

[두산로보틱스] 지능형 로보틱스 엔지니어

AMR 기반 간호 어시스턴트

SLAM(위치추정 및 공간지도생성)
기반 자율주행 로봇 시스템 구현

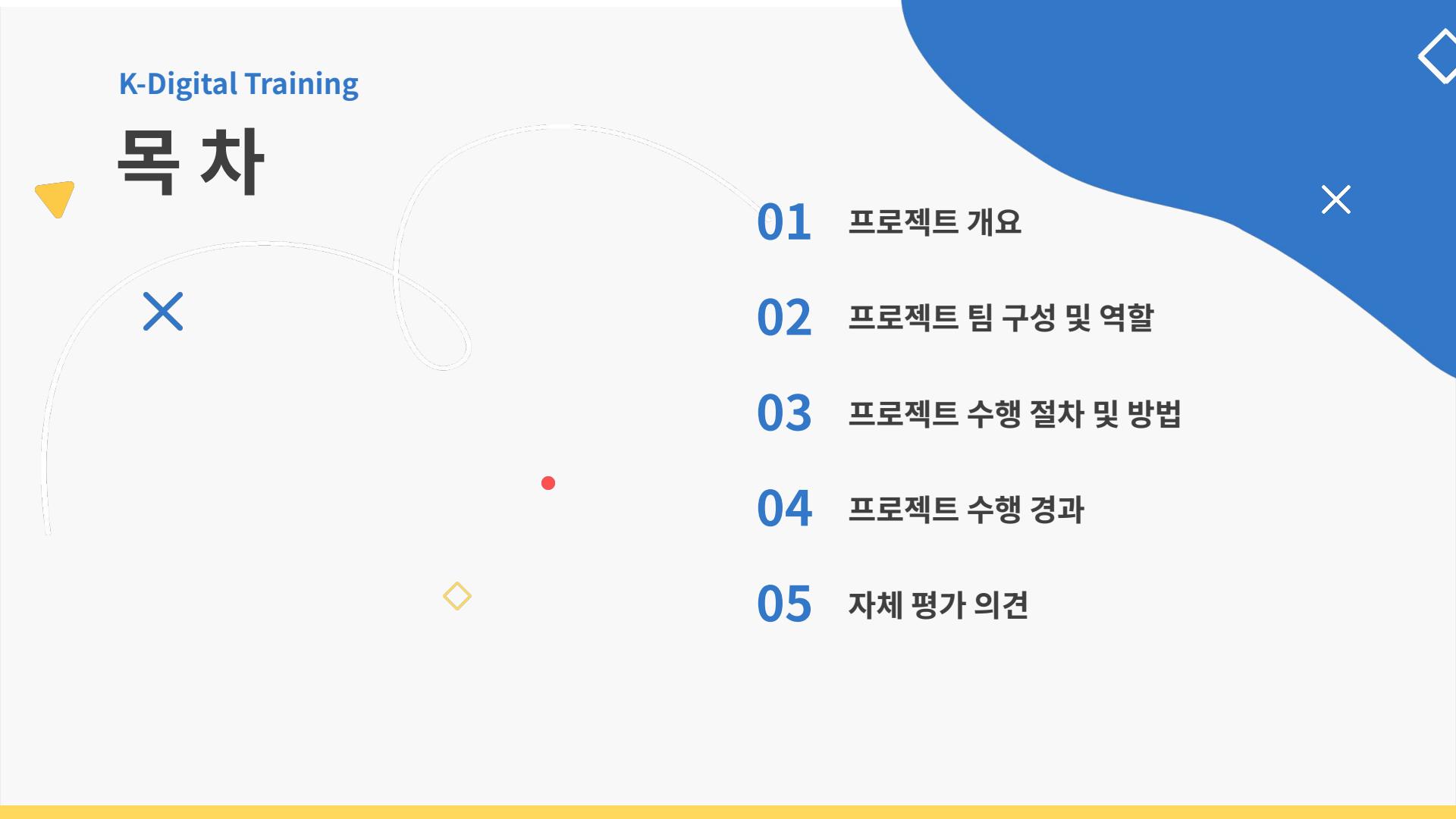
TEAM B-1, 2 B1A+4

[팀장] 백홍하

[팀원] 이하빈, 장연호, 정찬원, 문준웅, 이경민, 정민섭, 최정호

[멘토] 손미란 강사님

목 차

- 
- 01 프로젝트 개요
 - 02 프로젝트 팀 구성 및 역할
 - 03 프로젝트 수행 절차 및 방법
 - 04 프로젝트 수행 경과
 - 05 자체 평가 의견

01

프로젝트 개요

▶ 프로젝트 개요

1

프로젝트 주제 및 선정 배경, 기획의도

병원 내 간호사의 **반복적 업무**
부담을 줄이기 위해,
약품 전달과 환자 생체징후
확인을 지원하는 로봇을
개발하고자 본 주제를
선정하였다.

2

프로젝트 내용

turtlebot 4를 기반으로, CV2를 활용해 **약품을 인식하고 사람을**
감지해 정지할 수 있으며,
네비게이션과 LiDAR 센서를
이용해 병원 내 복잡한 환경에서도
안정적이고 안전한 자율주행이
가능하도록 설계한다.

