25
톰
2023-10-24

OpenCV/CNN 기반 자율주행 페인팅 로봇

김정호 201824451
정제영 201824581
최성렬 201724601

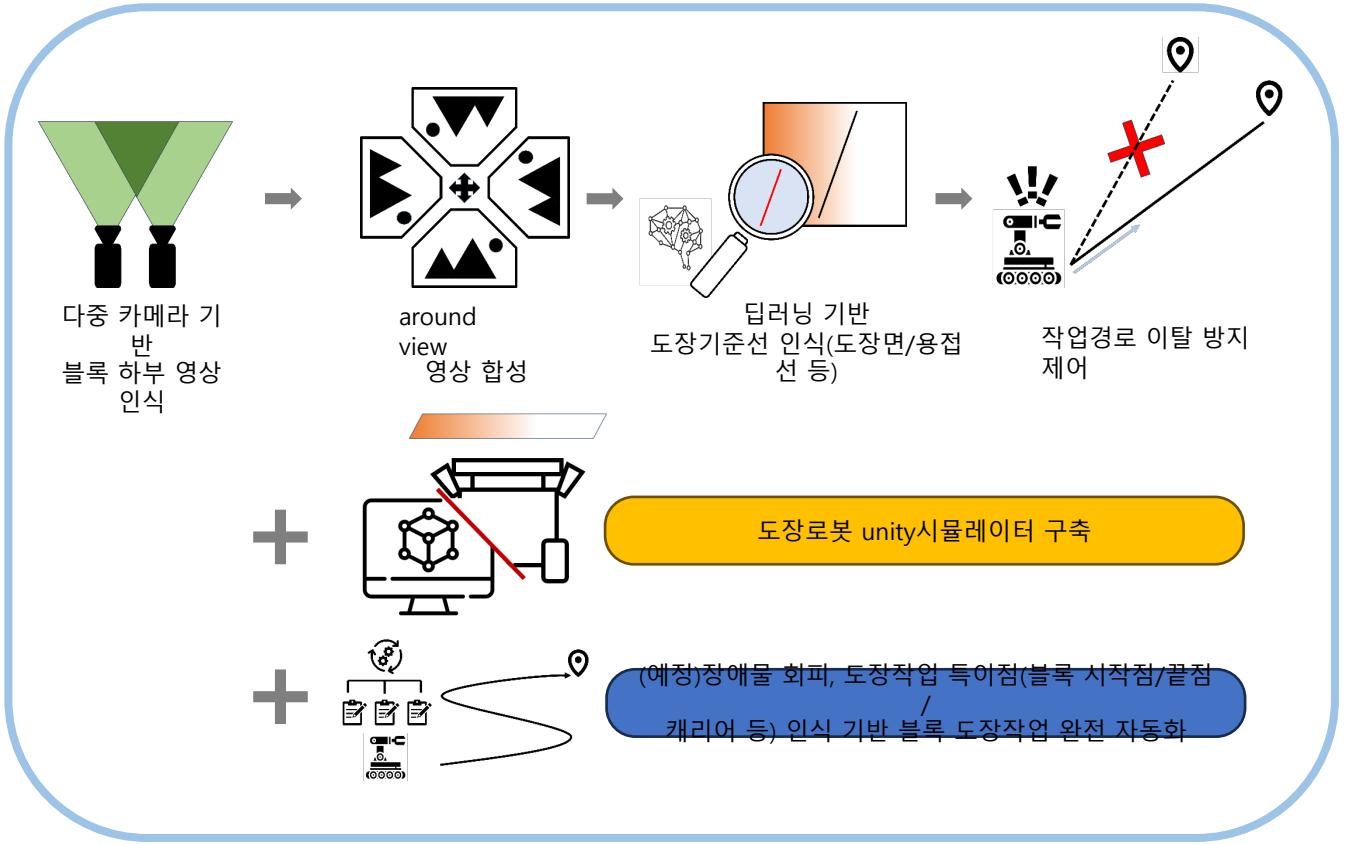
SAMSUNG 삼성중공업 부산대학교 PUSAN NATIONAL UNIVERSITY

