

Myoung Geun Jang Portfolio

소개



전공	
2015.03 ~ 2022.02	홍익대학교(서울) 전자전기공학부
SW 프로젝트 경험	
2020.07.01 ~ 2020.10.20 2022.07.05 ~ 2022.08.19 2022.08.22 ~ 2022.10.07 2022.10.10 ~ 2022.11.25	자율주행 드론(Python 딥러닝 영상처리 기반) / 동상 4족 보행 아이 케어 로봇 '보비' / 우수상 ROS시뮬레이터 기반 자율주행 배송 차량 '레스고' unity기반 증강현실 어플리케이션 '싸룬안' / 우수상
직무 관련 경력	
2022.01 ~ 2022.12 2022.11 ~ 2022.02	삼성청년SW아카데미(SSAFY) 임베디드SW 개발 프로그래머스 자율주행 영상인식 및 자율주행SW개발
주요 개발 기술 스택	
C++	PYTHON
OPENCV	LINUX
ROS	GIT
JIRA	SQL
C/C++ Python OpenCv	ESP32 로봇 자세 제어, 알고리즘 최적화, 임베디드 시스템 인지/판단/제어 알고리즘 구현, 카메라/레이더/라이다 센서퓨전, IoT 개발, DB 데이터 통신 차량 카메라 영상처리, Yolo/haar를 이용한 객체 탐색, 주행 차선 검출



자율주행 드론



4족 케어 로봇



자율주행 배송 차량

자율주행 드론 개요



프로젝트 : 자율주행 드론(OTD, Object Tracking Drone)

수행기간	- 2020.06.29 ~ 2020.10.21 (115일)
수행목표	<ul style="list-style-type: none"> - 목표물까지 스스로 비행하며, 위급시에만 수동조종이 가능한 Level 3 자율주행 드론을 구현 - 진행경로상의 모든 장애물을 회피하도록 설계 - 드론의 영상을 실시간으로 분석하여 실시간 자율주행을 구현 - 딥러닝 영상처리 과정에 있어서 객체 인식 오류가 발생하였을 경우, 해결책을 구축 - 드론을 지속적으로 tracking 하여 외부 환경요인으로 인한 위치변화 최소화
알고리즘	<ul style="list-style-type: none"> - Python Yolo를 구현하기 위한 딥러닝 프레임워크 OPENCV 활용 - Python 영상처리 모듈 CV2 를 활용한 드론 실시간 영상 수신 - Python 여러 모듈들 NUMPY 등을 활용한 전반적인 OTD 프로세스 구현 및 세부 데이터 파싱 - 객체 데이터와 드론에 내장된 IR센서의 데이터 값을 토대로 비행경로 설정
사용언어 & 프로그램	- PyCharm, YOLOv4 딥러닝 영상처리 모델, IR 센서, Python OPENCV, NUMPY, PyCharm, 녹스애플레이어



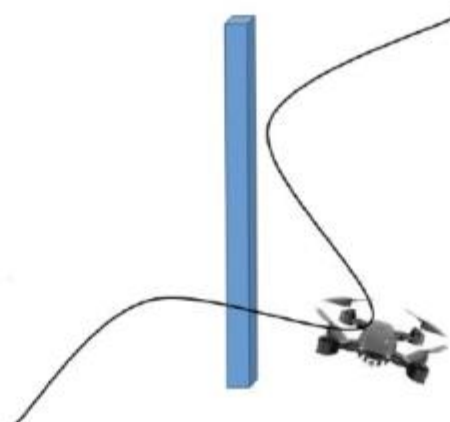
<Codrone II>

NO Human Control



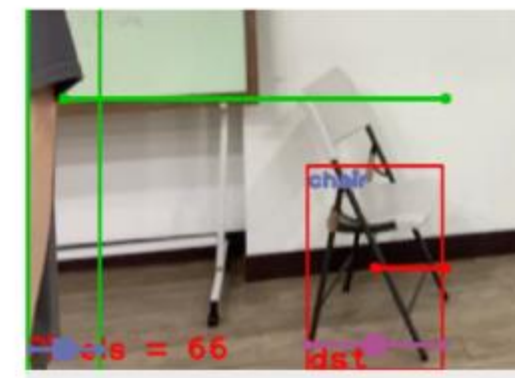
인간이 직접 드론 조종 X

Obstacle Detour



장애물 회피

Object Tracking



목표 객체 추적 & 도착

자율주행 드론 프로젝트 - 영상처리 프로세스



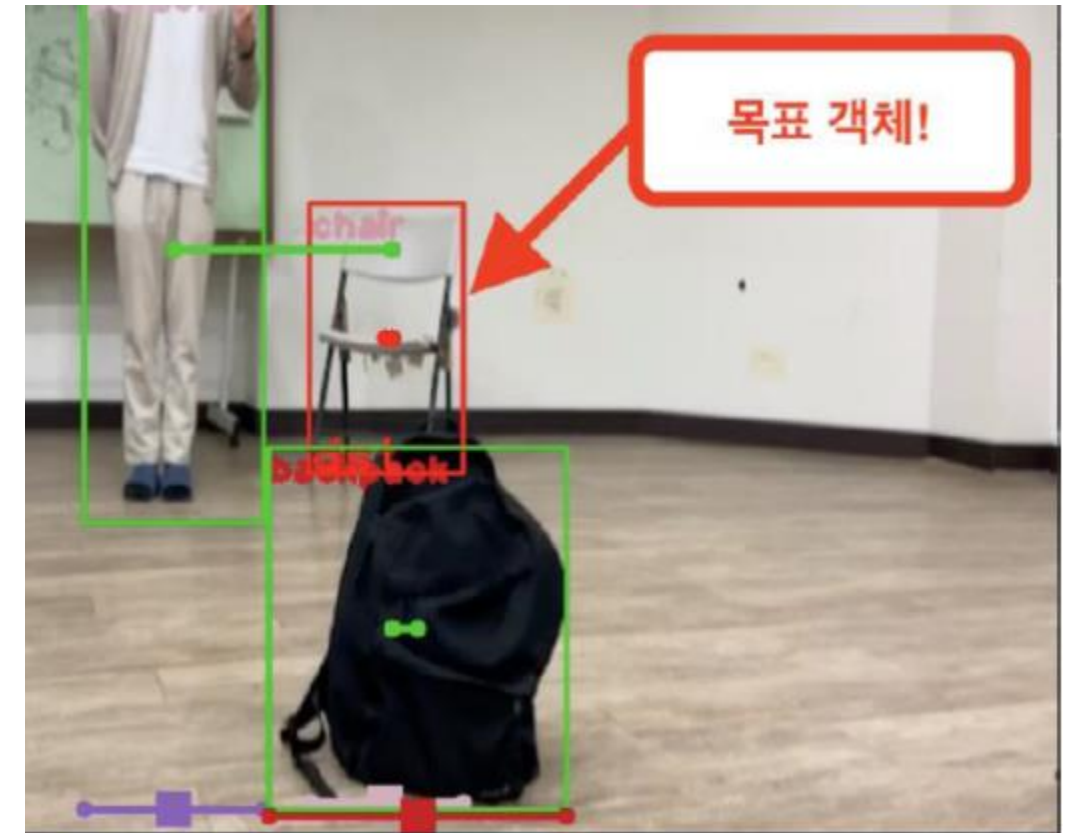
영상 이미지



화면상 객체 추출



object 종류 판단



object 위치 및 크기 판단

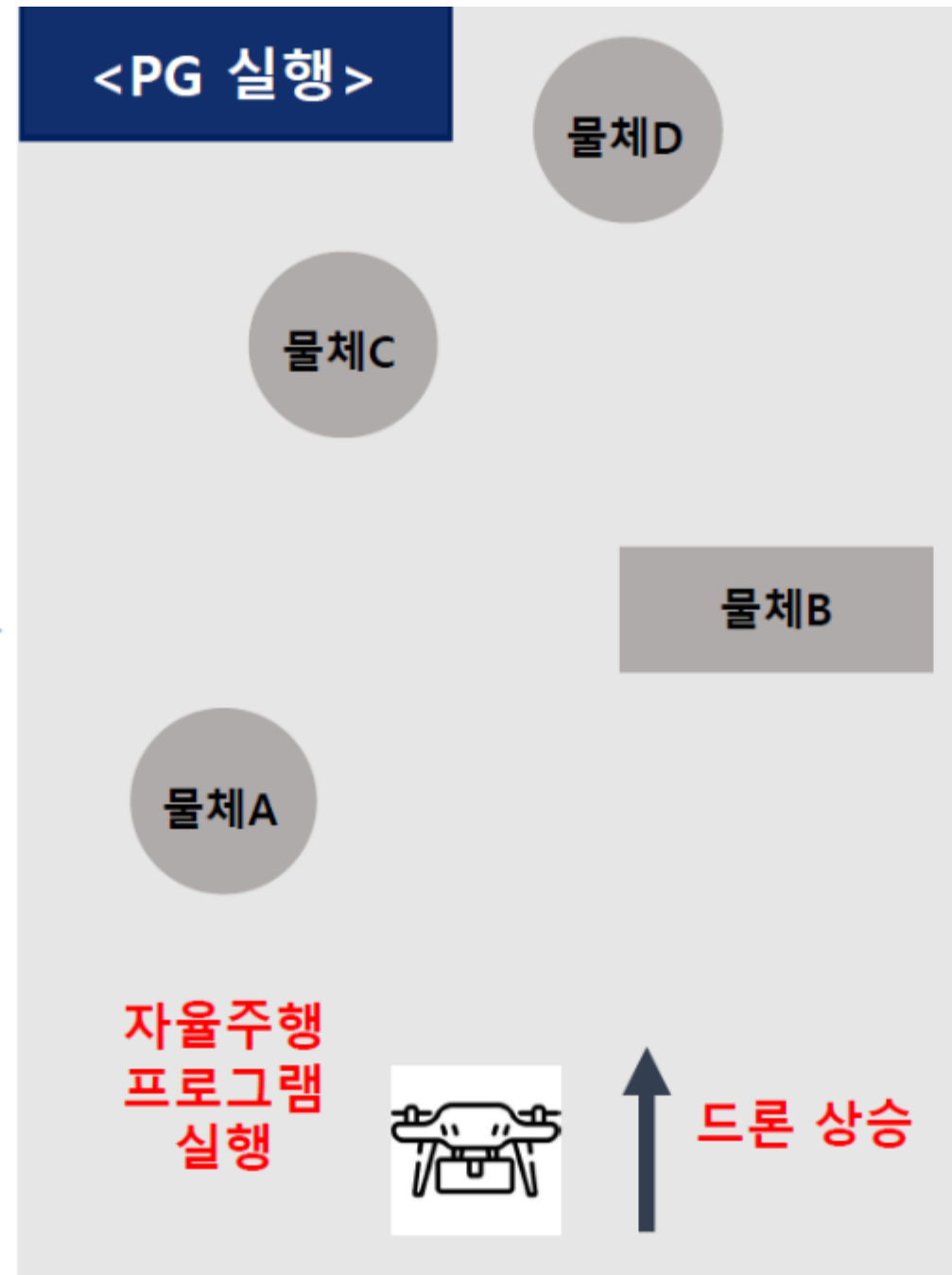
- 각 Box의 **중심 좌표**를 참조하여 드론의 **회전 및 경로**를 설정.



