

data preprocess documentation

☰ 태그

사전조사

data preprocess documentation

1. sample_data_into_yolo_structure.py

- 원천 데이터에서 —selection-ratio = 0.5로 설정, selected_images_0.5 샘플데이터를 뽑아낸다

→데이터가 너무 많기에 GPU가 감당할 수 있는 선으로 적게 뽑아보기 위함

실제 실행된 명령어

→ `python sample_data_into_yolo_structure.py --selection-ratio 0.5`

—selection-ratio type=float 어떤어떤 것을 위한 것, ~~경로. ~~비율

2. create_feature_dataset.py

- selected_images_0.25 안에 있는 파일명을 한글에서 영어로 변환시킨다

→파일명이 한글로 되어있을 경우 오류가 발생할 수 있기 때문에 변환

실제 실행된 명령어

→ `python create_feature_dataset.py --src-dir-path selected_images_0.5`

3. create_yolo_detection_dataset.py

- 2 과정에서 생성된 데이터셋을 yolo detection dataset으로 변환시킨다

→ yolo detection을 training 시킬 때 yolo가 요구하는 특정 파일 구조가 있기 때문에 변환

실제 실행된 명령어

→ `python create_yolo_detection_dataset.py`

4. split_train_val.py

a. yolo 형식에 맞게 바꾼 데이터셋을 split 해준다

→ train : val = 8 : 2 비율로 가져오고 test까지 가져와야 최종에서 test로 평가가 가능하기 때문이다

실제 실행된 명령어

→ `python split_train_val.py --annot-format yolo --task detection --train-ratio 0.8`

5. run.py

a. 앞서 뽑은 데이터셋으로 yolo detection 학습을 시킨다

→ 이때 pretrained로 yolov8n-face.pt를 사용하는데, 다른 데이터셋을 미리 학습시킨 가중치를 이어가면서 수월하게 학습을 시키기 위함이다

실제 실행된 명령어

→ `python run.py -run train -type yolo -cfg "/home/KDT-admin/work/github/moonsun/est_wassup/config/project_4_yolo.yaml"`

<yolo detect crop을 위한 준비>

1. 2번에서 뽑아낸 데이터셋을 새롭게 split 해준다

→ 또 나눠주는 이유는 run.py에서 쓰이는 학습 구조와 detect.py에서 쓰이는 학습 구조가 다르기 때문이다

2. detect.py

a. 앞서 split한 데이터셋으로 detect.py를 돌린다

→ run.py 학습을 통해 나온 weights를 사용해준다

3. 전과정을 통해 crop한 데이터셋이 완성된다

4. 직접 눈으로 보면서 잘못 인식된 데이터 걸러내는 작업