시스템 개요

❖ 데이터 분석 및 전처리
- 다중 카메라 기반 블록 하부 영상 인식
- Around view 영상 합성

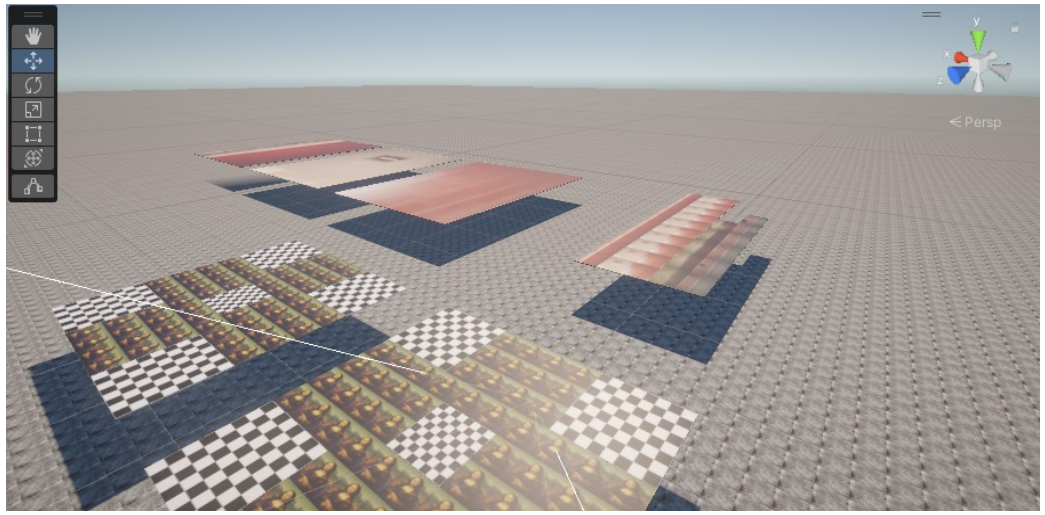
❖ 딥러닝 기반 경계면 인식(도장면/용접선 등)
- YOLOv8 모델을 이용해 도장면 object detection
- Airline line detection 및 curve fittin을 이용해 line detection

❖ 작업경로 생성 및 차량 이탈 방지
- 작업경로 이탈방지 제어
- 각도 및 도장면 object 정보를 이용한 제어 신호 생성
- Unity 기반 시뮬레이터 구성
- (예정) 장애물 회피, 도장작업 특이점(블록 시작점/끝점/캐리어 등) 인식 기반 블록 도장작업 완전 자동화