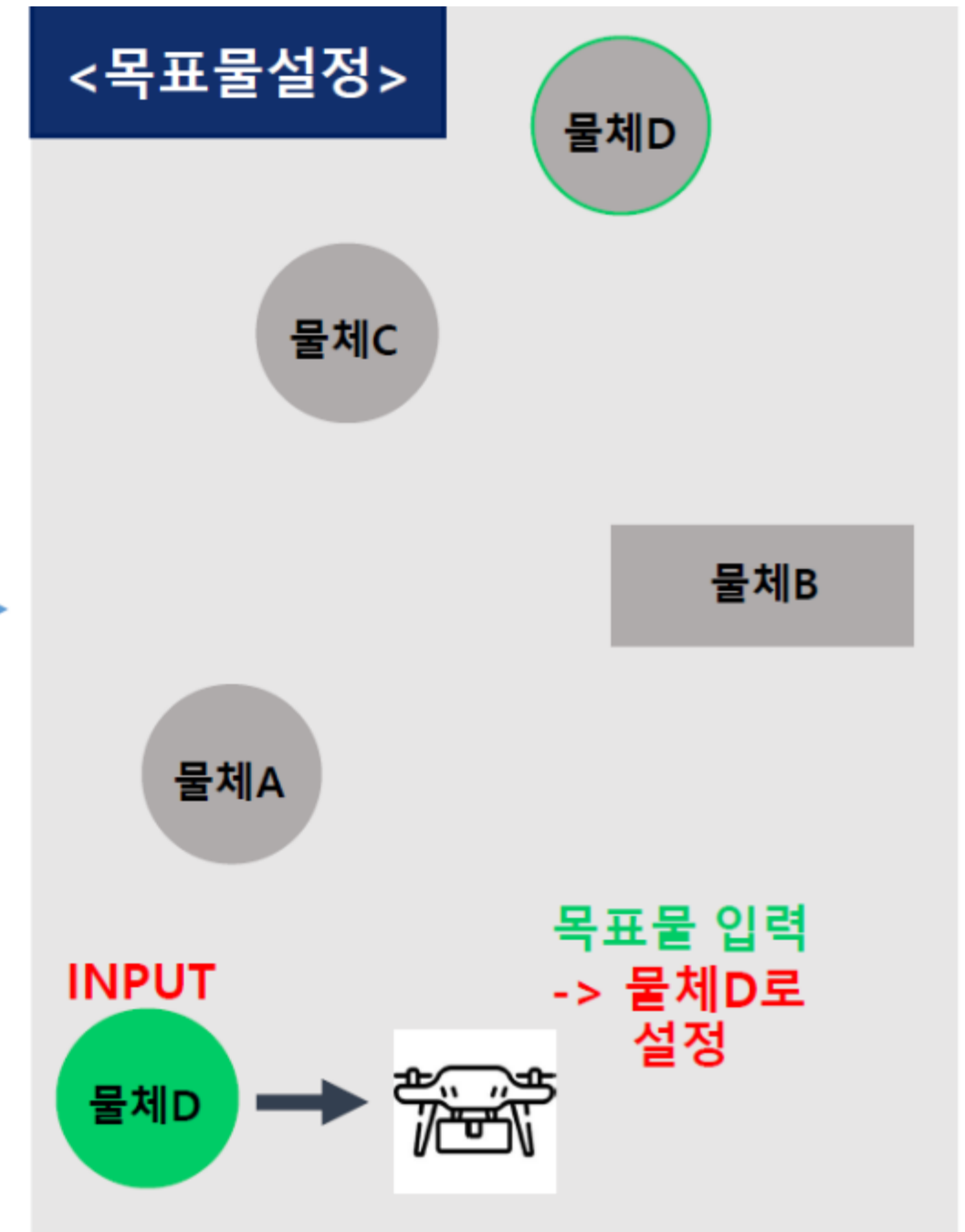
자율주행 드론 - 프로세스1



- 드론을 임의의 장소에 놓습니다

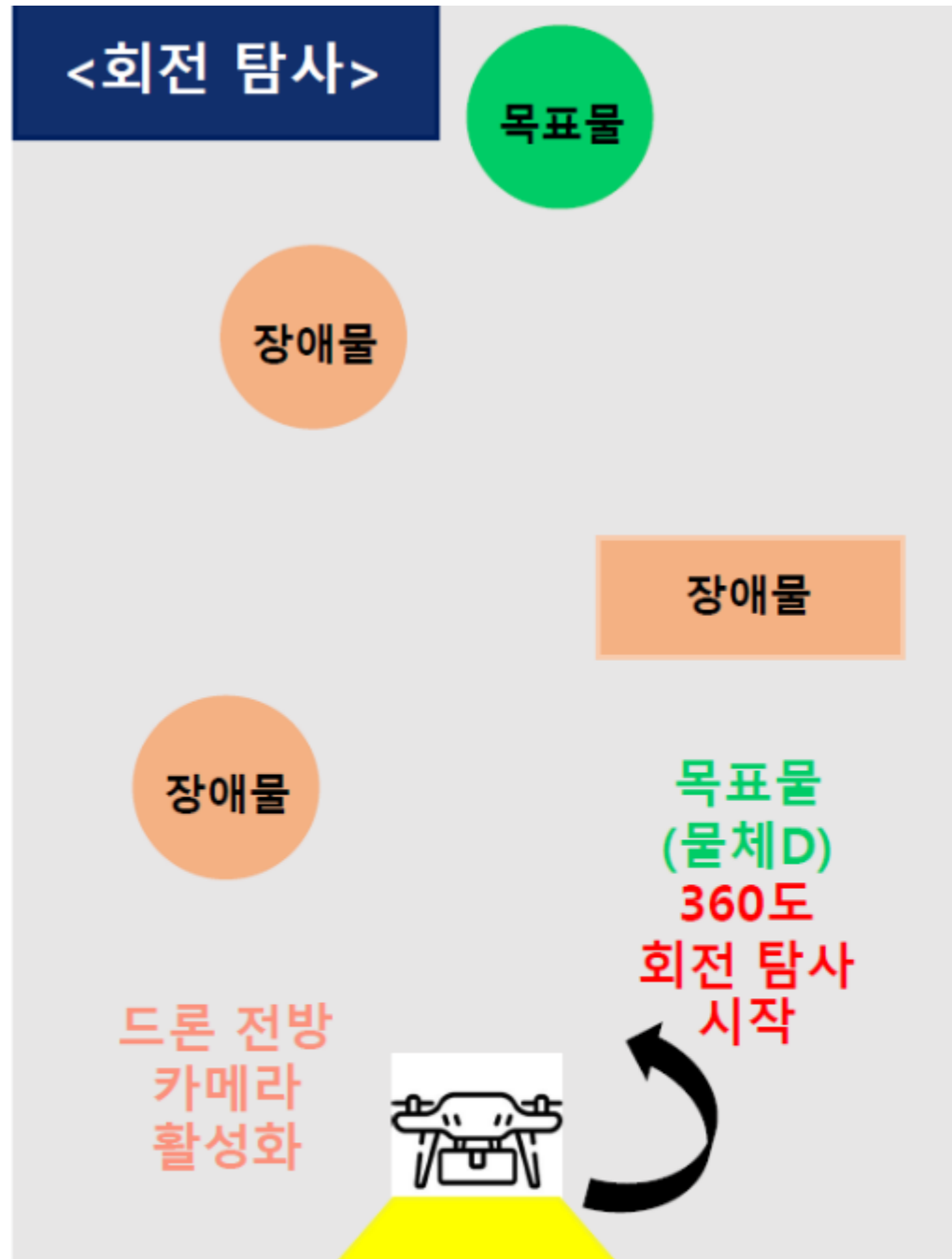


- 자율주행 프로그램을 실행시키면, 드론이 1m 상승합니다



- 드론이 찾아갈 목표물 사진을 입력해줍니다.
(본 상황에서는 가장 멀리 있는 물체D로 설정하였습니다.)

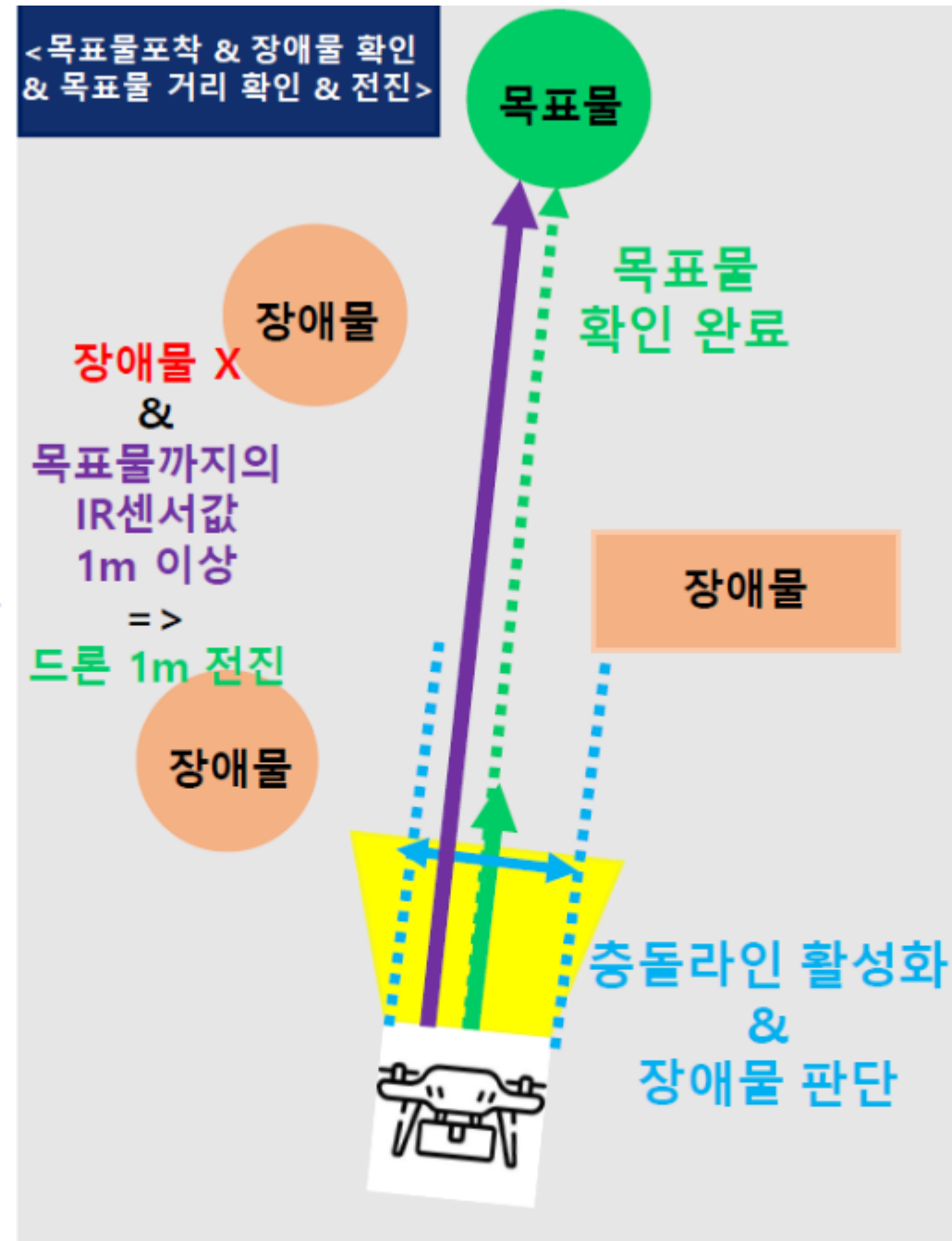
자율주행 드론 - 프로세스2



- 드론은 전방 카메라 상에 인식되는 물체들중, 목표물이 존재하는지 탐사합니다.

-만일 카메라 상에서 목표물을 찾지 못하였을 경우, 드론은 360도 회전하면서 목표물을 연속적으로 탐색합니다.