3

활용 장비 및 재료

TurtleBot4 * 2
Oak d camera
Aruco marker
EMQX Cloud
ROS2 humble
OpenCV 등

4

프로젝트 구조

약품 및 사람 인식
→ 주행 제어
→ 장애물 회피 및 이동
→ 약품 전달 및 생체징후 측정

5

활용방안 및 기대 효과

물류, 안내, 감시 등 다양한
서비스 로봇에 적용 가능한
기본 자율주행 기술 확보.
클라우드 기반 로봇 간
통신으로 범용성과 확장성 강화

01 K-Digital Training 프로젝트 개요

▶ 프로젝트 배경

HOME > 보건복지/사회

"인력부족으로 의료사고까지 발생"

▲ 이인문 기자 | ○ 승인 2023.06.21 16:23 | Ⓜ 댓글 0



| 보건의료노조, 200개 의료기관 종사 48,049명의 보건의료노동자 실태조사 결과 '발표'



평일 오후 7시 까지 주문 시
당일하고!
"관점 있는 뉴스언론과의
기사제휴로 한발 더 나아가겠습니다."
세월오 사고 환생자들의
명복을 빕니다.

최신뉴스

- 부산대지파병원, 찾이기는 의료봉사 시행
- 대야치, 내부역량 강화·외연확장에 주력
- 한양대병원 치과, 연송캠퍼스 '워크숍'

실제 밥먹는 시간과 이동시간 및 휴게시간까지 포함한 평균 식사시간을 묻는 질문에 20분 미만(31.6%), 20분~30분 미만(33.2%) 등 전체 응답자의 2/3에 해당하는 64.8%가 30분 미만이라고 응답했다. 40분 이상~50분 미만(7.3%), 50분 이상(17.3%) 등 식사시간에 비교적 여유있게 식사·휴식하는 경우는 24.6%에 그쳤다.

업무량도 과중했다. "내가 담당하는 업무가 아닌 다른 업무도 처리하고 있다"고 응답한 비율이 54.7%로 과반기 넘었고 "업무량이 근무시간 내에 수행할 수 없을 정도로 과도하다"고 응답한 비율도 38.5%에 이르렀다. 응답자의 42.6%는 "내가 담당하는 업무 범위가 부적절하다고 생각한다"고 응답했다.

3교대 근무자에게 근무조별 인력이 적정한지 묻는 질문에 긍정응답 비율은 43.1%였고 부정응답 비율은 56.9%였다. 결원 발생 시 인력충원에 대해서는 부정응답이 66.5%로 긍정응답 33.5%보다 2배 가량 많았다.



Medbot, developed by Richtech, which can deliver items in a hospital

That does not mean that engineers have given up on incorporating robotics into patient care, however. Robotics allow doctors to perform surgical procedures with smaller, more precise incisions than they can with conventional tools, often by operating the machines from a console inside the operating room.

<https://www.ft.com/content/aabb7d4b-1e2d-47b9-97f4-c3fc96448aa7>



대만 대형 병원 사례 연구 (331명 간호사 대상)

- 자동화 로봇이 물품 운반 등 비전문적인 반복 업무를 대체
- 간호사들은 환자 케어 등 전문적인 업무에 집중할 수 있게 됨
- 그 결과, 직무 만족도 및 건강 상태가 향상됨
- 아직 의향 또한 눈에 띄게 감소, 긍정적인 조직 내 변화 유도

01

프로젝트 개요



최근 의료 현장 간호사 업무 과중 및 **인력 부족** 심각
더 나아가 반복적이고 단순한 업무가 **간호 서비스 질 저하**로 이어짐

💡 단순 반복 업무를 간호 로봇이 대신 수행

→ 간호사는 전문 의료 서비스에 집중

→ 업무 효율 및 만족도 향상

💡 이에 따라 병원 내 자율주행 로봇 도입하여
약품 전달 및 환자 생체징후 확인 등 단순 업무 자동화

💡 간호사 부담 줄이고 **효율적인 의료 환경 조성**

02

프로젝트 팀 구성 및 역할

▶ 프로젝트 팀 구성 및 역할

훈련생	역할	담당 업무	
백홍하	팀장	 /robot4/Vision	 Main Controller GUI
이하빈	팀원	 /robot4/Vision	 ROS 통신
장연호	팀원	 /robot4/Navigation	 EMQX Cloud 통신
정찬원	팀원	 /robot4/Navigation	 영상 편집
이경민	팀원	 /robot1/Vision	
최정호	팀원	 /robot1/Vision	
문준웅	팀원	 /robot1/Navigation	 Main Controller GUI
정민섭	팀원	 /robot1/Navigation	 바이탈 체크 기능 개발
손미란 강사님	멘토	 주제 선정 피드백, 프로젝트 질의응답	