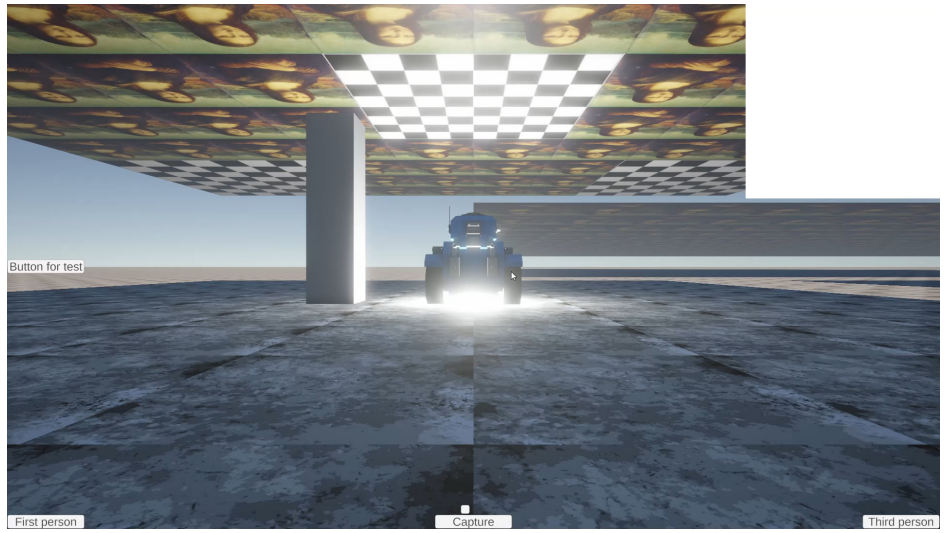


블록 하부 환경 테스트 환경 구축

- 실제 블록 하부 환경의 접근성이 상당히 낮고 테스트용 로봇 또한 부재
- Unity 기반 테스트 환경 구축을 통해 한계점 극복



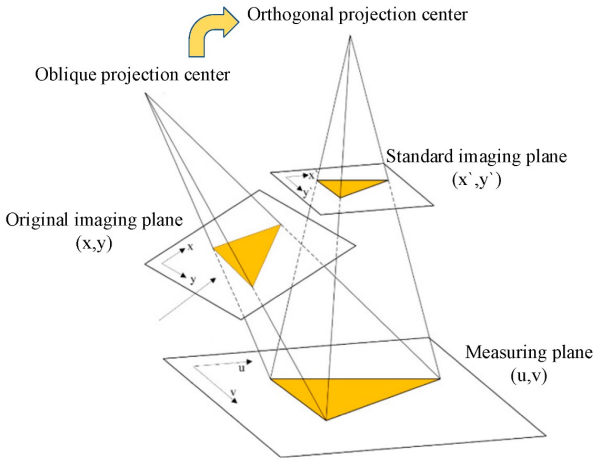
Unity 가상 공간



Unity 환경 스트리밍 화면

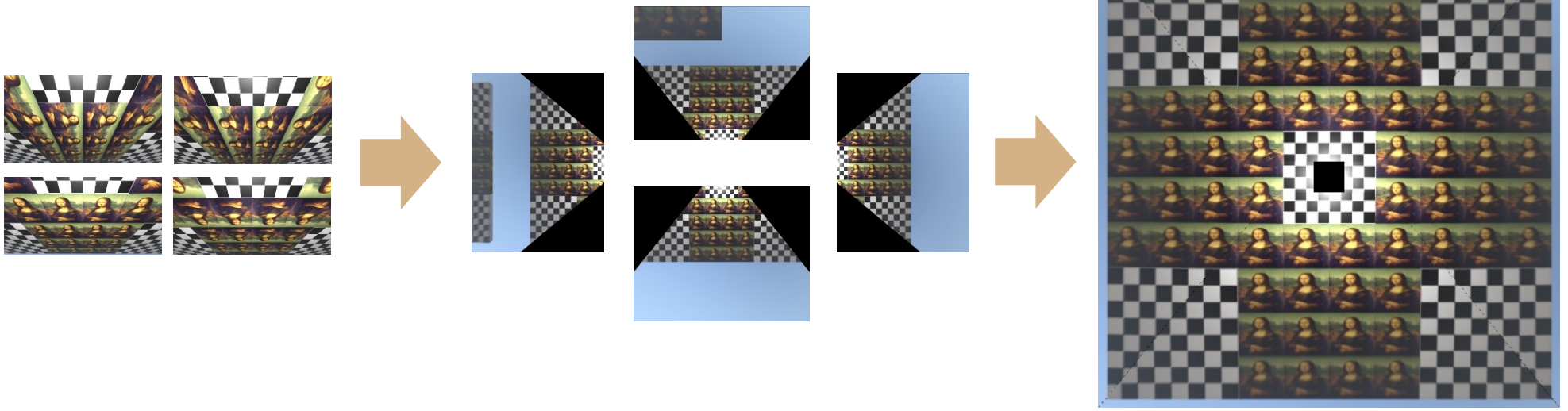
데이터 수집 : Around view 구현

❖ 도장면에는 다양한 object가 존재



❖ 도장면 정보의 광범위한 수집을 위해 4개의 카메라 활용

- 카메라 하나를 사용하여 도장면의 정보를 수집하기에는 도장면과 로봇의 간격이 너무 가까움
- 광각/어안렌즈 카메라 하나만 사용 시 왜곡 발생
- 4개의 카메라를 활용해 상하좌우의 이미지 촬영 후 perspective transform matrix 를 정립 후 calibration 및 stitching 과정을 거쳐 orthogonal 영상 확보