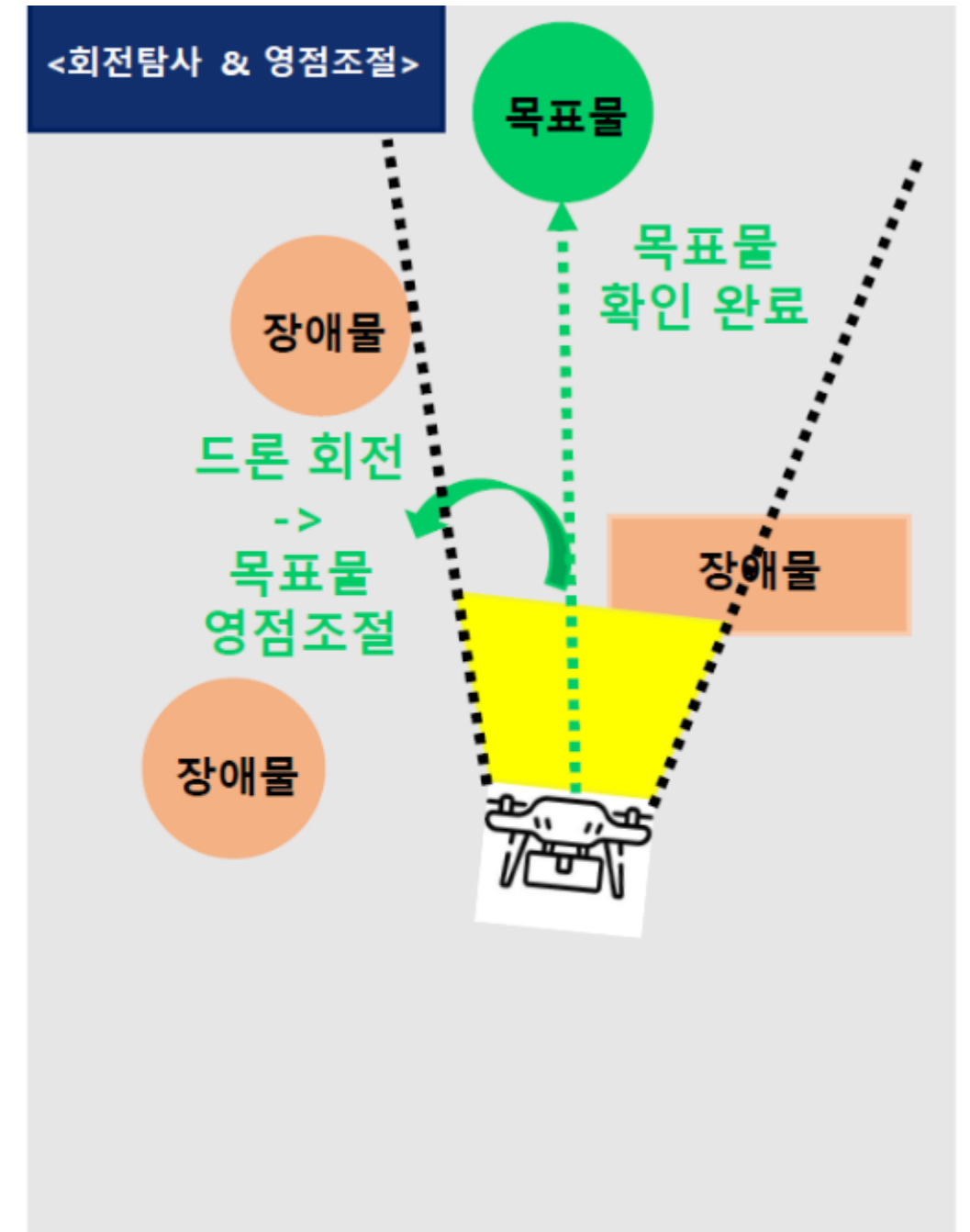
-만일 360도 회전 후에도 목표물을 찾지 못하였을 경우, 임의의 방향으로 0.5m 전진 후 다시 360도 탐사를 반복합니다.



- 회전 탐사 후 목표물을 포착하였다면, 드론 충돌 라인상에 장애물이 있는지 확인합니다.

- 충돌라인 내부에 장애물이 없다면, IR센서를 활용하여 목표물까지의 거리를 확인합니다.

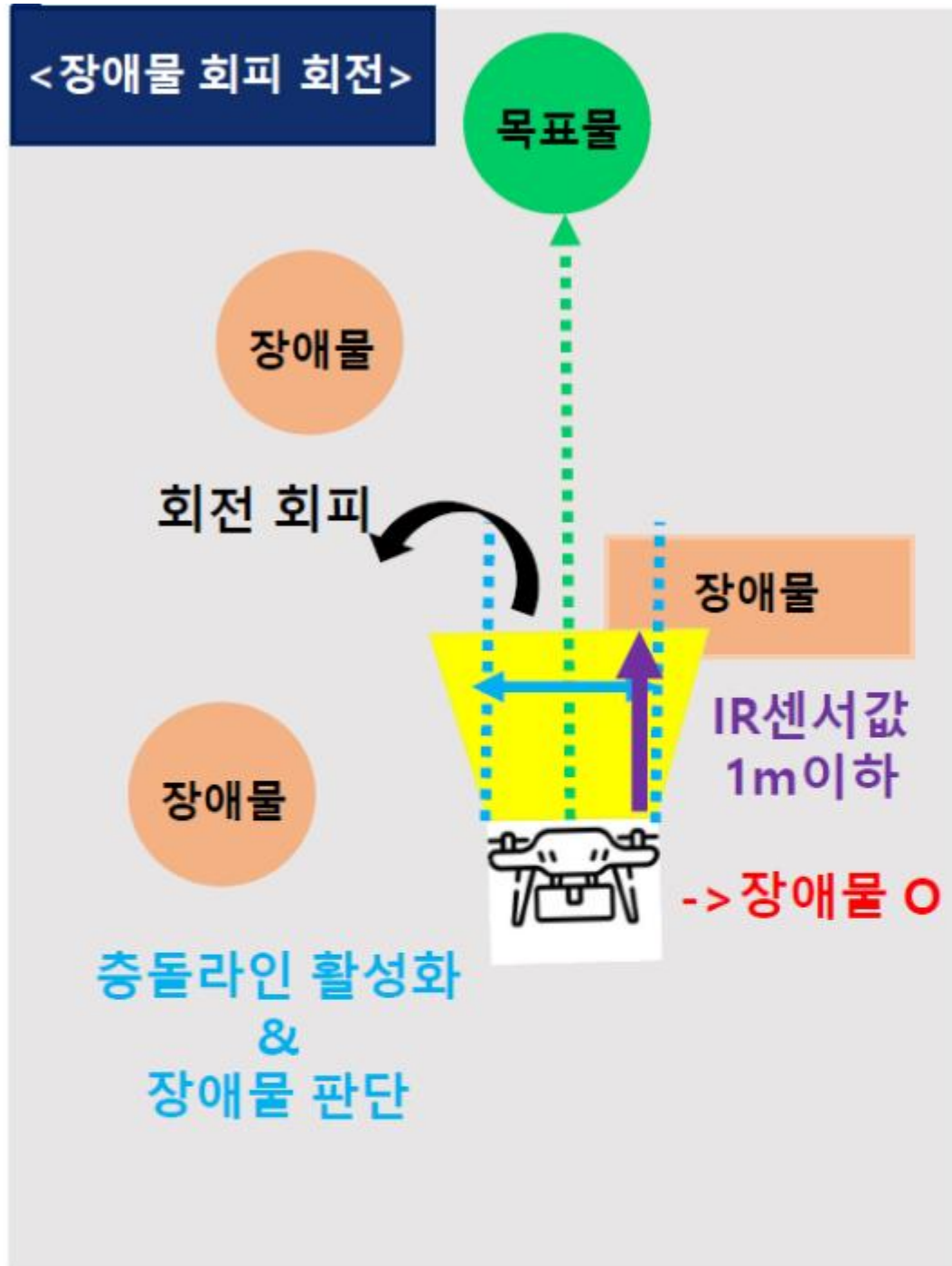
- IR센서값이 1m 이상이면, 1m 전진합니다.



- 1m 전진 하였다면, 다시 회전 탐사를 진행합니다.

- 카메라상에서 목표물을 찾았다면, 목표물이 드론 카메라의 정 중앙에 오도록 드론을 회전시킵니다.

자율주행 드론 - 프로세스3

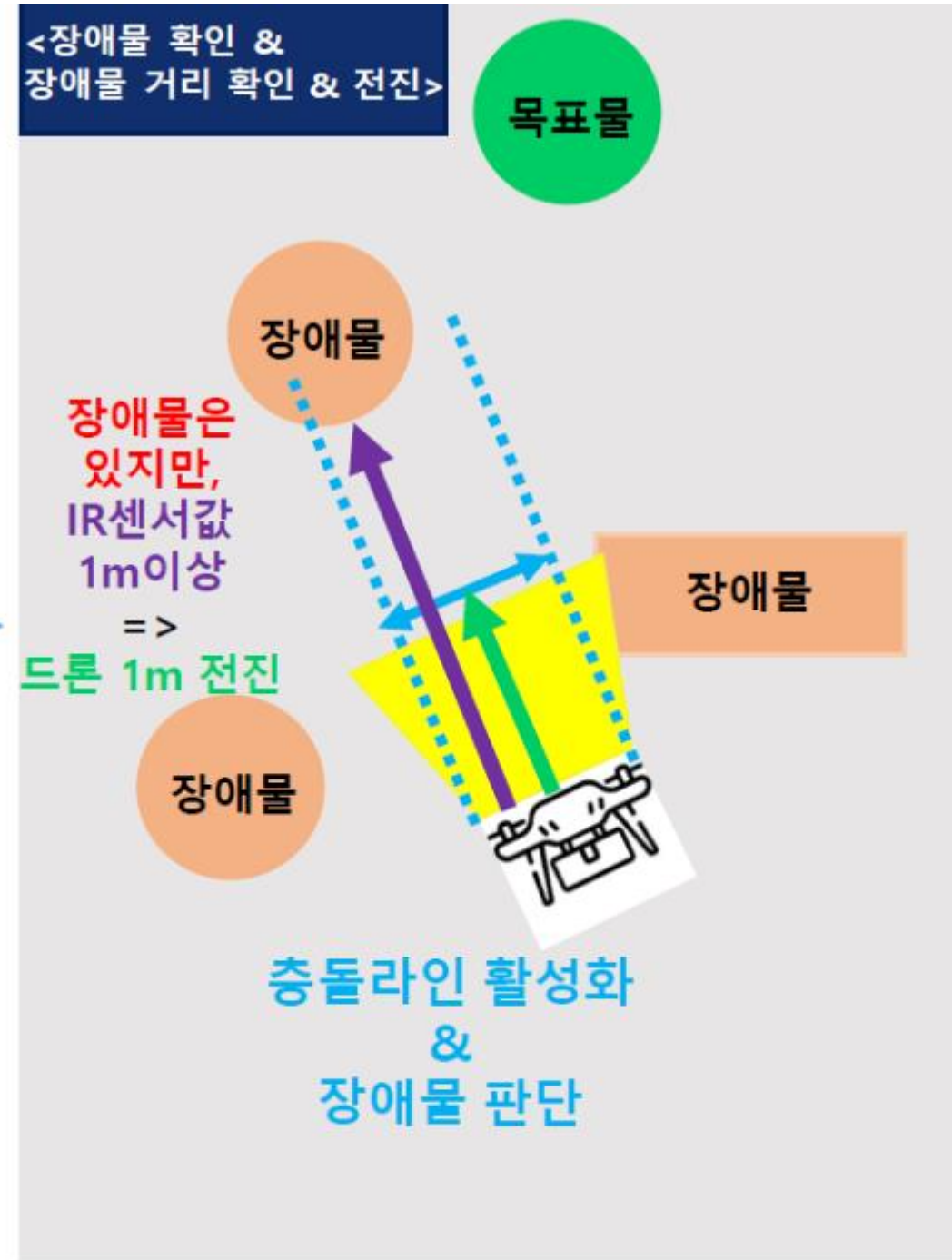


- 목표물이 카메라상의 정중앙으로 정렬되었다면, 드론 충돌라인을 활성화하여 장애물의 존재 여부를 확인합니다.

- 라인 내부에 장애물이 존재한다면, IR센서 값으로 장애물과의 거리를 측정합니다.

- 거리가 1m 이하이면, 충돌 위험 감지 후 회피 준비를 합니다.