03 프로젝트 수행 절차 및 방법

▶ 프로젝트 수행 절차 및 방법

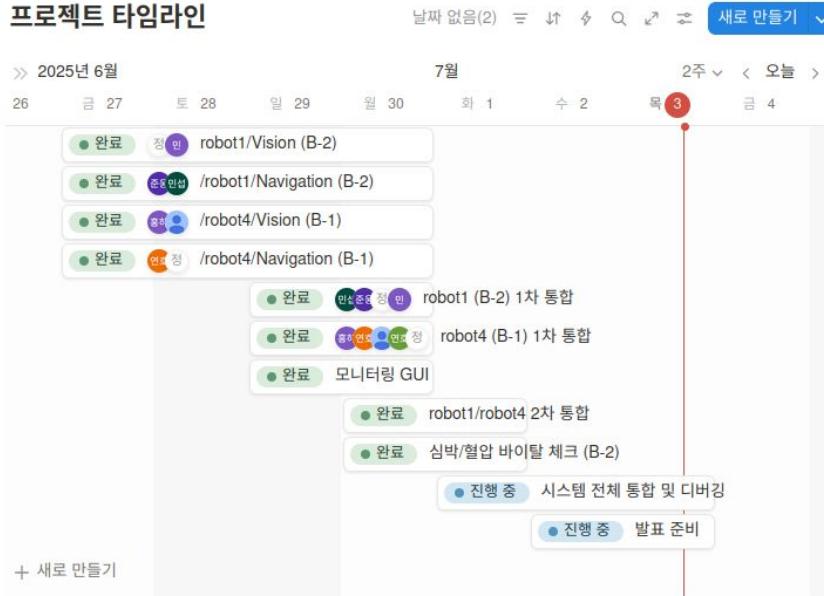
구분	기간	활동		비고
사전 기획 및 기술 검증	6/23(월) ~ 6/26(목)	🚩 프로젝트 기획 및 주제 선정	🚩 기술 검증	아이디어 및 주제 선정
개별 로봇 기능 개발	6/27(금) ~ 6/28(토)	🚩 robot1 노드 개발	🚩 robot4 노드 개발	내비게이션, YOLO 기능 개발
개별 로봇 1차 통합	6/29(일) ~ 6/30(월)	🚩 ROS 서비스 통신 통합	🚩 모니터링 GUI 개발	
전체 통합 및 디버깅	7/1(화) ~ 7/3(목)	🚩 MQTT 메시지 통신 통합	🚩 바이탈 체크 노드 개발	
발표자료 제작	7/3(목) ~ 7/4(금)	🚩 PPT 제작	🚩 문서 작업	
총 개발기간	6/23(월) ~ 7/3(목)(총 11일)			

프로젝트 수행 절차 및 방법

▶ 프로젝트 타임라인 및 역할분담

Project management

프로젝트 타임라인



❸ 프로젝트 팀 구성 및 역할분담 (8명)

역할	담당자	주요 업무
/robot1/Vision (B-2)	최정호 이경민	<ul style="list-style-type: none"> - 메인기능(최정호, 이경민) - 병실 도착 후 사람 YOLO detect 및 얼굴 인식 - 신원 확인 및 얼굴 캡처 - 병실로 이동 중 휠체어와 넘어진사람 YOLO detect - 인식 후 회피기동
/robot1/Navigation (B-2)	문준웅 정민섭	<ul style="list-style-type: none"> - 메인기능(문준웅, 정민섭) - 병실 도킹스테이션에서 출발 - 병실 위치로 내비게이션 - 초기 위치 복귀
/robot4/Vision (B-1)	백홍하 이하빈	<ul style="list-style-type: none"> - 메인기능(백홍하, 이하빈) - 약품 YOLO detect - 약품 위치 추정 및 위치값 서비스 response
/robot4/Navigation (B-1)	장연호 정찬원	<ul style="list-style-type: none"> - 메인기능(장연호, 정찬원) - 제조실 도킹스테이션에서 출발 - 약품 보관 위치, 전달 위치로 내비게이션 - 초기 위치 복귀
심박/혈압 바이탈 체크	정민섭	<ul style="list-style-type: none"> - 서브기능 - 병실에서 생체신호 추정 및 결과 처리 - 심박/혈압 측정, 데이터 전송
모니터링 GUI	백홍하 문준웅	<ul style="list-style-type: none"> - 서브기능 - 병실 목적지 설정 - 약품 전달 요청 인터페이스 개발 - 위치 등 TurtleBot 상태 모니터링 GUI 개발