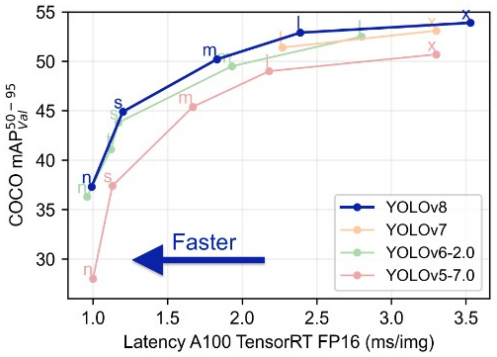
❖ YOLOv8의 특징 및 사용 이유

- 다중 클래스 탐지
- 기존의 YOLO모델 보다 더 높은 정확도 제공
- 도장면 영상에서 얻어진 line의 종류를 파악하기 위해 사용
- 얻어진 line을 Categorization하여 경로 생성 시 용도에 맞게 활용

Object Detection Performance Comparison (YOLOv8 vs YOLOv5)

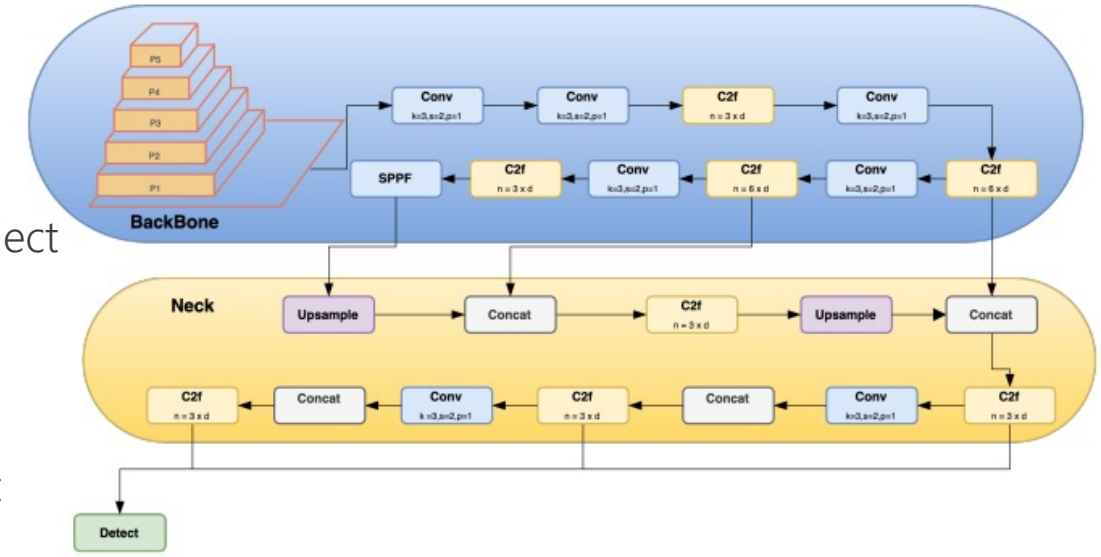
Model Size	YOLOv5	YOLOv8	Difference
Nano	28	37.3	+33.21%
Small	37.4	44.9	+20.05%
Medium	45.4	50.2	+10.57%
Large	49	52.9	+7.96%
Xtra Large	50.7	53.9	+6.31%

*Image Size = 640



❖ Class 분류

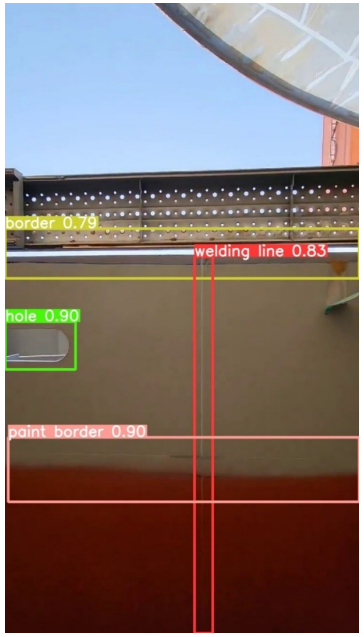
- welding line : 작업 시 기준선의 역할을 할 object
- paint border : 작업 시 도장 경계 및 기준선의 역할을 할 object
- obstacle, hole : 도장 불가 지점
- plate border : 진입점의 기준이 될 object
- carrier border : 작업 시/종 점 및 기준선의 역할을 할 object