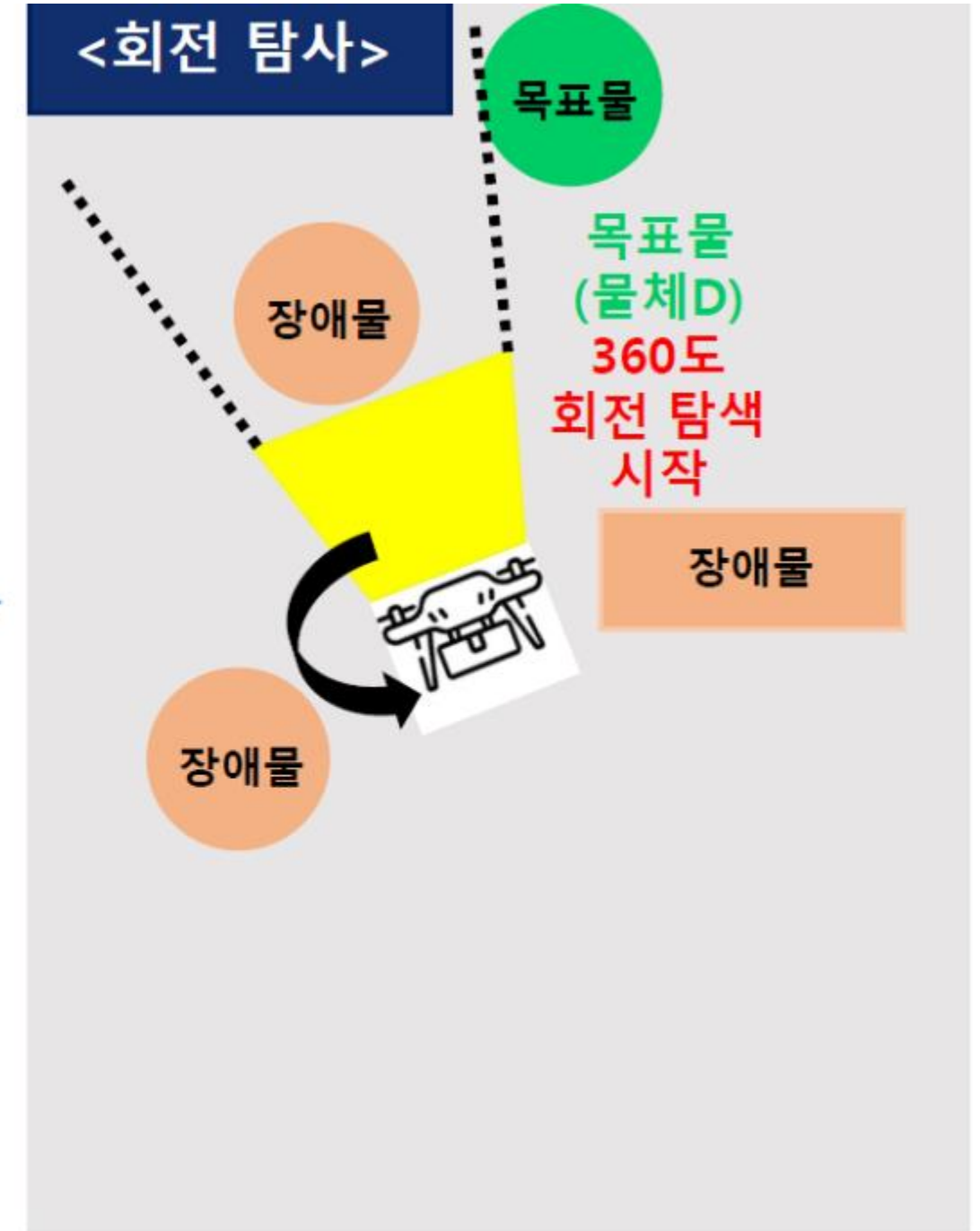
- 장애물 대한 반대 방향으로 일정 각도만큼 회전합니다.



- 회전을 통하여 충돌라인 상에서 장애물이 벗어났다면, 다시 충돌라인을 활성화 시켜서 장애물 충돌 유무 판단을 반복합니다(IR센서 값 활용)

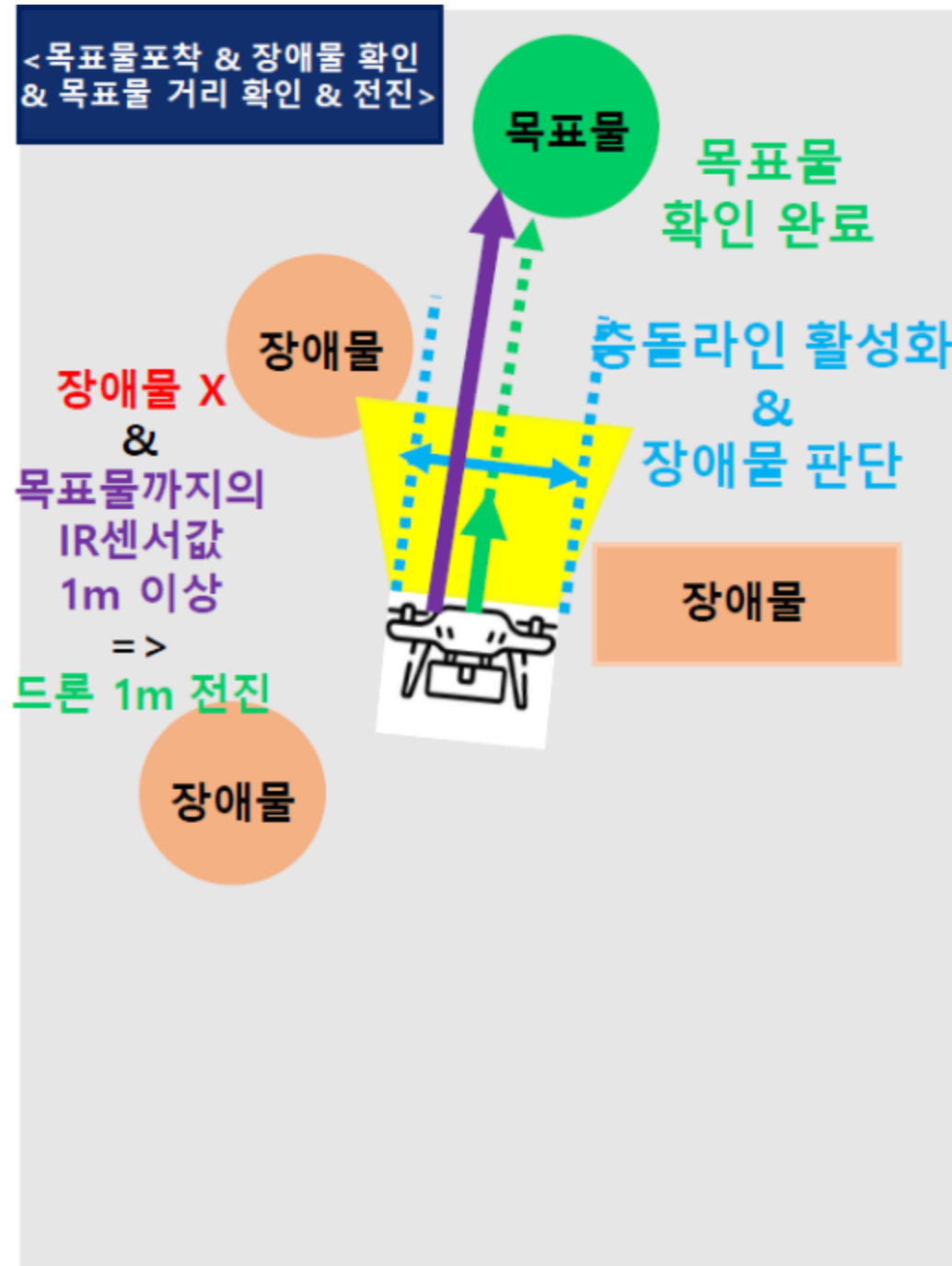
- 충돌 라인 상에 장애물은 있지만, IR센서값이 1M이상이므로, 충돌 위험은 없습니다.

- 그러므로 1M 전진합니다.



- 1M 전진 하였다면, 다시 360도 탐사를 진행하면서 목표물을 찾아냅니다.

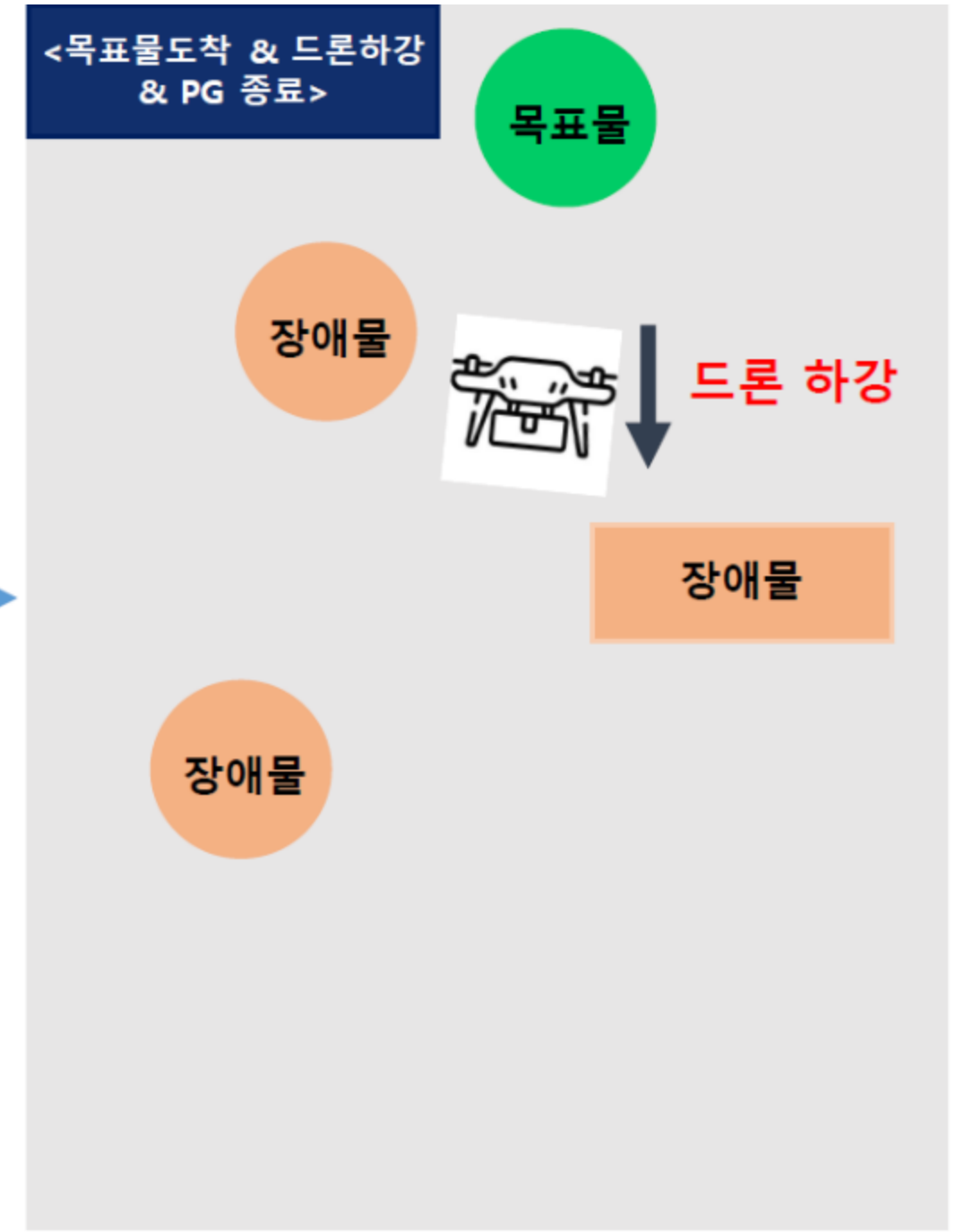
자율주행 드론 - 프로세스4



- 회전 탐사 후, 목표물을 확인하였다면, 다시 충돌라인을 활성화하여 장애물 존재 유무 및 충돌 유무를 파악합니다.
- 라인 상에 장애물이 존재하지 않는다면, 목표물과의 거리를 확인합니다.
- 목표물과의 IR센서값이 1M 이상이면, 드론을 1M 전진 시킵니다.



- 1m 전진 하였다면, 다시 회전 탐사를 진행합니다.
- 회전 탐사 후, 목표물을 확인하였다면, 다시 충돌라인을 활성화하여 장애물 존재 유무 및 충돌 유무를 파악합니다.
- 라인 상에 장애물이 존재하지 않는다면, 목표물과의 거리를 확인합니다.



- 목표물과의 거리가 1m이하라면, 목표물에 도착한 것으로 간주합니다.
- 드론을 지면에 하강시킨 후, 자율주행 프로그램을 종료합니다.