03 프로젝트 수행 절차 및 방법

K-Digital Training

▶ 기술 스택

개발환경



하드웨어



인터페이스



협업툴



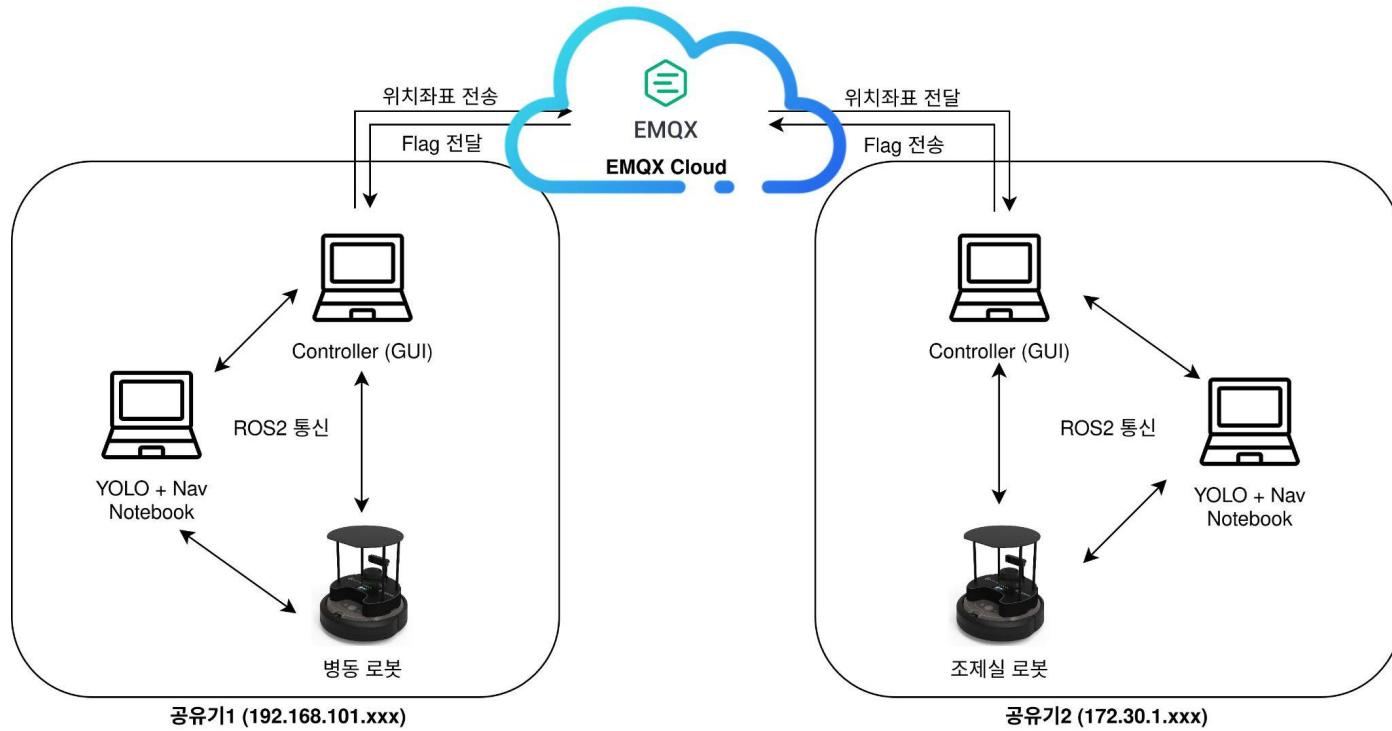
AI



03

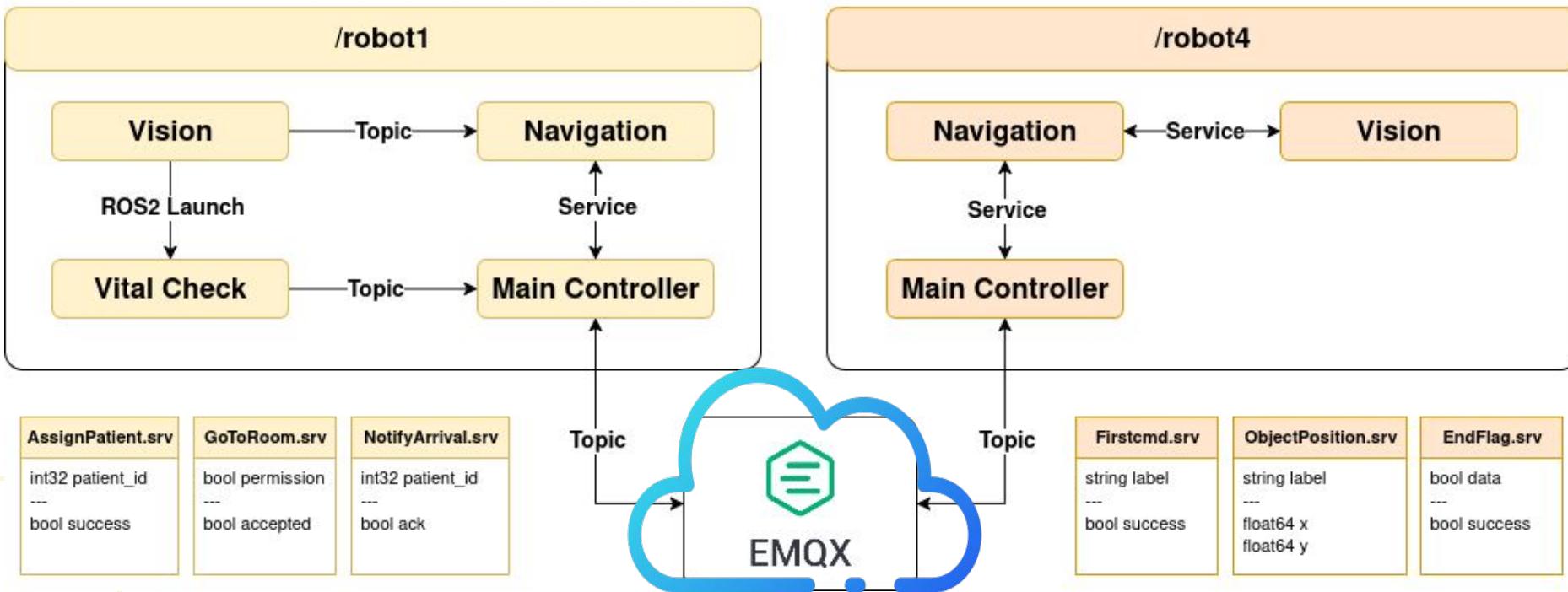
프로젝트 수행 절차 및 방법

