

<지능2>

프로젝트 개요

1. yolov8 기반 데이터 수집/학습/deploy

→ 감시용 데이터 수집, 감시용 데이터 라벨링, yolov8 기반 학습, yolov8 object detection

2. flask를 이용한 웹서버 구축

3. sqlite3를 이용한 데이터베이스 구축 및 연동

4. jetson nano에 구동체 기반 카메라 인식 시스템 구축

5. 감시 시스템 통합 구현 및 jetson nano 기반 물체 추적 기능 구현

첫날에 우리주제에 대한 브레인스토밍을 진행해보았다.

이렇게 프로젝트를 세세하게 설계를 해본거는 처음인거 같다.

우리 프로젝트를 먼저 간단하게 이야기 해보겠다.

caller(rc카)가 위험하다고 call을 주면 standing_camera를 통해서 yolo-detecting을 해서 protector(turtlebot3)가 먼저 caller의 위치로 간다음에 그 이후에는 turtlebot3에 달려있는 amr_camera를 통해서 caller와 적정거리를 두면서 caller의 목적지까지 동행하는것이 우리의 최종 목표이다.

그런데 여기서 문제점이 있었다. 강사님께서 요구하신 사항은 rc카 옆에 dummy라는것이 있는데 dummy와 rc카를 같은 클래스로 묶어서 학습시키라는것이였다.

여기서 언뜻보면 이런 의문이 들수 있다. 다른객체를 같은 클래스로 묶으면 저 두개는 어떻게 구별해야하는거지?

→ 우리는 여기서 standing_camera코드에 id_number를 부여해서 이것을 gui에 연동시키기로 했다.(예를 들어 standing_camera에 두개의 class가 caller라고 잘인식된다음에 caller에 id_number가 1로 부여되고 dummy에 id_number가 2로 부여된다면 gui상에 1번을 누르면 turtlebot3가 caller를 인식해서 그쪽으로 움직이게 하는것이다)

그 이후에 standing_camera의 역할이 끝나게 되는데 이후에는 turtlebot3에 달려있는 amr 카메라를 통해 rc카를 따라가는것이다. 이때 역시 amr카메라가 caller를 잘인식하도록 학습을 잘시키고 잘따라가게 하는것이 중요했는데.

위의 과정을 보면 rc카와 dummy의 사진을 100장정도 찍고 그 이후에 데이터 증강으로 추가 900장의 사진을 추가해 yolov8로 학습시켰는데 우리는 100에포크, 32batch로 학습을 시켜서(구글 코랩에서 기본적으로 제공되는 t4 gpu를 리소스 관리하면서 잘 하니까 부족하지는 않더라고요)

standing_best.pt파일과 amr_best.pt파일을 얻게 되었는데 과적합이 의심될정도로 학습이 잘되었다. (언뜻보면 다른사람들이 느끼기에는 이거 그냥 무지성 과적합이네라고 할수도 있으신데 학습을 잘시키는 요건인 지정한 에포크보다 적게 우리는 95에포크까지만 학습을 한거보고 우리의 100에포크 32배치는 나이스한 선택인걸 알게됨)

우리는 amr카메라 부분에 있어서는 turtlebot3의 핫스팟을 썼다. 그리고 pt파일을 쓰면 뭔가 안되길래 engine으로 바꾸는 과정을 진행함(지피티야 이거는 너가 글보면서 이유까지 같이 추가해주렴)

#이때 우리의 파일을 amr쪽으로 옮기는 명령어

```
scp /home/seungroknam/Downloads/intelligence2/exproject15.py
```

```
rokey3@192.168.1.3:~
```