#Colab에서 train

# Roboflow에서 제공한 데이터셋을 다운로드 받음.

!wget -O violence\_train.zip https://app.roboflow.com/ds/Fa4VJ6GYHU?key=1uaF3xbDKQ

pwd

# Colab Notebooks 디렉토리로 이동합니다.

cd /content/drive/MyDrive/Colab Notebooks

pwd

# zipfile 모듈을 사용하여 압축 파일을 해제합니다.

import zipfile

with zipfile.ZipFile('/content/drive/MyDrive/Colab Notebooks/violence\_train.zip') as target\_file:

target\_file.extractall('/content/violence\_train/')

# data.yaml 파일의 내용을 확인합니다.

!cat /content/violence\_train/data.yaml

# PyYAML 라이브러리를 설치합니다.

!pip install PyYAML

import yaml

# data.yaml 파일을 새로 작성합니다.

data = {'train' : '/content/violence\_train/train/images',

'val' : '/content/violence\_train/valid/images',

'names' : ['v1'],

'nc' : 1 }

with open('/content/violence\_train/data.yaml','w') as f:

yaml.dump(data,f)

# 새로 작성된 data.yaml 파일을 확인합니다.

with open('/content/violence\_train/data.yaml', 'r') as f:

violence\_train = yaml.safe\_load(f)

display(violence\_train)

# ultralytics 패키지를 설치합니다.

!pip install ultralytics

import ultralytics

ultralytics.checks()

# ultralytics의 YOLO 모델을 불러옵니다.

from ultralytics import YOLO

model = YOLO('yolov8s.pt')

# YOLO 모델의 클래스 이름을 확인합니다.

print(model.names)

# YOLO 모델을 훈련시킵니다.

model.train(data='/content/violence\_train/data.yaml', epochs=10, batch=32, imgsz=640, patience=5)

# 훈련된 모델의 클래스 이름을 확인합니다.

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

print(model.names)

<class 'dict'> 1

{0: 'v1'}

#학습이 완료된 후 아래 경로의 best.pt 파일 다운합니다.



#Anaconda 다운로드 후 설치를 한뒤 가상환경을 다음 명령어를 사용하여 생성합니다.

conda create -n test1 python=3.8

#가상환경 진입

conda activate test1

# YOLO 모델을 사용하여 객체를 감지합니다.

yolo predict model=best.pt source=0 show=True

\* 실행이 안된다면

- pip install ultralytics 설치

- pip install yolo 설치

\* format을 torchscript로 변형하기

from ultralytics import YOLO

# Load a model

model = YOLO("yolov8s.yaml") # build a new model from scratch

model = YOLO("yolov8s.pt") # load a pretrained model (recommended for training)

# Use the model

model.train(data="/content/violence\_train/data.yaml", epochs=3) # train the model

metrics = model.val() # evaluate model performance on the validation set

results = model(data="/content/violence\_train/data.yaml") # predict on an image

path = model.export(format="torchscript") # export the model to ONNX format