▶ 시스템 아키텍처



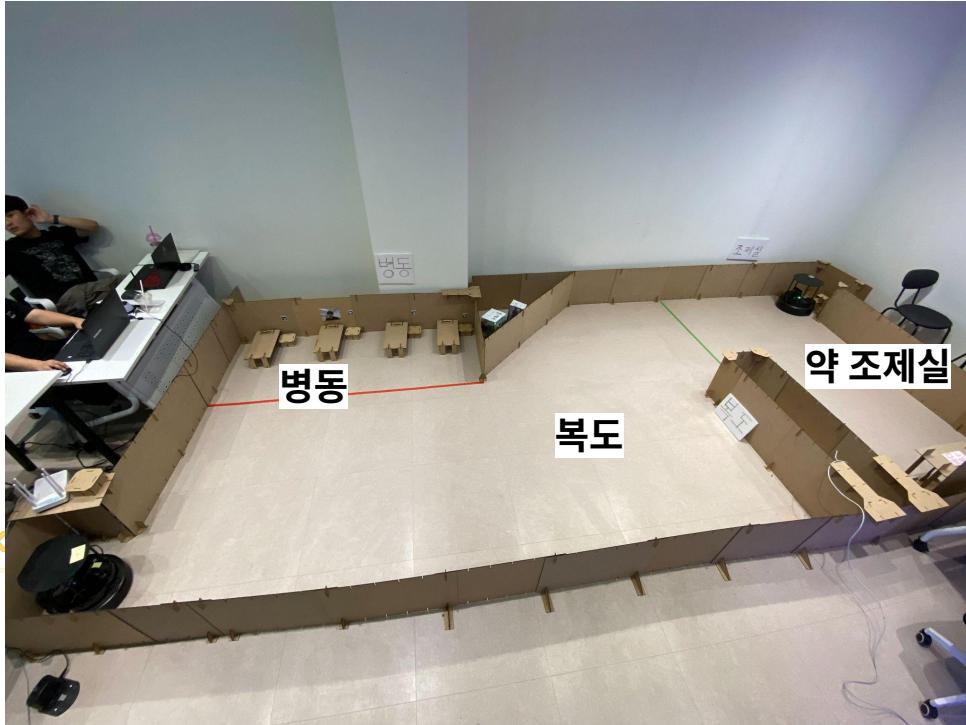
03 프로젝트 수행 절차 및 방법

▶ 노드 아키텍처



03 프로젝트 수행 절차 및 방법

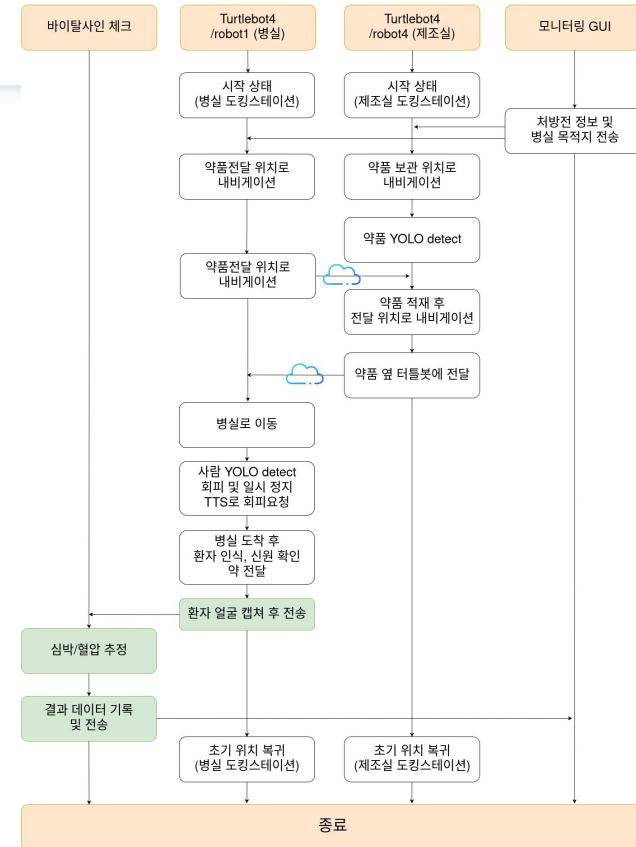
▶ 프로젝트 맵



03

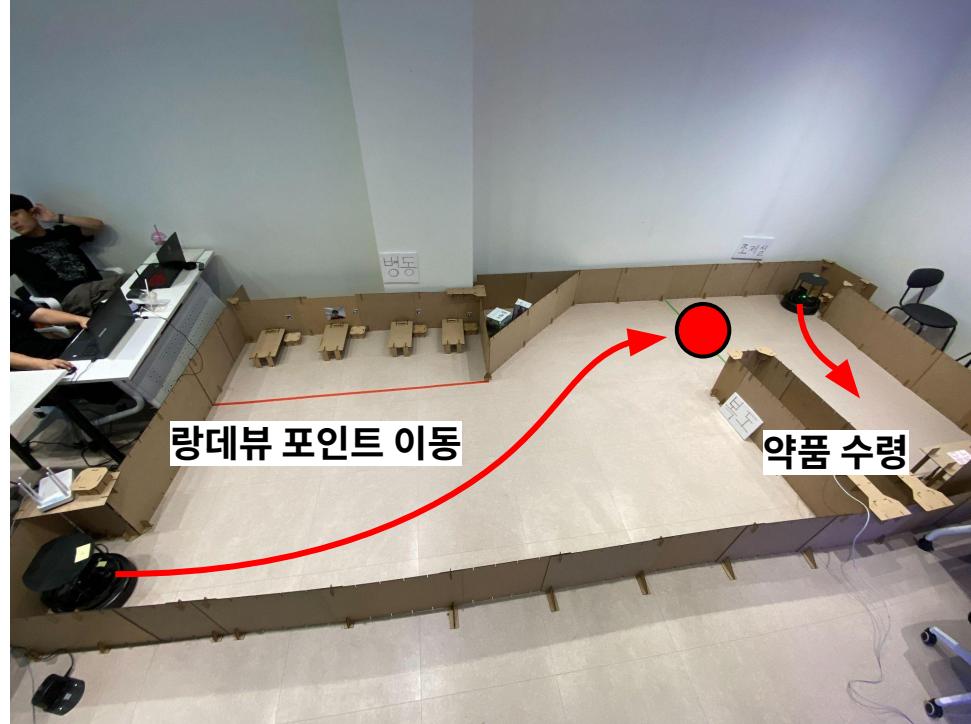