4족 케어 로봇 - 개요



<BoBi>



프로젝트 명: 아이 케어 로봇 '보비'	
수행기간	2022.07.05 ~ 2022.08.19
참가인원	6명(IoT 4명, front-end 1명, back-end 1명)
담당업무 & 세부업무	<p>담당업무: 팀장, 프로젝트 기획 ,IoT HW 및 SW개발 (기여도 80%)</p> <p>> 팀장: 팀원 선정, 프로젝트 역할 배분, 팀원간 프로젝트시 규칙 확립</p> <p>> 기획: 4족 보행 로봇 선정, 고객중심의 기능 도출</p> <p>> 개발:</p> <ul style="list-style-type: none"> - Haar cascade 를 이용한 객체 탐색 및 위치 추적 - 자율주행 알고리즘 개발 - ESP32를 이용한 자세 및 다리 제어 - 몸통 모델링 및 3D프린터를 사용하여 외관 출력
수행목표	<ul style="list-style-type: none"> - 아이를 스스로 인식하고 아이 이동시 자율이동을 통해 아이를 계속해서 팔로잉 - 영상처리에서 오류가 발생한 경우 상단에 탑재된 초음파 센서를 이용하여 아이와의 충돌을 방지 - 4족 보행시 카메라 떨림으로 인한 오류 발생 최소화 및 상호작용시 동작 구현 - DHT11을 이용한 온도 측정, MQ2를 이용한 유해가스 등을 측정하여 보호자가 Web을 통해 실시간으로 아이 주변환경 확인
사용기술	<ul style="list-style-type: none"> - Raspberry Pi4: 영상처리 및 로봇의 연산 담당 - ESP32: 12개의 모터 제어 및 RPI사이 UART통신 - OpenCv, haar cascade: 아이 탐색 및 추적을 위한 측정 - AWS: DB 구축, 보비에서 측정된 데이터값을 Web으로 전송시 사용 - MQTT: 로봇에서 측정된 센서값을 Mysql로 전송

4족 케어 로봇 - 소개



<WeaveGo>



- Google STT 기술을 활용
- 아이의 말을 Text로 변환
- 말을 알아 듣고 로봇과 상호 작용

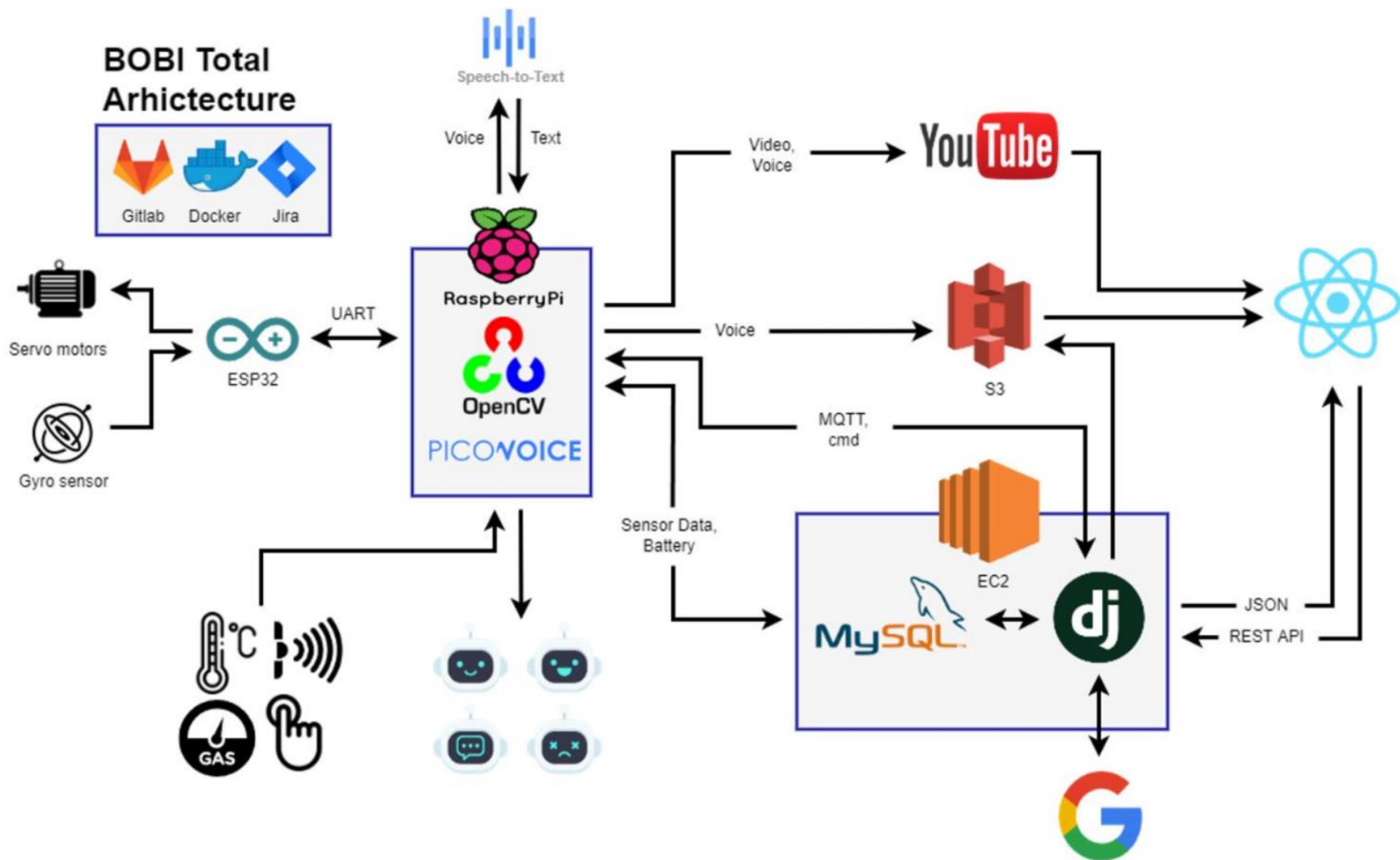


- 아이를 실시간으로 촬영
- OpenCv , haar cascade 이용, 아이를 추적 및 자율주행을 통한 아이 팔로잉

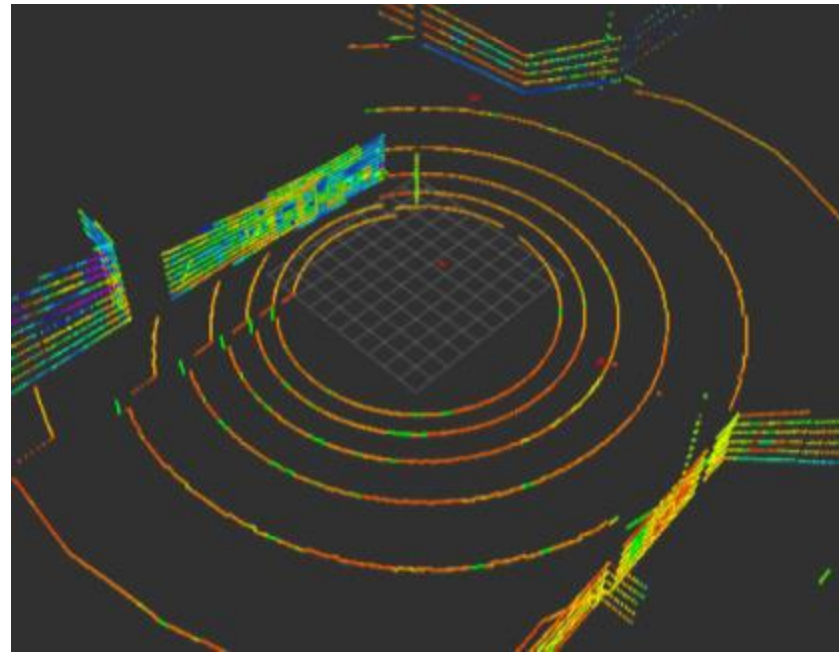


- WEB을 통해 보호자가 아이의 모습을 실시간으로 확인
- 그래프를 통해 아이의 주변 환경 상황을 확인

4족 케어 로봇 - Total Architecture



자율주행 배송 차량 - 개요



<개발시 사용된 Niro 2016>



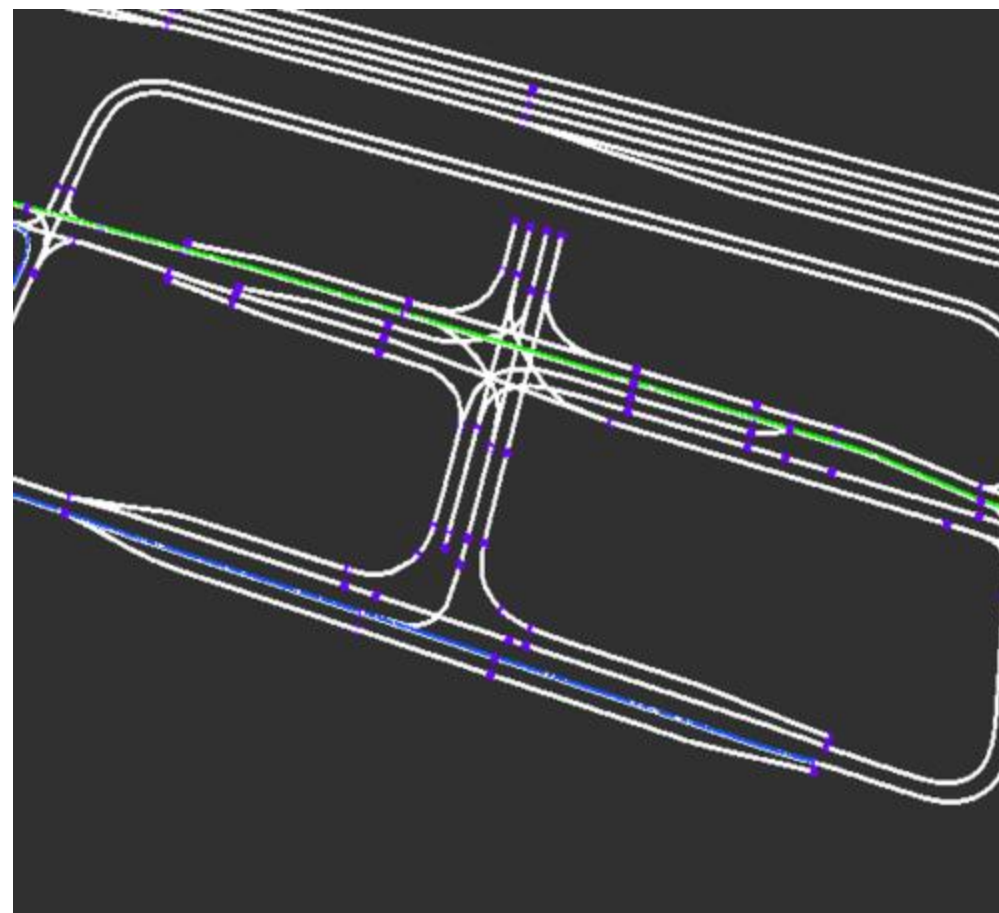
<AWS를 통해 시뮬레이터와 연동되어 작동되는 실제 모델>

프로젝트 명: 무인 자율주행 배송 차량 'LessGo'	
수행기간	2022.08.22~ 2022.10.07
참가인원	5명(IoT 4명, Web 1명)
담당업무 & 세부업무	<p>담당업무: 프로젝트 기획 , 자율주행 SW개발, 실제 차량 모델 제작 (기여도 80%)</p> <p>> 기획: 배송 차량 선정, 서비스를 위한 기능 도출 및 구현</p> <p>> 개발:</p> <ul style="list-style-type: none"> - OpenCv를 이용한 차선 검출 - PID제어, Pure pursuit 주행 알고리즘 개발 - 시뮬레이터와 실제 연동되어 구동되는 Test차량 제작 - Ros 시각화 도구 Rviz를 통한 데이터 이미지화
수행목표	<ul style="list-style-type: none"> - 카메라, 라이다 , 레이더 센서를 시뮬레이션을 통해 개발하여 각 센서의 특징점을 직접 확인 - 사용자가 목적지를 지정하면 다익스트라 알고리즘을 활용하여 최단 거리를 계산 - V2X를 통해 신호체계를 인식하여 자율주행 - 영상 처리를 통해 차선만을 추출하여 자율주행이 가능하도록 개발
사용기술	<ul style="list-style-type: none"> - MORAI: ROS기반 자율주행 시뮬레이터. 카메라, 라이다, 레이더 자율주행 필수 센서를 제공 - ROS: ROS통신 프로토콜을 이용하여 시뮬레이터와 우분투간 통신 - Ubuntu: 시뮬레이터에서 ROS통해 받아온 정보를 바탕으로 인지/판단/제어를 수행 - OpenCv: 차량의 카메라를 통해 들어온 정보 중 차선 추출시 사용 - Raspberry Pi: 시뮬레이터와 동일하게 작동하는 Test차량 제작시 사용 - Mysql: 차량 연동시 주행 데이터를 DB에 저장하기 위해 사용

