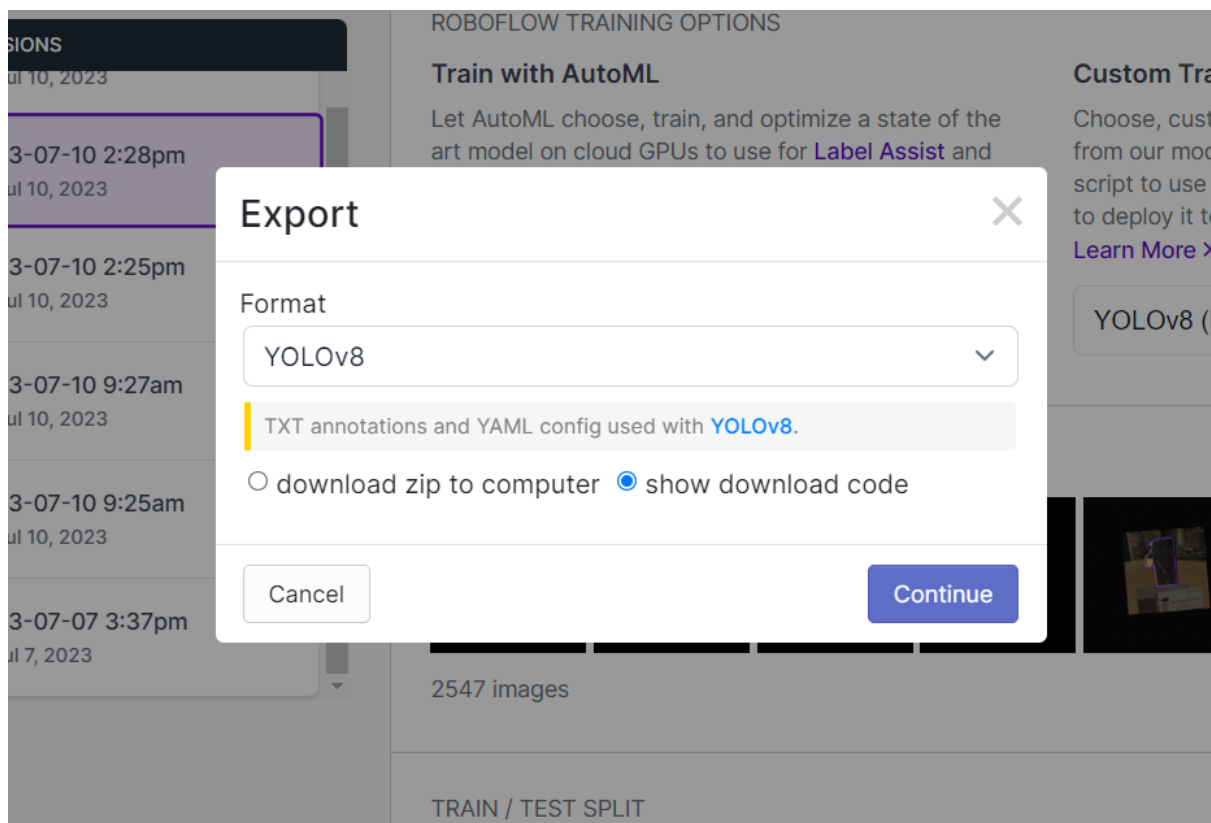


2023년 07월 10일

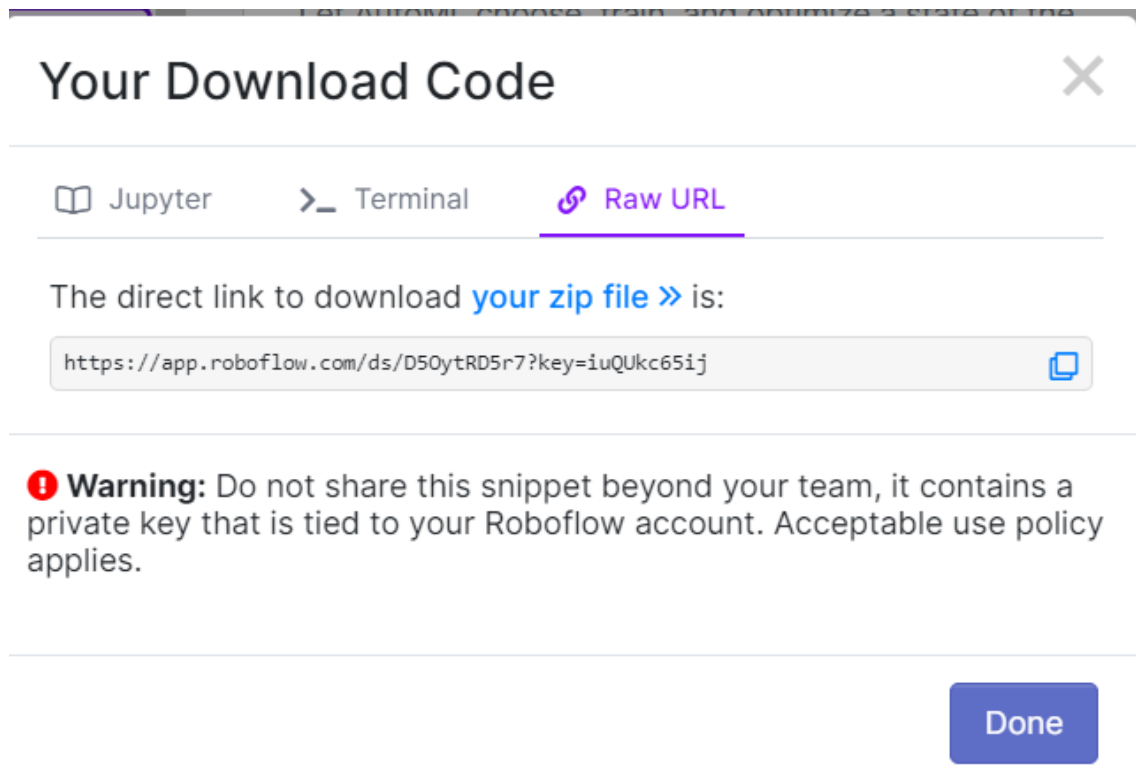
Yolov8 학습 과정

1. Custom data 준비
2. Roboflow에서 제공하는 YAML파일 만들기
3. Yolov8 실행에 필요한 라이브러리 설치
4. 학습
5. 예측

Roboflow 사이트에서 Custom data를 준비한다. zip파일과 다운로드 코드로 다운받을 수 있다.



Google Colab Code 에서 작업할 예정이므로 Raw URL을 복사해준다.



Google Colab

아래의 코드를 실행하기전 구글 드라이브 연동을 먼저 한다.

```
!wget -O FileName.zip URL
```

복사한 코드를 URL대신 붙여 넣는다.

```
import zipfile

with zipfile.ZipFile('/content/FileName.zip') as target_file:

    target_file.extractall('/content/FileName')
```

다운 받은 zip파일 압축을 해제한다.

***다운로드 받은 파일의 경로가 잘 되어있는지 확인해 보는게 좋다.**

```
!pip install PyYAML
```

구성요소 정보가 담겨있는 YAML파일을 사용하기위해 install 해준다.

```
# yaml 파일을 학습이 가능하도록 경로 설정.  
# key-value 데이터인 dict 데이터타입으로 data['train'], data['val'], data['nc'], data['names'] 에 넣어주는데,  
# 가장 중요한 부분은 데이터 경로 설정임.
```

```
import yaml  
  
data = { 'train' : '/content/Cup.v4i.yolov8/train/images/',  
