UltraLytics YOLO v8 일단 해보기

라이브러리 설치

```
!pip install ultralytics
```

```
Successfully installed nvidia-cublas-cu12-12.1.3.1 nvidia-cuda-cupti-cu12-12.1.105 nvidia-cuda-nvrtc-cu12-12.1.105 nvidia-cuda-runtime-cu12-12.1.105
```

라이브러리 불러오기

YOLO v8 설정

```
from ultralytics import settings
```

```
settings
```

```
{'settings_version': '0.0.4',
 'datasets_dir': '/content/datasets',
 'weights_dir': 'weights',
 'runs_dir': 'runs',
 'uuid': '569f3ba64b326db489132663f79cd37279811de477381b83ac131e6cdd129cbb',
 'sync': True,
 'api_key': '',
 'openai_api_key': '',
 'clearml': True,
 'comet': True,
 'dvc': True,
 'hub': True,
 'mlflow': True,
 'neptune': True,
 'raytune': True,
 'tensorboard': True,
 'wandb': True}
```

YOLO v8 모델

```
from ultralytics import YOLO
```

```
YOLO
```

```
ultralytics.models.yolo.model.YOLO
def __call__(source: Union[str, Path, int, list, tuple, np.ndarray, torch.Tensor]=None,
stream: bool=False, **kwargs) -> list

/usr/local/lib/python3.10/dist-packages/ultralytics/models/yolo/model.py
YOLO (You Only Look Once) object detection model.
```

모델링

모델 선언

- 모델의 구조와 해당 구조에 맞게 사전 학습된 가중치를 불러온다.
- Parameters
    1. model : 모델 구조 또는 모델 구조 + 가중치 설정. task와 맞는 모델을 선택해야 한다.
    2. task : detect, segment, classify, pose 중 택일

```
model = YOLO(model='yolov8n.pt', task='detect')
```

```
Downloading https://github.com/ultralytics/assets/releases/download/v8.1.0/yolov8n.pt to 'yolov8n.pt'...
100%|██████████| 6.23M/6.23M [00:00<00:00, 115MB/s]
```

```
# model = YOLO()
```

