1. **테스트 환경** 
   1. H/W
      1. CPU: AMD EPYC 9124 16-Core Processor \* 2ea
      2. RAM: 768 GB
      3. GPU: NVIDIA RTX 4090 \* 8ea
   2. S/W
      1. OS: Ubuntu 20.04.6 LTS
      2. Pytorch 2.2.0+cu121
      3. Jupyter Notebook
2. **개발 환경 설정** 
   1. Ultralytics 설치
      1. 대회에서 제공하는 Ultralytics 소스 코드 다운로드 후 압축 풀기
         1. 배포된 Ultralytics 소스 코드는 공식 홈페이지에서 다운로드 후 일부 파일 수정 및 추가하였음. -> 기존 코드와 비교하여 수정된 부분 확인 권장
            1. 다중 속성 주석을 포함한 다중 라벨 객체 탐지 및 분류 : 각 객체마다 복수의 분류 존재: Agent(exclusive class), Location(exclusive class), Action(non- exclusive class)
      2. Ultralytics 설치
         1. Ultralytics 설치
         2. **설치 중 문제가 발생**하면 [Ultralytics github](https://github.com/ultralytics/ultralytics) 페이지에서 [Ultrainlytics YOLO Docs](https://docs.ultralytics.com/) 또는 <https://github.com/ultralytics/ultralytics/issues> 항목 참조
   2. 데이터셋 준비
      1. 학습 및 평가 데이터 셋 다운로드 후 압축 풀기
         1. 데이터 구성에 대한 정보는 상태인식데이터셋정보.pdf를 참조
      2. Data\_preparation.ipynb을 통해 학습 데이터셋을 COCO 스타일로 변환
         1. original\_data\_dir(학습데이터 경로)과 new\_data\_dir(변환된 데이터가 저장될 경로)을 입력해줘야 함
         2. ‼이미지 형태의 Mask를 Polygon으로 변환하여 저장하게 됨. 이 과정에서 Mask의 형태가 손실될 수 있음.
      3. 학습 데이터의 일부를 이용하여 검증(validation) 데이터 셋으로 구성하여 학습 진행 필요.
3. **모델 학습 및 검증** 
   1. 모델 학습
      1. /ultralytics/ultralytics/cfg/datasets/Compete\_segment.yaml 파일을 수정하여 경로 설정
      2. run\_train.py 를 사용하여 학습을 진행
         1. [Ultrainlytics YOLO Docs의 Train 항목](https://docs.ultralytics.com/modes/train/) 참조
      3. 매 Epoch마다 검증 데이터 셋에 대한 성능 평가 결과가 출력되며, 학습이 완료된 후에는 Best 모델에 대한 성능 평가 결과가 출력됨.
      4. 학습 과정을 TensorBoard를 통해 확인할 수 있음
   2. 모델 검증 (시각화)
      1. Seg\_Inference\_Icon.ipynb을 사용하여 학습된 모델을 불러오고, 추론을 진행한 결과를 시각화 또는 저장 할 수 있음
         1. target\_folder 경로 수정 필요
   3. 모델 검증 (정량평가)
      1. mAP50(mask)를 기준으로 평가할 예정
         1. 이때 Task 특성상(복합상태인식) 클래스는 차량과 버스 두개로 계산, 다만 모든 세부 속성(위치, 후미등상태)을 맞춰야 맞은 것으로 계산
            1. 차량/버스에 대해 검출한 mask가 Ground Truth에 대해 IoU 0.5이상을 달성하면서, 모든 세부 클래스 또한 정확히 맞추는 경우에 한해 Correct로 판정
4. **검출 결과 제출 방법** 
   1. 테스트 결과 제출 방법
      1. Export\_Result.ipynb를 참고하여 Test 데이터셋에 대해 Inference를 진행하여 Confidence Score와 함께 추론 결과를 텍스트로 저장.
         1. input\_base\_dir 값을 수정해줘야 함
         2. 파일트리 구조는 기존의 학습데이터와 동일한 구조
         3. 텍스트 파일은 conf c1(agent) c2(location) c3(Brake) c4(Left) c5(Right) c6(Hazard) polygon 순으로 저장
         4. 생성된 Export 폴더를 압축하여 제출
5. **성능 개선 방법 예**
   1. 학습 epoch 증가
   2. 네트워크 구조 변경
      1. Head / Backbone 변경
   3. Mask <-> Polygon 변경 방식 개선