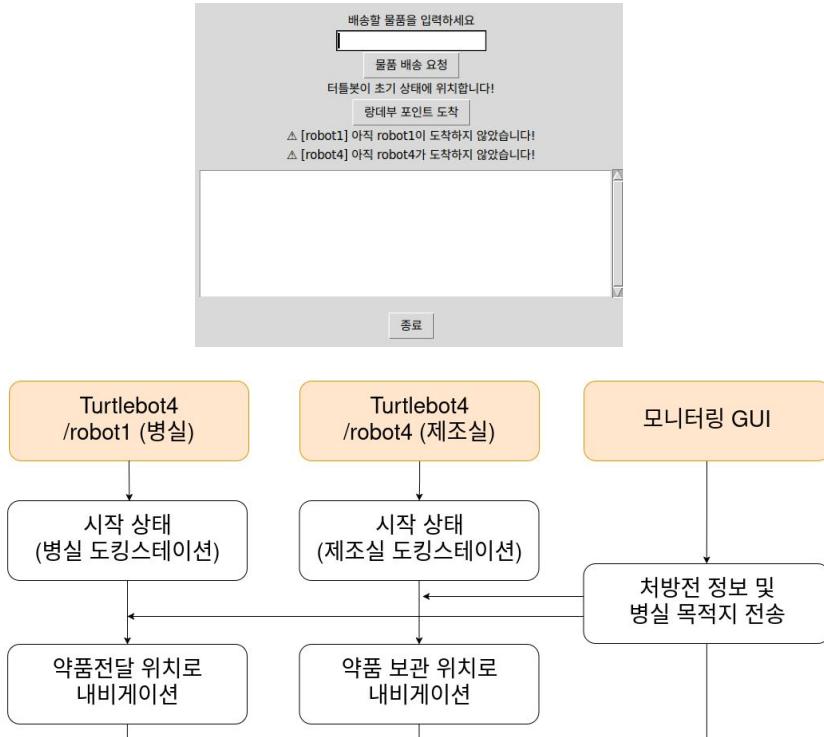
프로젝트 수행 절차 및 방법

▶ 시스템 플로우



03 프로젝트 수행 절차 및 방법

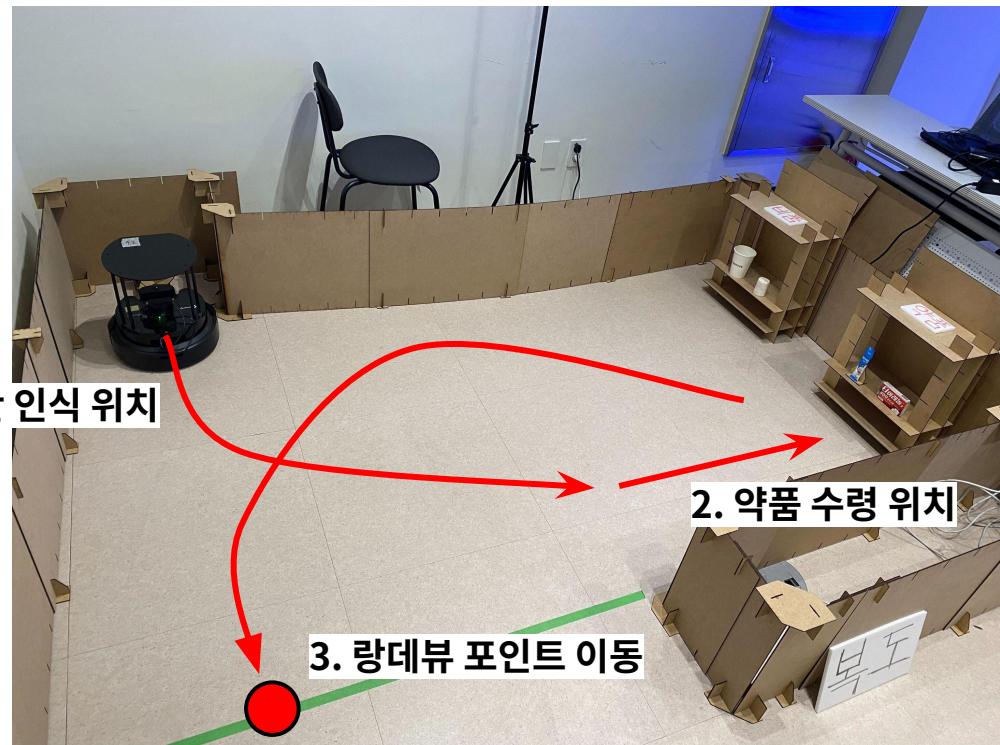
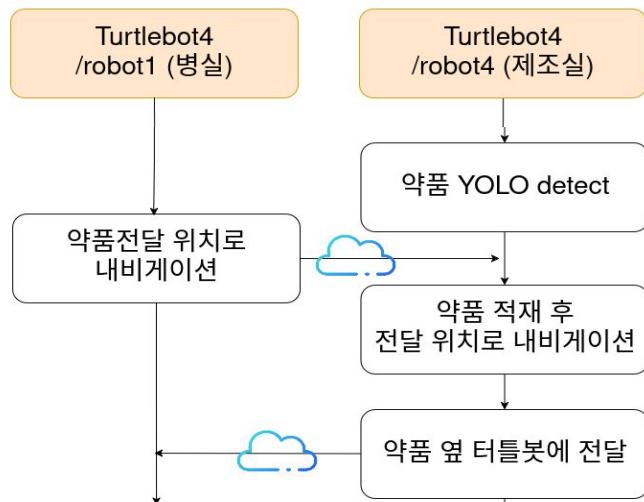
▶ 프로젝트 시나리오