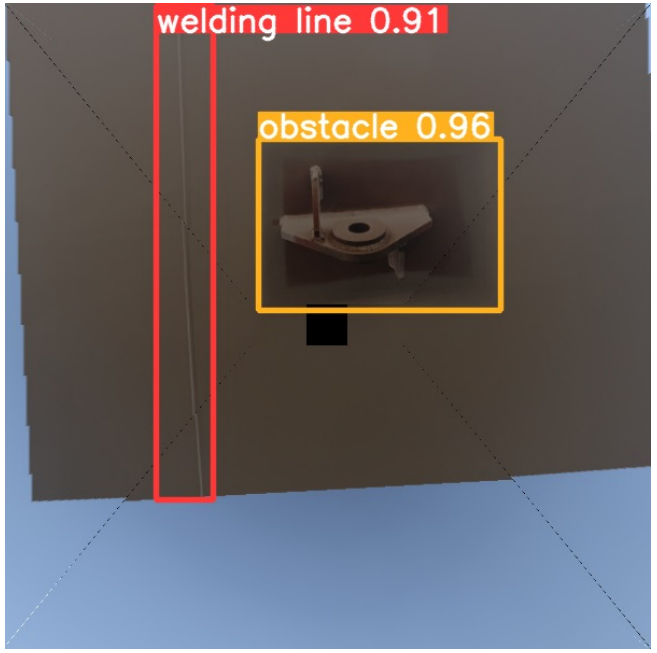


딥러닝 기반 도장면 인식: 학습 결과

❖ 학습결과

- 실제 하부 도장 환경 및 aroundview로 구성된 Unity 시뮬레이터 상의 이미지 또한 높은 인식율을 보임





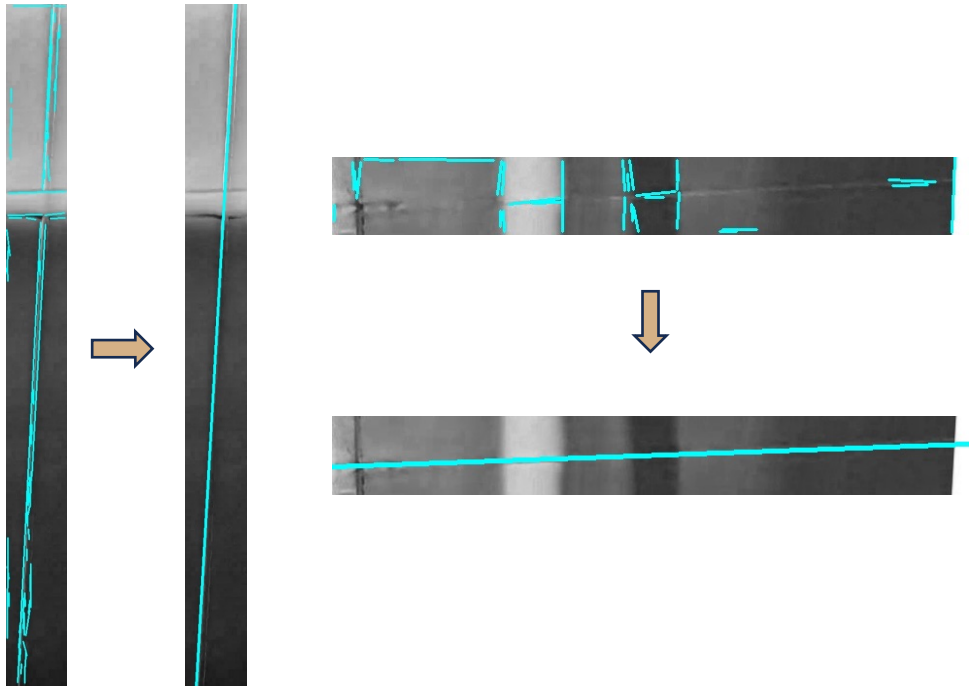
도장면 인식 결과

딥러닝 기반 도장면 및 경계면 인식

❖ 다양한 상황에서 경계면 인식을 위해 딥러닝 활용

