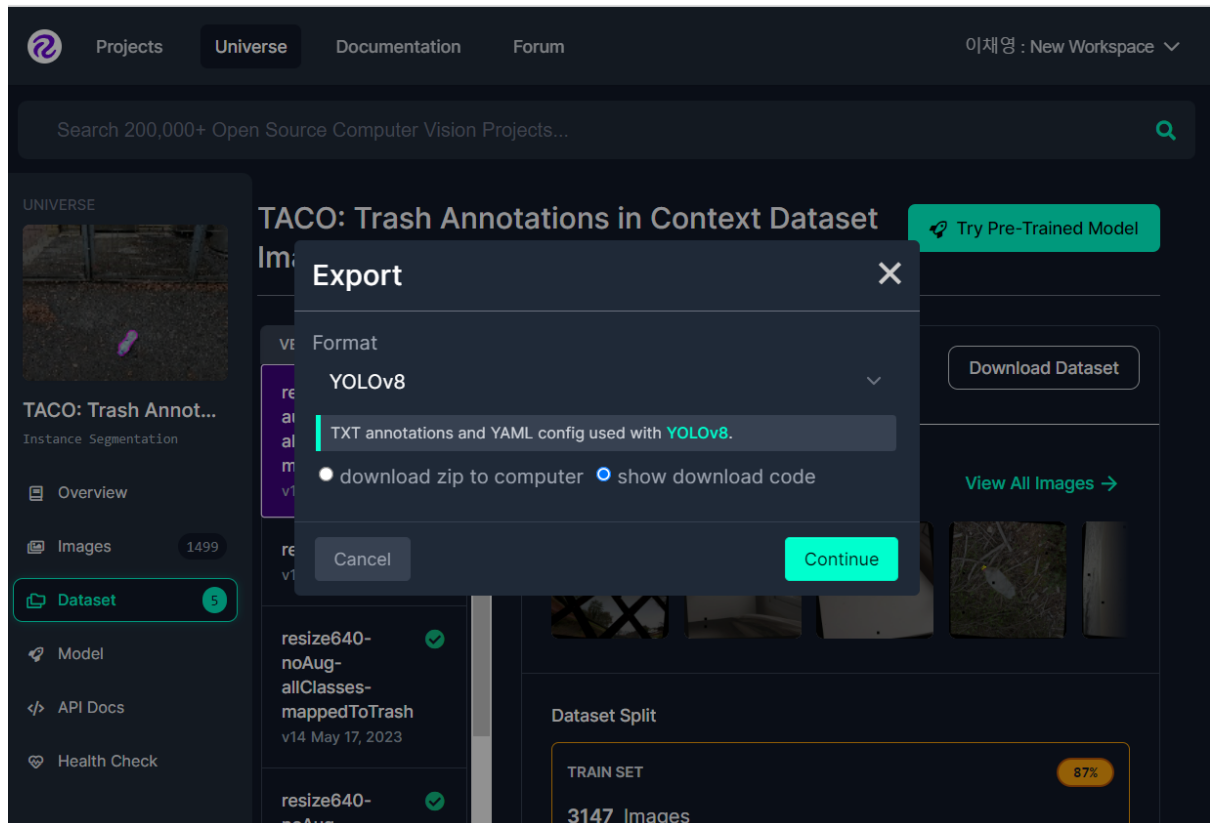
[6] Prediction

Prediction	
<pre>results = model.predict(source='/content/test/')</pre>	<pre># predict on test images</pre>

실습

[1] Custom data 구축

- Roboflow 사이트에서 TACO Dataset(쓰레기 인식하는 dataset) 을 다운받아 사용하기
- <https://universe.roboflow.com>



⇒ Raw URL 클릭시 나오는 url 링크로 이동하여 추후 Colab 에서 wget 명령어를 통해 url 대응되는 곳에 저장되어 있는 데이터를 가져온다.