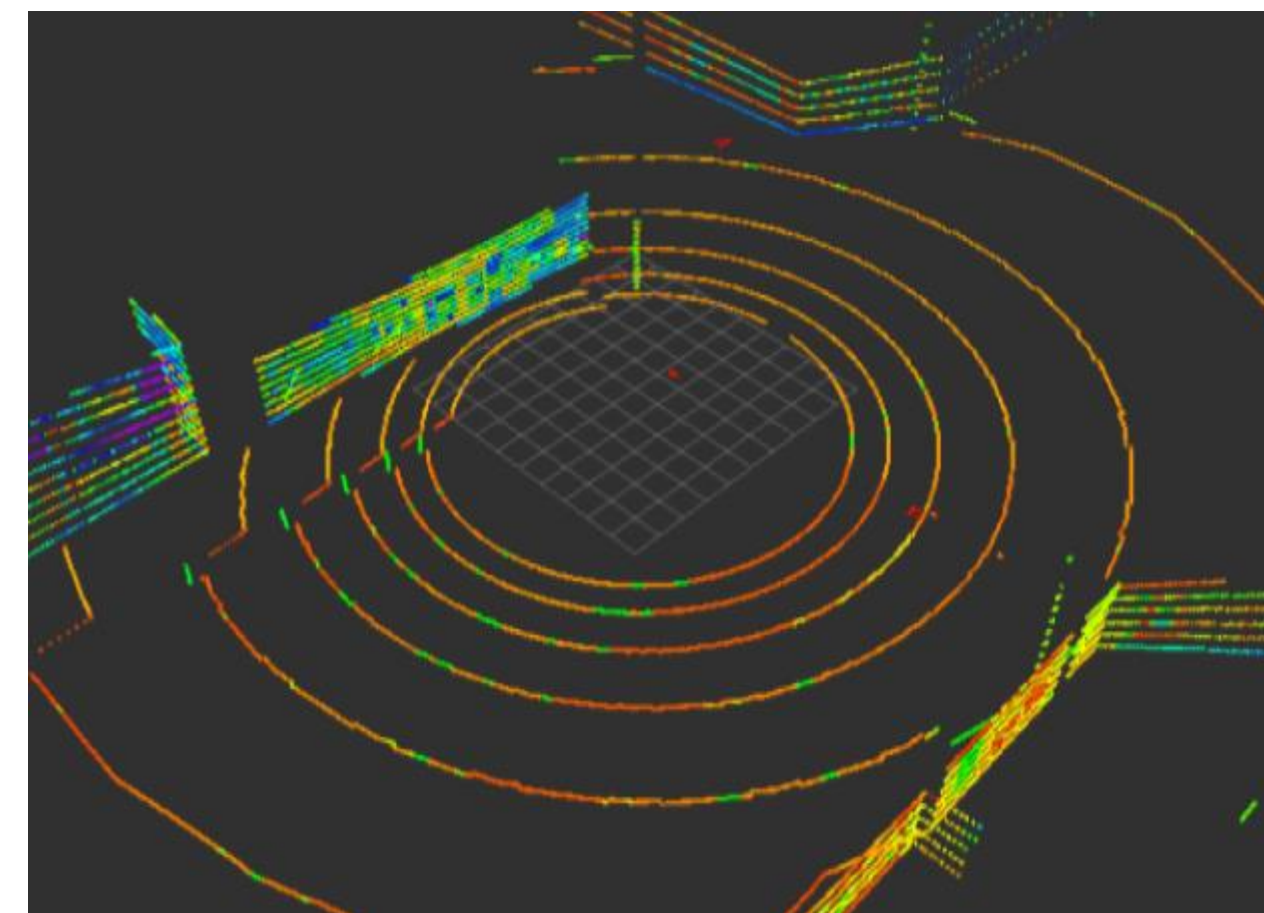
자율주행 배송 차량 - 소개



- ROS 기반 자율주행 시뮬레이터
- 실제와 동일한 센서, 환경, 모델 제공
- 게임엔진 Unity를 사용하여 현실과 동일한 물리 환경 적용
- 현대 그룹 자율주행 개발시 활용



- 목적지를 입력 후 dijkstra 알고리즘을 활용, 최단 거리를 탐색



- 카메라, 라이다, 레이더 센서 를 이용하여 주변 상황을 인지/판단
- V2X를 사용하여 신호등 및 도로 정보를 수집, 보다 안전한 자율주행 구현

자율주행 배송 차량 - 시뮬레이터 연동 차량



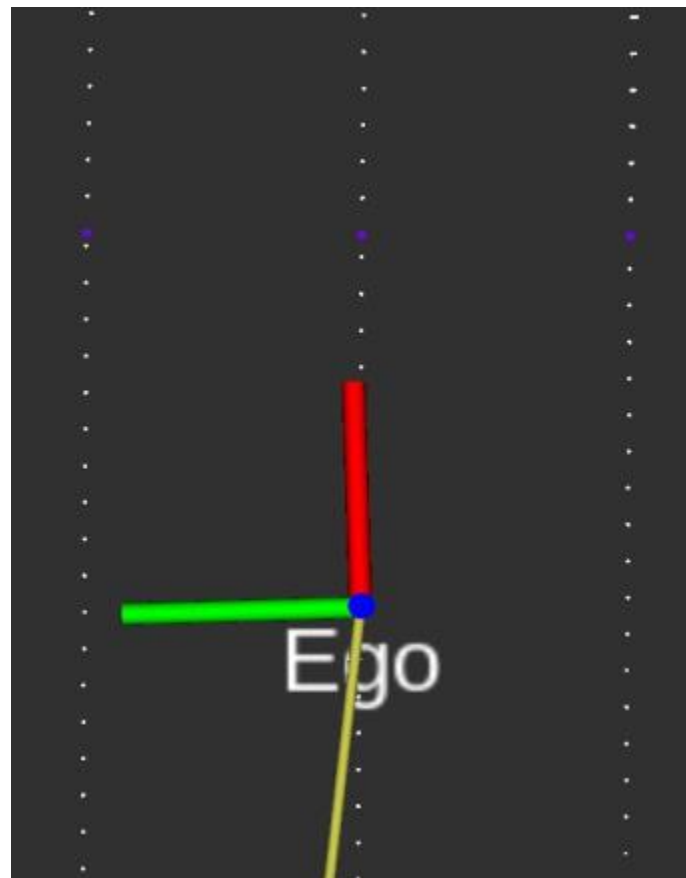
자율주행 배송 차량 - 자율주행 프로세스1

[경로생성]

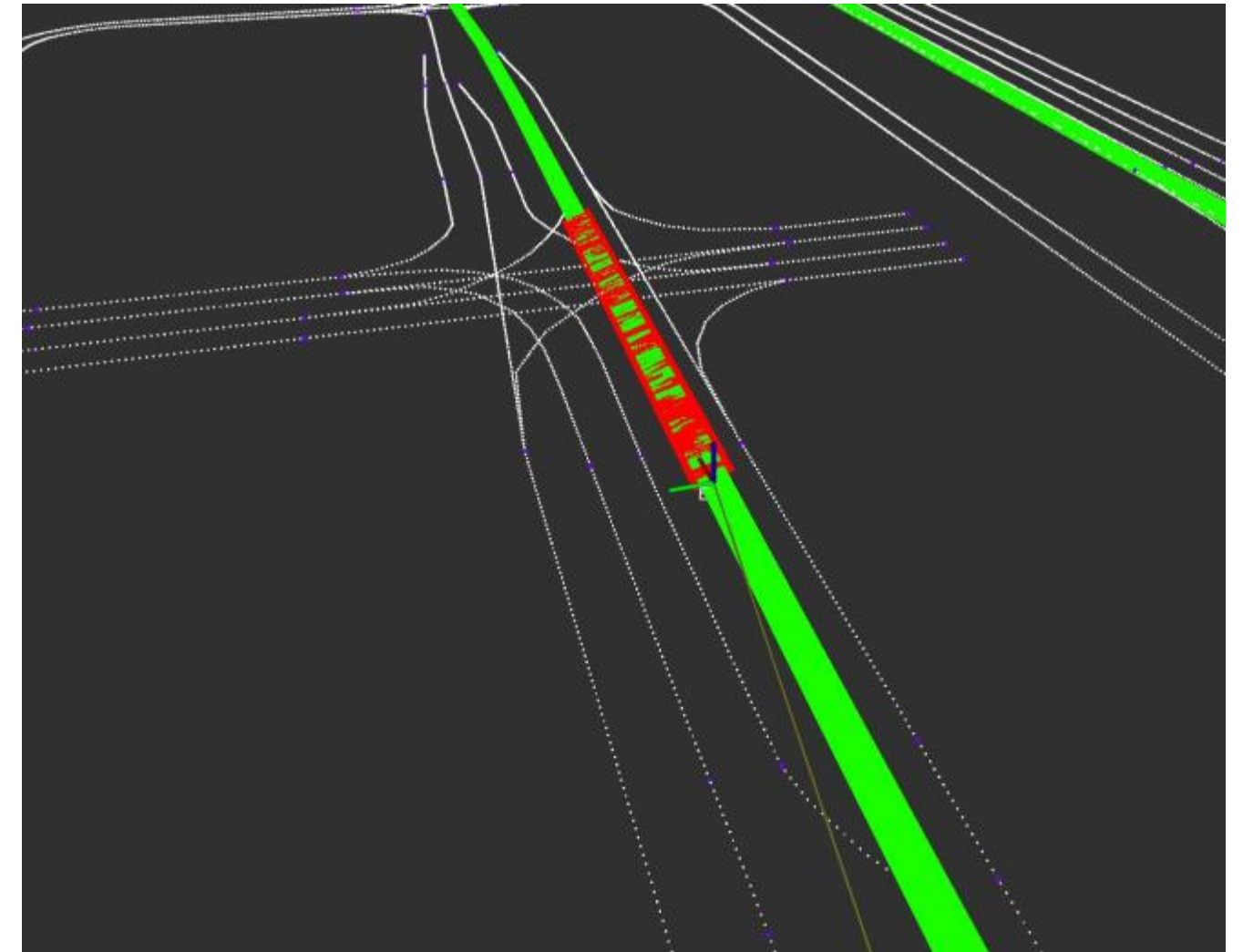
MORAI

ROS

ubuntu



정밀지도

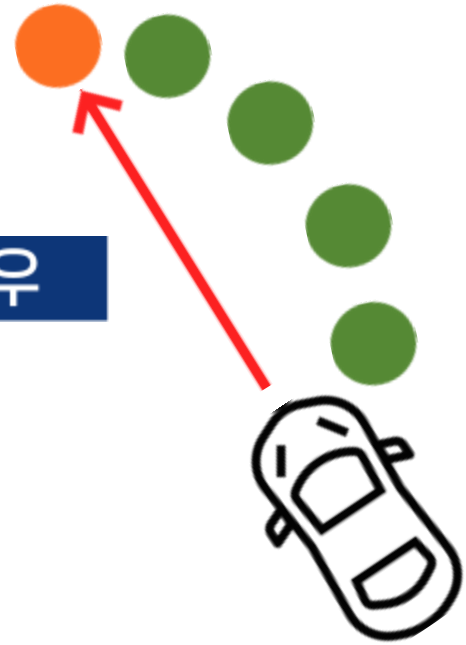


Global path 생성

자율주행 배송 차량 - 자율주행 프로세스2

LFD
(Look Forward Distance)