모델 학습

- Parameters
  1. data : 학습시킬 데이터셋의 경로. default 'coco128.yaml'
  2. epochs : 학습 데이터 전체를 총 몇 번씩 학습시킬 것인지 설정. default 100
  3. patience : 학습 과정에서 성능 개선이 발생하지 않을 때 몇 epoch 더 지켜볼 것인지 설정. default 100
  4. batch : 미니 배치의 사이즈 설정. default 16. -1일 경우 자동 설정.
  5. imgsz : 입력 이미지의 크기. default 640
  6. save : 학습 과정을 저장할 것인지 설정. default True
  7. project : 학습 과정이 저장되는 폴더의 이름.
  8. name : project 내부에 생성되는 폴더의 이름.
  9. exist_ok : 동일한 이름의 폴더가 있을 때 덮어씌울 것인지 설정. default False
  10. pretrained : 사전 학습된 모델을 사용할 것인지 설정. default True
  11. optimizer : 경사 하강법의 세부 방법 설정. default 'auto'
  12. verbose : 학습 과정을 상세하게 출력할 것인지 설정. default False
  13. seed : 재현성을 위한 난수 설정
  14. resume : 마지막 학습부터 다시 학습할 것인지 설정. default False
  15. freeze : 첫 레이어부터 몇 레이어까지 기존 가중치를 유지할 것인지 설정. default None

```
model.train(data='coco128.yaml',
            epochs=10,
            patience=5,
            save=True,
            # project='trained',
            # name='trained_model',
            exist_ok=False,
            pretrained=False,
            optimizer='auto',
            verbose=False,
            seed=2024,
            resume=False,
            freeze=None
            )
```

```
0.58659,     0.58759,     0.58859,     0.58959,     0.59059,     0.59159,     0.59259,     0.59359,     0.59459,      0.5956,      0.5966,
0.5976,      0.5986,      0.5996,
              0.6006,      0.6016,      0.6026,      0.6036,      0.6046,     0.60561,     0.60661,     0.60761,     0.60861,     0.60961,
0.61061,     0.61161,     0.61261,     0.61361,     0.61461,     0.61562,     0.61662,     0.61762,     0.61862,     0.61962,     0.62062,
0.62162,     0.62262,     0.62362,
             0.62462,     0.62563,     0.62663,     0.62763,     0.62863,     0.62963,     0.63063,     0.63163,     0.63263,     0.63363,
0.63463,     0.63564,     0.63664,     0.63764,     0.63864,     0.63964,     0.64064,     0.64164,     0.64264,     0.64364,     0.64464,
0.64565,     0.64665,     0.64765,
             0.64865,     0.64965,     0.65065,     0.65165,     0.65265,     0.65365,     0.65465,     0.65566,     0.65666,     0.65766,
0.65866,     0.65966,     0.66066,     0.66166,     0.66266,     0.66366,     0.66466,     0.66567,     0.66667,     0.66767,     0.66867,
0.66967,     0.67067,     0.67167,
             0.67267,     0.67367,     0.67467,     0.67568,     0.67668,     0.67768,     0.67868,     0.67968,     0.68068,     0.68168,
0.68268,     0.68368,     0.68468,     0.68569,     0.68669,     0.68769,     0.68869,     0.68969,     0.69069,     0.69169,     0.69269,
0.69369,     0.69469,      0.6957,
              0.6967,      0.6977,      0.6987,      0.6997,      0.7007,      0.7017,      0.7027,      0.7037,      0.7047,     0.70571,
0.70671,     0.70771,     0.70871,     0.70971,     0.71071,     0.71171,     0.71271,     0.71371,     0.71471,     0.71572,     0.71672,
0.71772,     0.71872,     0.71972,
             0.72072,     0.72172,     0.72272,     0.72372,     0.72472,     0.72573,     0.72673,     0.72773,     0.72873,     0.72973,
0.73073,     0.73173,     0.73273,     0.73373,     0.73473,     0.73574,     0.73674,     0.73774,     0.73874,     0.73974,     0.74074,
0.74174,     0.74274,     0.74374,
             0.74474,     0.74575,     0.74675,     0.74775,     0.74875,     0.74975,     0.75075,     0.75175,     0.75275,     0.75375,
0.75475,     0.75576,     0.75676,     0.75776,     0.75876,     0.75976,     0.76076,     0.76176,     0.76276,     0.76376,     0.76476,
0.76577,     0.76677,     0.76777,
             0.76877,     0.76977,     0.77077,     0.77177,     0.77277,     0.77377,     0.77477,     0.77578,     0.77678,     0.77778,
```

모델 검증

```
# model.val()
```

예측값 생성

- Parameters
  1. source : 예측 대상 이미지/동영상의 경로
  2. conf : confidence score threshold. default 0.25
  3. iou : NMS에 적용되는 IoU threshold. default 0.7. threshold를 넘기면 같은 object를 가리키는 거라고 판단.
  4. save : 예측된 이미지/동영상을 저장할 것인지 설정. default False
  5. save_txt : Annotation 정보도 함께 저장할 것인지 설정. default False
  6. save_conf : Annotation 정보 맨 끝에 Confidence Score도 추가할 것인지 설정. default False
  7. line_width : 그려지는 박스의 두께 설정. default None

```
results = model.predict(source='https://images.pexels.com/photos/139303/pexels-photo-139303.jpeg',
                        #conf=0.5,
                        iou=0.5,
                        save=True, save_txt=True, line_width=2)
```

```
Found https://images.pexels.com/photos/139303/pexels-photo-139303.jpeg locally at pexels-photo-139303.jpeg
Results saved to runs/detect/train4
1 label saved to runs/detect/train4/labels
```

코딩을 시작하거나 AI로 코드를 생성하세요.

코딩을 시작하거나 AI로 코드를 생성하세요.

코딩을 시작하거나 AI로 코드를 생성하세요.