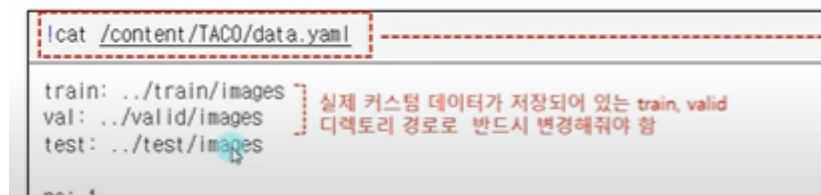
[2] 데이터를 Colab 으로 다운로드

<https://universe.roboflow.com/ds/FqdTyxUCxU?key=E5QebUehcj>

```
wget -O TACO.zip https://universe.roboflow.com/ds/FqdTyxUCxU?key=E5QebUehcj
```

➤ Loading Data (데이터를 Colab 으로 다운로드)

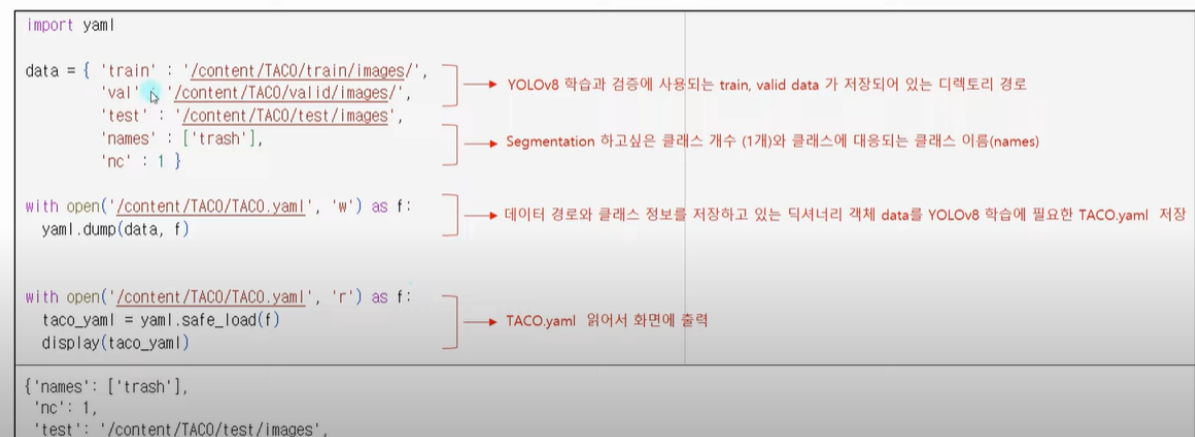
- ✓ wget 으로 다운받은 압축 파일을 해제하면 다음과 같은 이미지(images)와 정답(labels) 데이터가 있는 test, train, valid 디렉토리 경로, 클래스 개수와 이름이 저장되어 있는 data.yaml 파일이 있는데, yaml 파일에서 커스텀 데이터로 학습 (파인튜닝) 할 때 필요한 train, valid 데이터가 저장되어 있는 train, val 디렉토리 경로는 반드시 본인의 환경에 맞게 변경해줘야 함



다운받은 yaml 파일에서 커스텀 데이터로 학습할 때 필요한 train, valid 데이터가 저장되어 있는 디렉토리 경로를 로컬환경에 맞게 변경해줌.

[3] 커스텀 데이터에 맞는 YAML 파일 만들기

```
pip install PyYAML
```



- 실제 경로로 바꿔주기
- 클래스 이름 고치고
- 이름 바꿔주기

변경 후 TACO.yaml 로 저장

잘 저장되었는지는 safe_load 로 확인 가능.

[4] YOLOv8 라이브러리 설치

`pip install ultralytics` 설치

`checks()` 로 버전확인

```

> Install YOLOv8

!pip install ultralytics

import ultralytics

ultralytics.checks()

Ultralytics YOLOv8.0.151 Python-3.10.12 torch-2.0.1+cu118 CUDA:0 (Tesla T4, 15102MiB)
Setup complete (2 CPUs, 12.7 GB RAM, 25.1/78.2 GB disk)

> Load a pre-trained model

# Load YOLOv8n-seg
from ultralytics import YOLO

model = YOLO('yolov8n-seg.pt') # load a pretrained YOLOv8n segmentation model

Downloading https://github.com/ultralytics/assets/releases/download/v0.0.0/yolov8n-seg.pt to 'yolov8n-seg.pt'...
100%|██████████| 6.73M/6.73M [00:01<00:00, 7.01MB/s]

print(type(model.names), len(model.names))
print(model.names)

<class 'dict'> 80
{0: 'person', 1: 'bicycle', 2: 'car', 3: 'motorcycle', 4: 'airplane', 5: 'bus', 6: 'train', 7: 'truck', 8: 'boat', 9: 'traffic light', 10: 'fire hydrant', 11: 'stop sign', 12: 'parking meter', 13: 'bench', 14: 'parking meter', 15: 'parking meter', 16: 'parking meter', 17: 'parking meter', 18: 'parking meter', 19: 'parking meter', 20: 'parking meter', 21: 'parking meter', 22: 'parking meter', 23: 'parking meter', 24: 'parking meter', 25: 'parking meter', 26: 'parking meter', 27: 'parking meter', 28: 'parking meter', 29: 'parking meter', 30: 'parking meter', 31: 'parking meter', 32: 'parking meter', 33: 'parking meter', 34: 'parking meter', 35: 'parking meter', 36: 'parking meter', 37: 'parking meter', 38: 'parking meter', 39: 'parking meter', 40: 'parking meter', 41: 'parking meter', 42: 'parking meter', 43: 'parking meter', 44: 'parking meter', 45: 'parking meter', 46: 'parking meter', 47: 'parking meter', 48: 'parking meter', 49: 'parking meter', 50: 'parking meter', 51: 'parking meter', 52: 'parking meter', 53: 'parking meter', 54: 'parking meter', 55: 'parking meter', 56: 'parking meter', 57: 'parking meter', 58: 'parking meter', 59: 'parking meter', 60: 'parking meter', 61: 'parking meter', 62: 'parking meter', 63: 'parking meter', 64: 'parking meter', 65: 'parking meter', 66: 'parking meter', 67: 'parking meter', 68: 'parking meter', 69: 'parking meter', 70: 'parking meter', 71: 'parking meter', 72: 'parking meter', 73: 'parking meter', 74: 'parking meter', 75: 'parking meter', 76: 'parking meter', 77: 'parking meter', 78: 'parking meter', 79: 'parking meter', 80: 'parking meter'}