LFD가 큰 경우



경로를 벗어나 중앙선 침범

LFD가 작은 경우



주행시 차량이 좌우로 출렁이는 yawing 현상 발생

PID 제어 및 속도에 따른
LFD 가중치 가변화 적용



안정적인 주행 가능

자율주행 배송 차량 - 영상처리 프로세스1

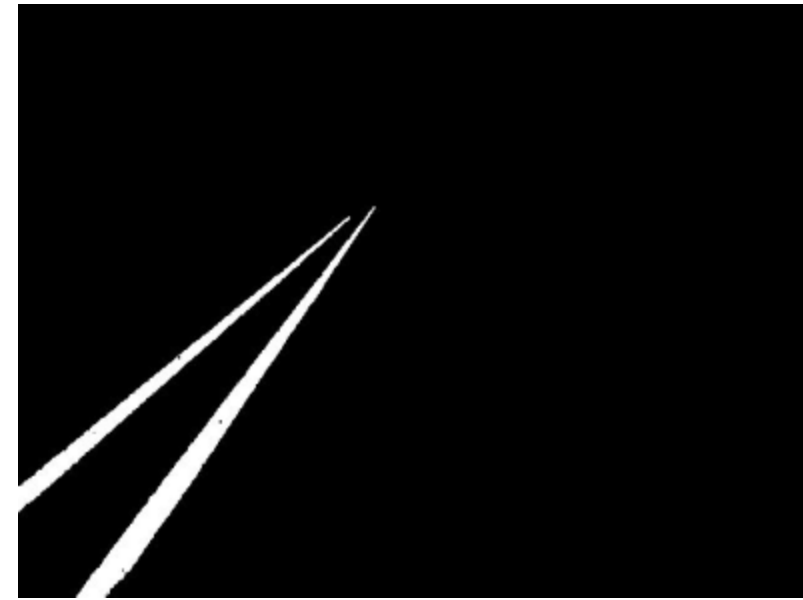
[차선 검출]



BGR 타입



HSV 타입



황색 차선 검출



흰색 차선 검출

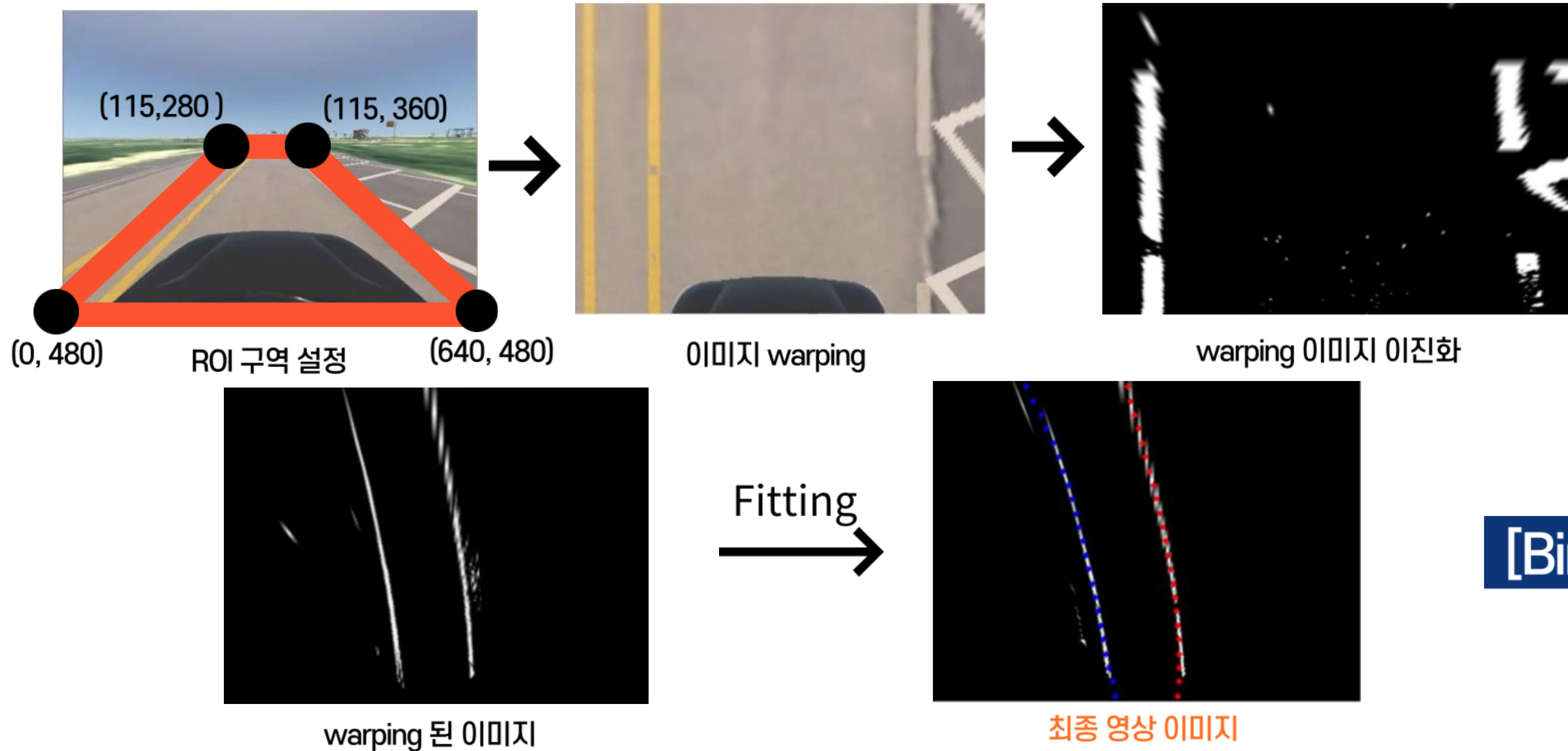


검출 차선 결합



=> **BGR** 타입에서는 차선 경계가 뚜렷하지 않다. -> **HSV**로 변환시 차선의 경계가 한층 더 뚜렷해진다.
도로 차선의 핵심인 **황색**과 **흰색**을 따로 검출하여 결합 -> 더 **정확한 차선 구분**이 가능

자율주행 배송 차량 - 영상처리 프로세스2

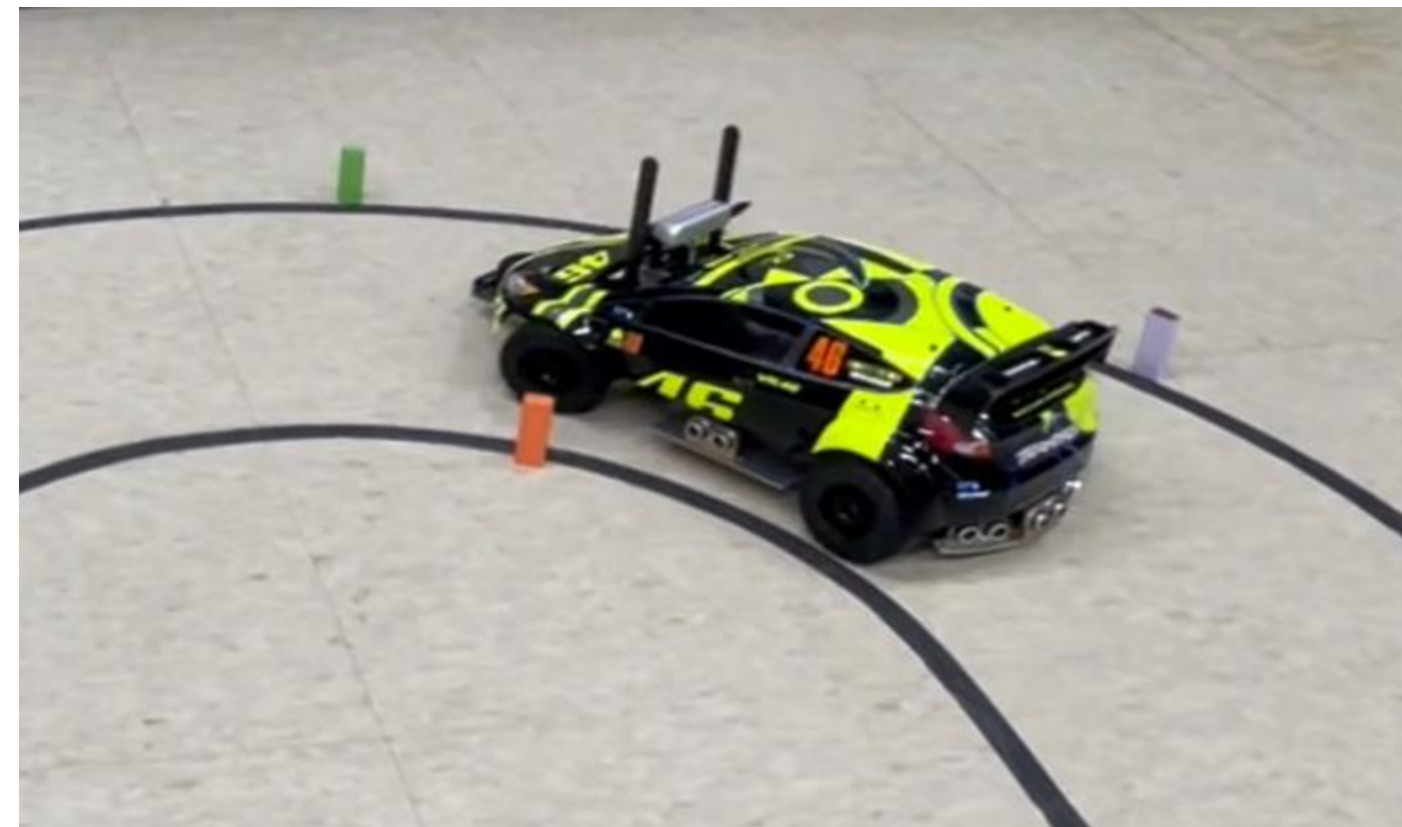
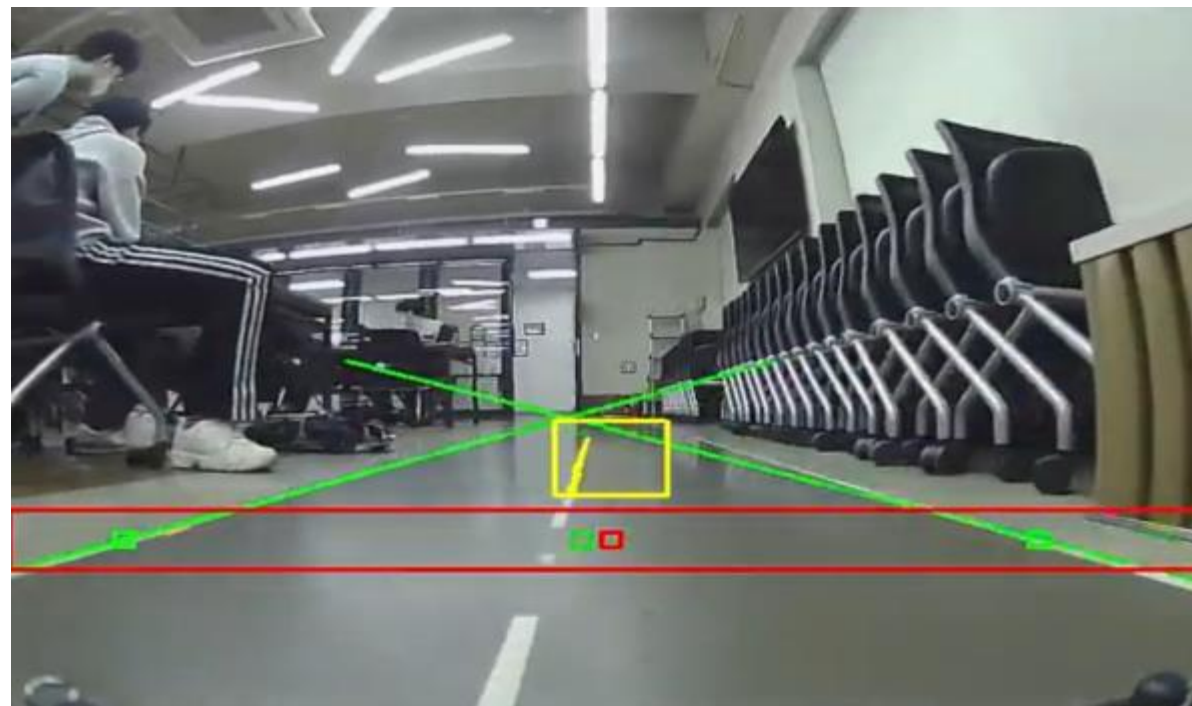
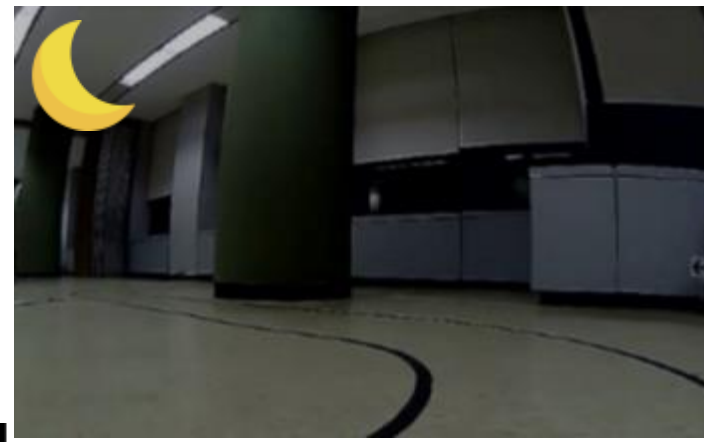


- **RoI** 영역을 설정하여 영역 외 부분은 모두 제거 후 영역 내 범위만 퍼주는 작업(**warping**)을 합니다.
- warping 된 차선을 따라 **curve fitting** 을 하여 차선을 따라 node를 그립니다.