03 프로젝트 수행 절차 및 방법

K-Digital Training

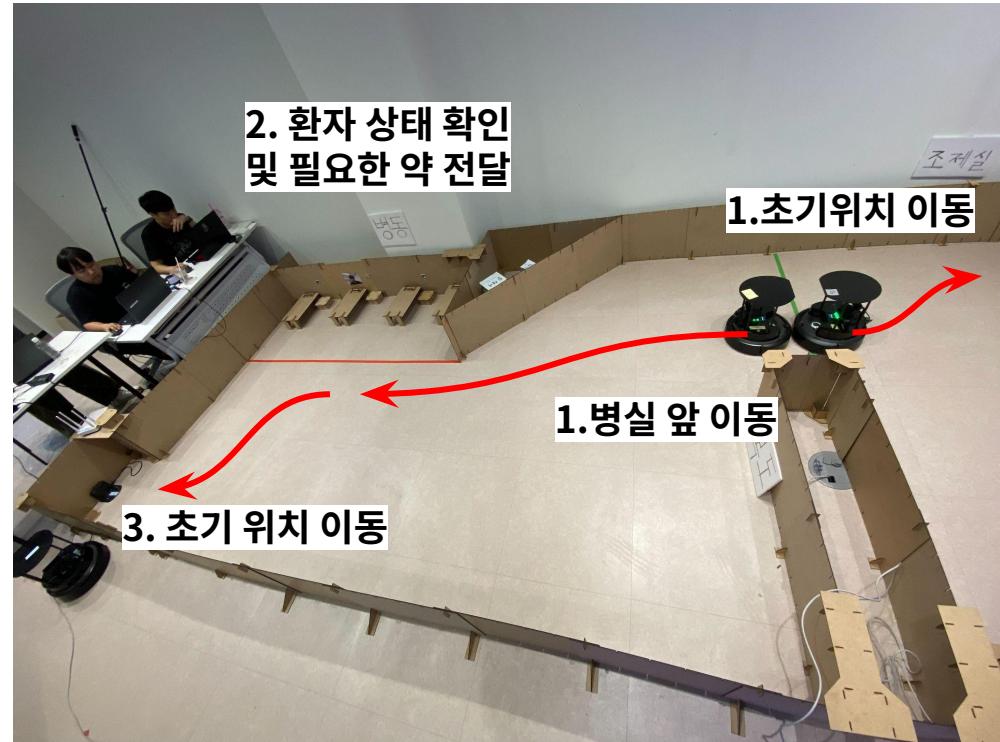
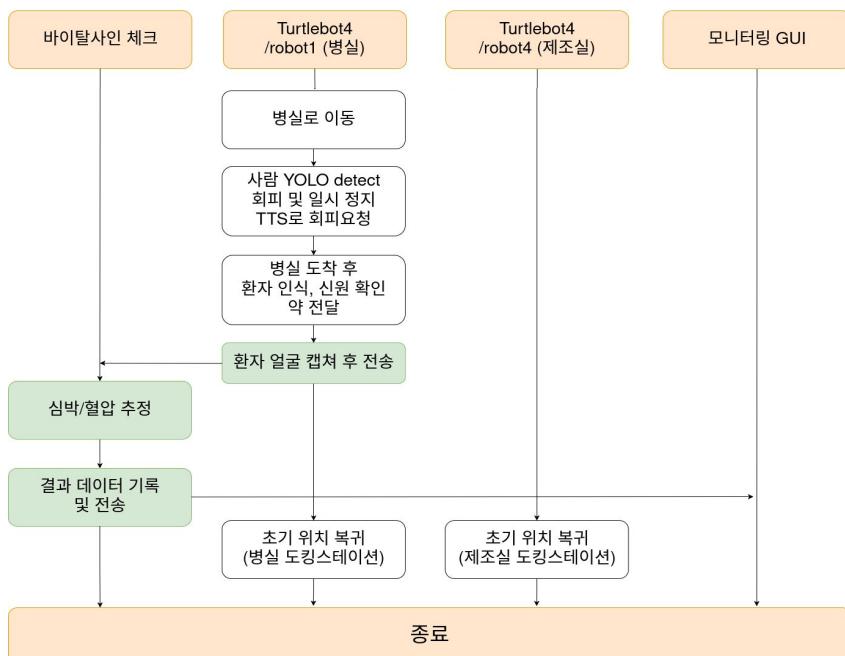
▶ 프로젝트 시나리오