        'val' : '/content/Cup.v4i.yolov8/valid/images/',  
        'test' : '/content/Cup.v4i.yolov8/test/images/',  
        'names' : ['Cup(glass)', 'Cup(plastic)', 'Cup(paper)', 'Tumbler'],  
        'nc' : 4 }  
  
with open('/content/Cup.v4i.yolov8/data.yaml', 'w') as f:  
    yaml.dump(data, f)  
  
with open('/content/Cup.v4i.yolov8/data.yaml', 'r') as f:  
    cup_yaml = yaml.safe_load(f)  
    display(cup_yaml)
```

```
!pip install ultralytics  
import ultralytics  
ultralytics.checks()  
print(type(model.names), len(model.names))  
print(model.names)
```

코드 실행 결과

```
<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:
```

기본으로 80개의 라벨링 데이터가 있다. 우리는 새로 학습을시켜 라벨링 데이터를 커스텀할 예정이다.

모델학습

```
model.train(data='/content/Cup.v4i.yolov8/data.yaml', epochs=100, patience=30, batch=32, imgsz=416)
```

주로 버려진 쓰레기 위주로 이미지 학습을 시키니 테스트 이미지는 인식이 잘되나 실제 이미지는 인식이 잘 안된다.

카페내 이미지 위주로 학습시킬 예정입니다.