Xycar 자율주행



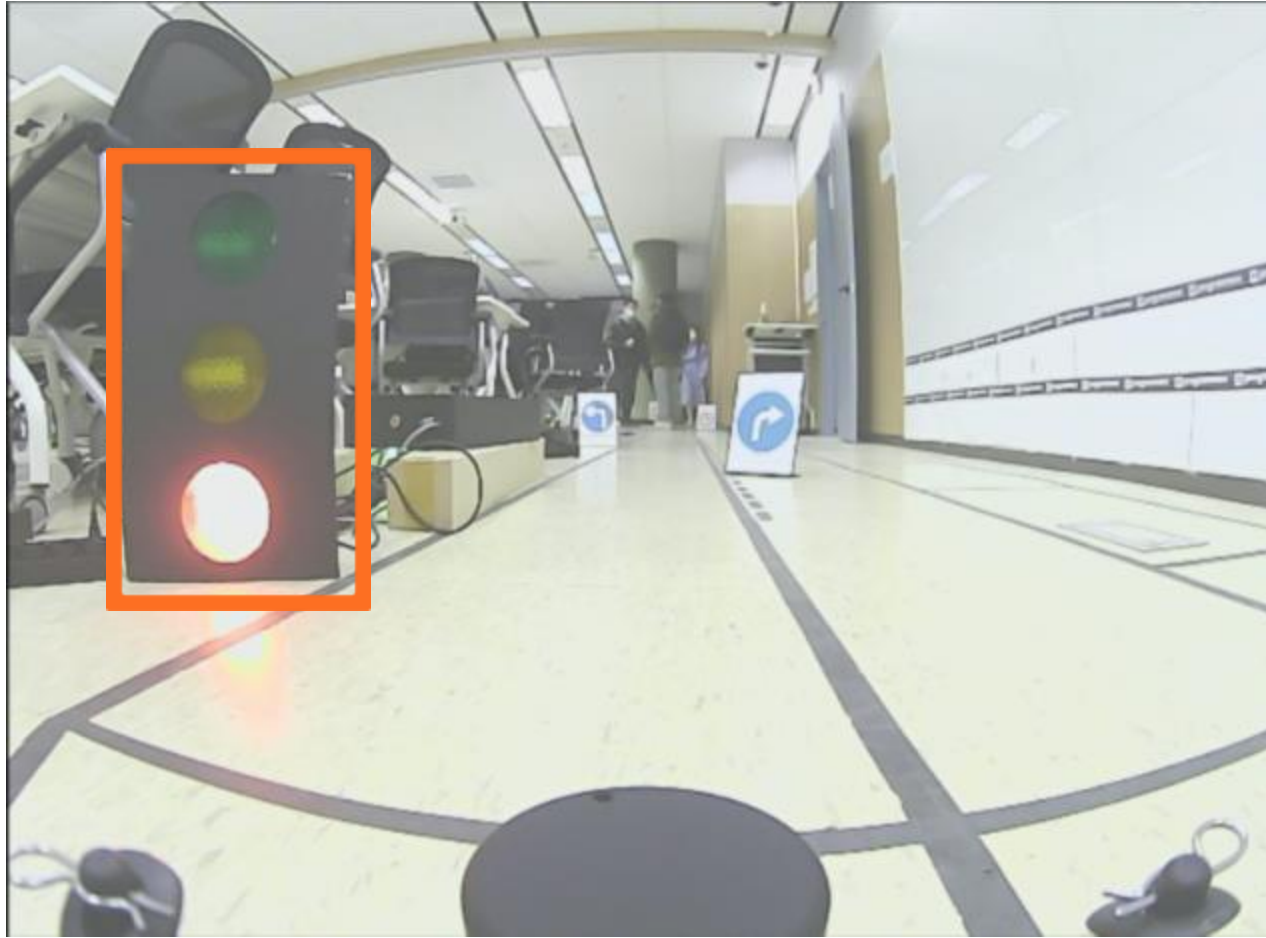
시간대별 환경 조성



프로젝트 명: XyCAR 기반 차선인식 및 자율주행 프로젝트

수행기간	2022.10.31 ~ 2022.12.23 / 참가인원: 3명
담당업무 & 세부업무	담당업무: > 개발: - OpenCv를 이용한 차선 검출 - PID제어, Pure pursuit 주행 알고리즘 실차 적용
수행목표	- 차량에 장착된 Lidar, 초음파, 카메라 특성을 이해 - 현실에서 차선을 인식하는 경우 발생할 수 있는 문제점 해결 - 하드웨어 성능에 최적화된 알고리즘 개발
사용기술	- ROS: ROS통신 프로토콜을 이용하여 차량의 센서간 통신 - Ubuntu: 시뮬레이터에서 ROS통해 받아온 정보를 바탕으로 인지/판단/제어를 수행 - OpenCv: 차량의 카메라를 통해 들어온 정보 중 차선 추출시 사용 - Jetson TX2: 실습차량에 탑재된 중앙 임베디드 시스템 - Xycar: IMU, LIDAR, 초음파센서, 카메라 등이 탑재된 1/10 크기의 실습차량

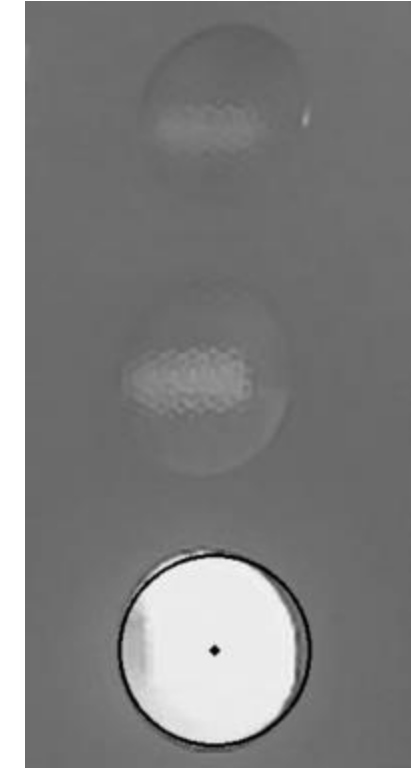
OpenCv 신호인식 프로세스



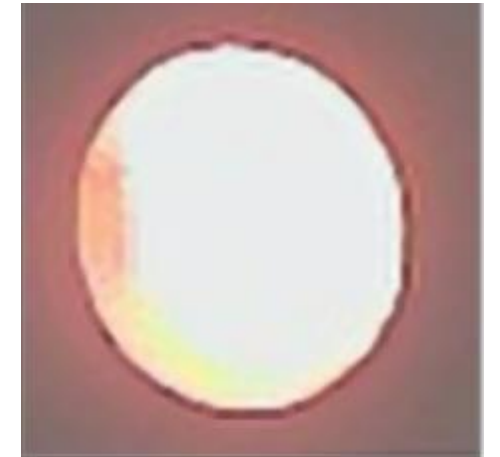
원본 영상 이미지



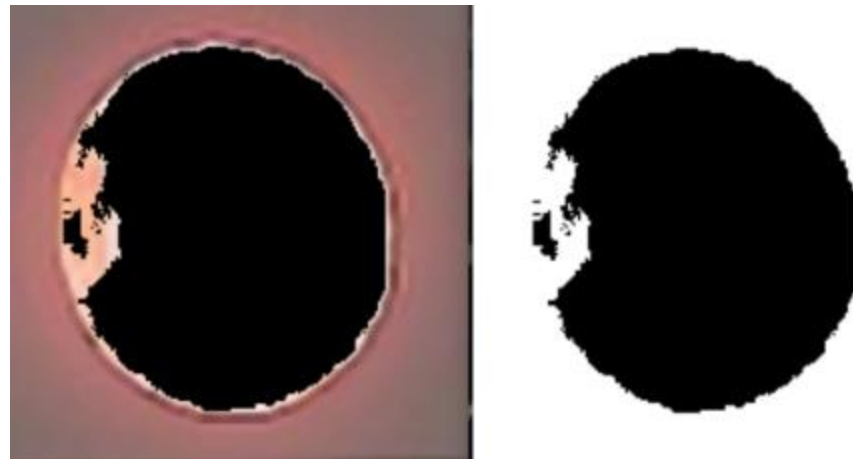
Yolo를 통한
신호등 검출



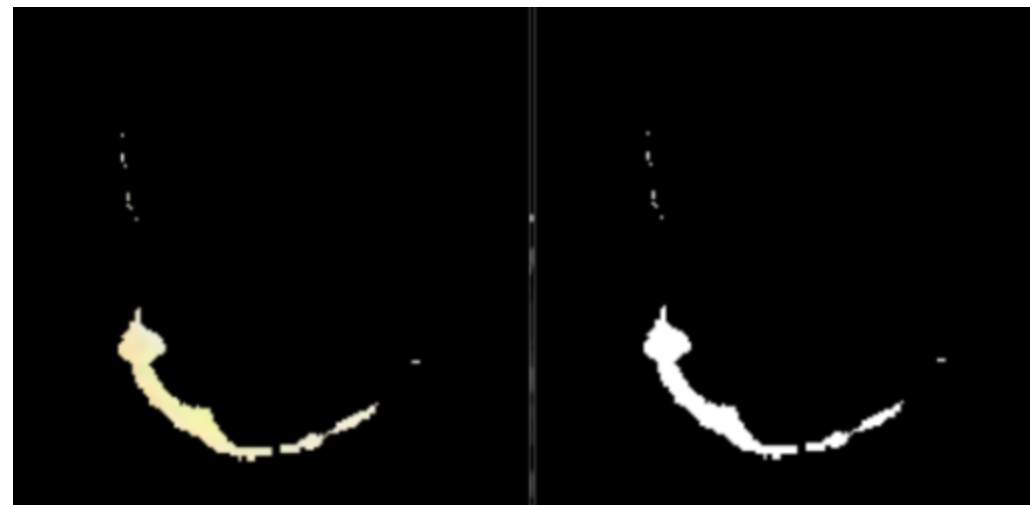
명도(value)와
Houghcircle을
통해 원 검출



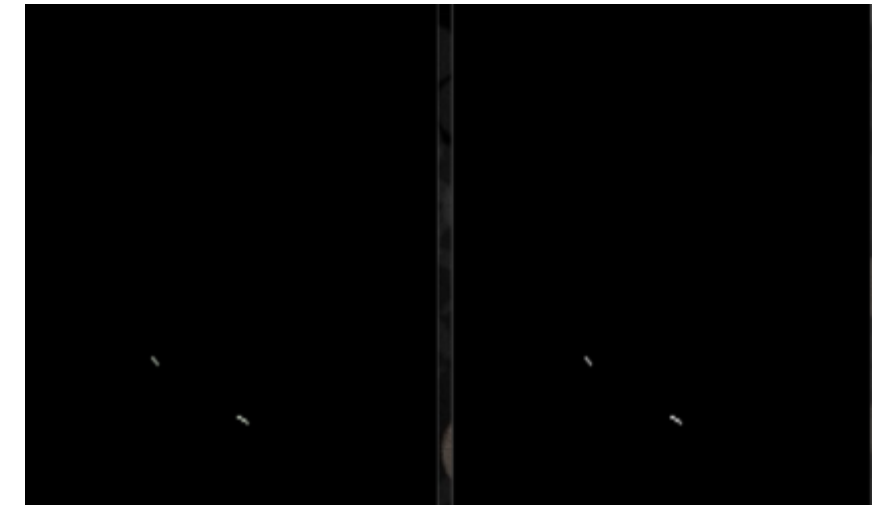
원 주변 부를 포함
해서 Roi 추출



Red



yellow



green

-> R,G,B를 HSV 변환후 masking을 통해 가장 많은 영역이 분포하고 있는 색상을 선택하여 신호를 결정.

감사합니다