```

사전학습 된 yolov8n-seg 모델 로드

[5] YOLOv8 커스텀 데이터 학습하기

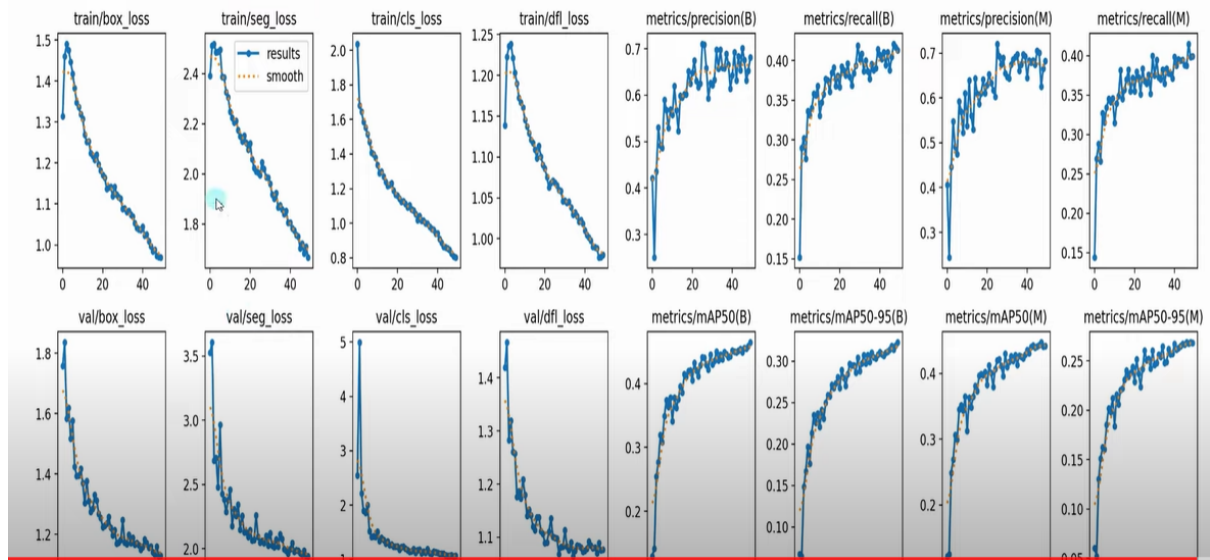
`model.train(data= '~~~')` yaml 파일을 파라미터로 지정해주고

epochs → 반복횟수

커스텀 데이터로 학습된 모델이 `best.pst`, `last.pt` 로 저장된다

사전학습된 데이터로 학습되었기 때문에, 클래스 개수가 80개가 아닌 1개로 변경되어있음.

➤ YOLOv8 커스텀 데이터 학습하기



손실값이 지속적으로 감소하고 있으며, 정확도와 리콜, MAP 는 지속적으로 증가.

[6] 학습된 데이터로 테스트 이미지 예측

```
result = model.predict(source = '파라미터', save = True)
```

predict 함수로 실행

<https://github.com/neowizard2018/neowizard/tree/master/DeepLearningProject/>