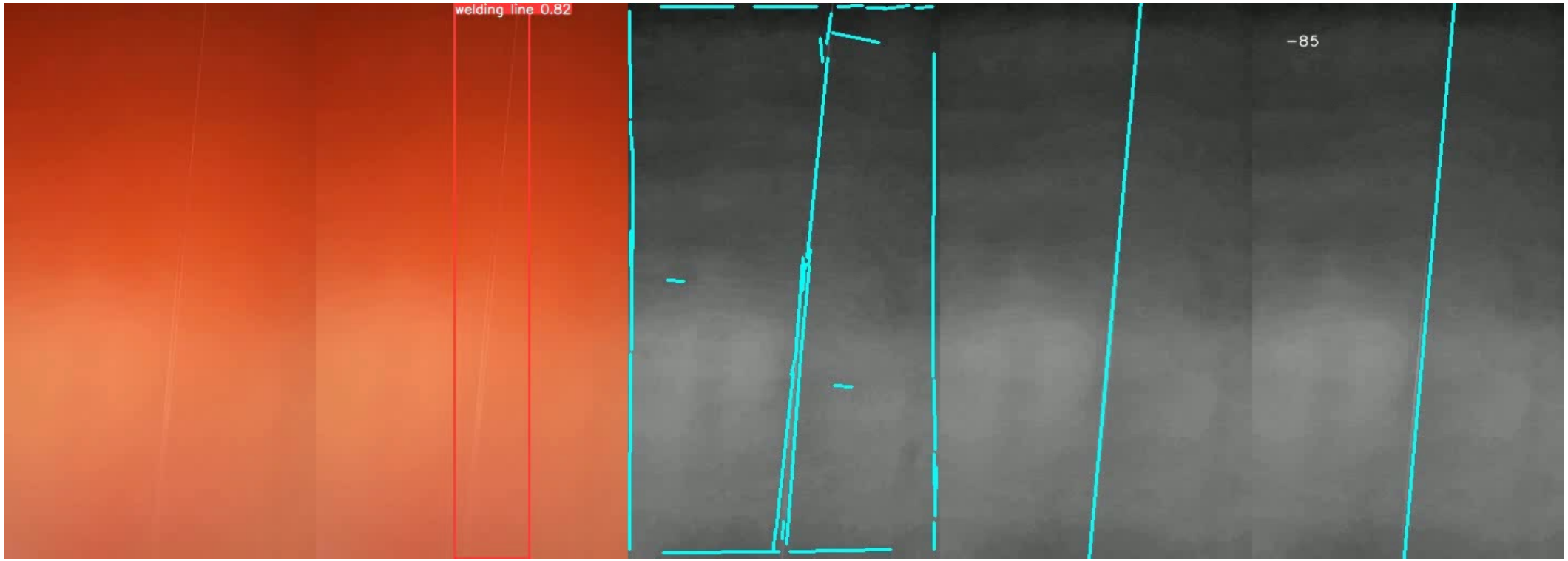
❖ 기준선 인식 방법

- Airline을 이용하여 영상 프레임에 존재하는 선 정보를 검출한다.
- 먼저 수행한 YOLO의 결과를 활용하여 bounding box 내부의 선만 남긴다.
- 그 중 기울기, 길이 정보등을 활용하여 적절한 선만 남긴다.
- 해당 선들을 이용 해 curve fitting을 수행한다. 이 때 선의 길이 정보를 활용하여 가중치를 부여한다.



detecting 된 가로 및 세로 용접선(기준선)

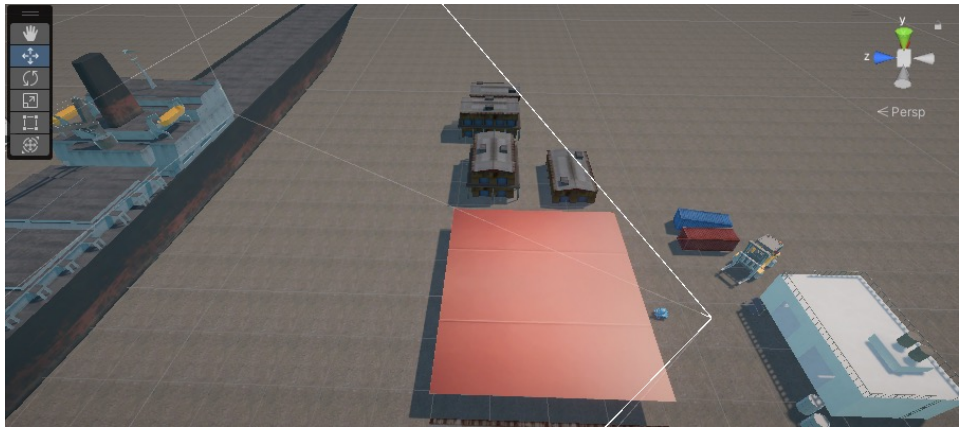
❖ 영상 적용 및 과정



원본영상 YOLO Airline line detection noise제거 및 각도 표시

검출결과를 활용한 시뮬레이터 구현

❖ 결과 검증을 위해 앞선 결과를 활용하여 자율주행 시뮬레이터 구현



❖ 시뮬레이터 구현 방법

- 스마트 야드 환경과 비슷한 도장 환경 조성
- 물리엔진을 활용하여 실제 차량과 유사한 움직임 구현
- 이미지 프로세싱 결과(기준선과의 거리, 기준선 기울기, 진입점과의 거리)를 활용하여 제어신호 생성 및 차량 상태 검출
- 생성된 제어신호 및 차량 상태를 이용하여 자율 주행
- 도장 작업 중 주행정보 및 기준선/진입점 정보, 차량 상태 등을 화면에 출력

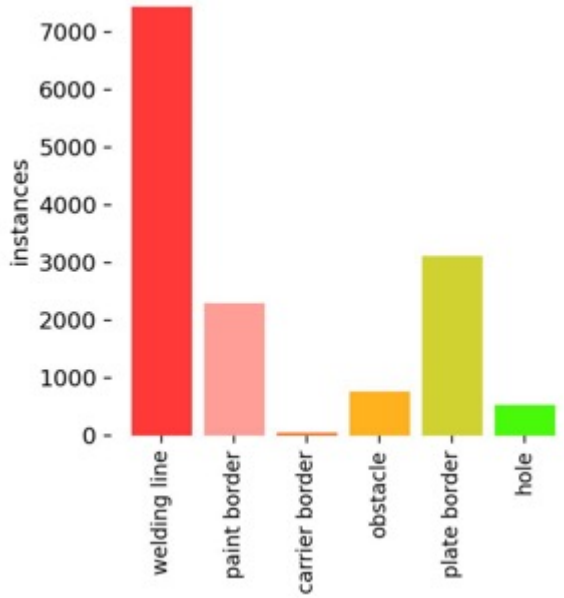
자율주행 도장작업 시뮬레이터 실행 화면

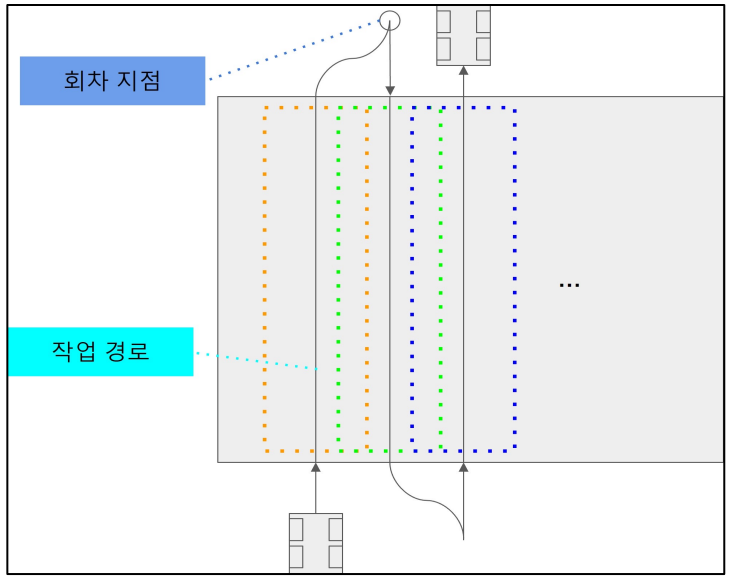
❖ 추가 Data 확보

- 일부 class는 data가 턱없이 부족, 지속적인 하부도장환경 데이터 수집으로 모델 성능 증대 필요

❖ 작업경로 생성 모듈 및 작업차량 위치 추적 프레임워크 개발

- 작업경로 생성 및 추적 프레임 워크 개발로 유사 시 작업 차량 원격제어 및 자동화 수준 향상





작업경로 생성