03

프로젝트 수행 절차 및 방법

▶ 프로젝트 시나리오



03 프로젝트 수행 절차 및 방법

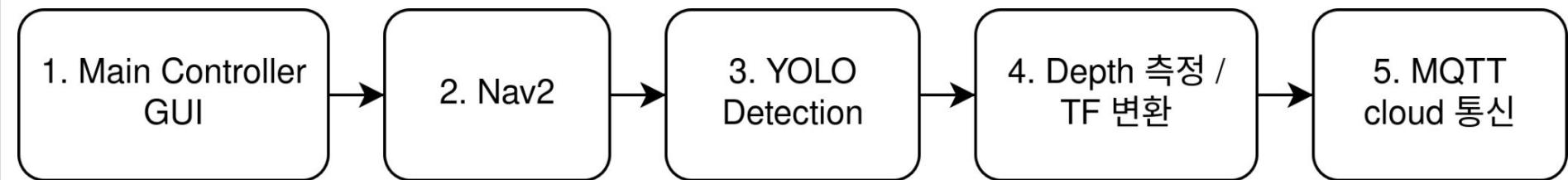
▶ 시연 영상



X

04 프로젝트 수행 경과 - 조제실 로봇

▶ turtle4 - 조제실 로봇



04 프로젝트 수행 경과 - 조제실 로봇

▶ Main Controller GUI

1. 물품 배송 요청

조제실에 있는 약품 입력 시
터틀봇 출발(request)

터틀봇 출발 시
배송 요청을 클릭해도
더이상 request 하지 않음



2. 랑데부 포인트 도착

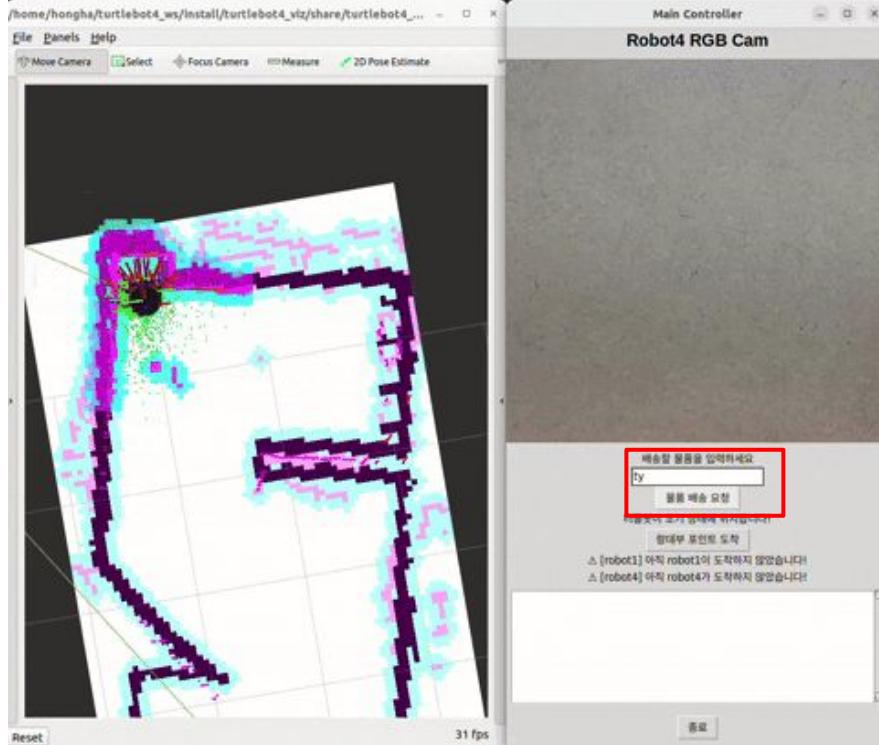
터틀봇 둘 다
랑데부 포인트 도착 시
버튼이 활성화 → 터틀봇 출발

하나라도 도착하지 않을 시
버튼 활성화 되지 않음



04 프로젝트 수행 경과 - 조제실 로봇

▶ Main Controller GUI



2. 랑데부 포인트 도착

robot1, robo4 둘 다
랑데부 포인트 도착 시
버튼이 활성화 → 터틀봇 출발

하나라도 도착하지 않을 시
버튼 활성화 되지 않음



04 프로젝트 수행 경과 - 조제실 로봇

▶ Nav2

요청받은 물품이 있는 구역으로
Nav2를 통해 이동

약품(타이레놀, 콜대원),
물품(붕대, 컵)
선반으로 구분

해당 선반이 보이는 위치로 이동 후,
객체인식 실행



04 프로젝트 수행 경과 - 조제실 로봇

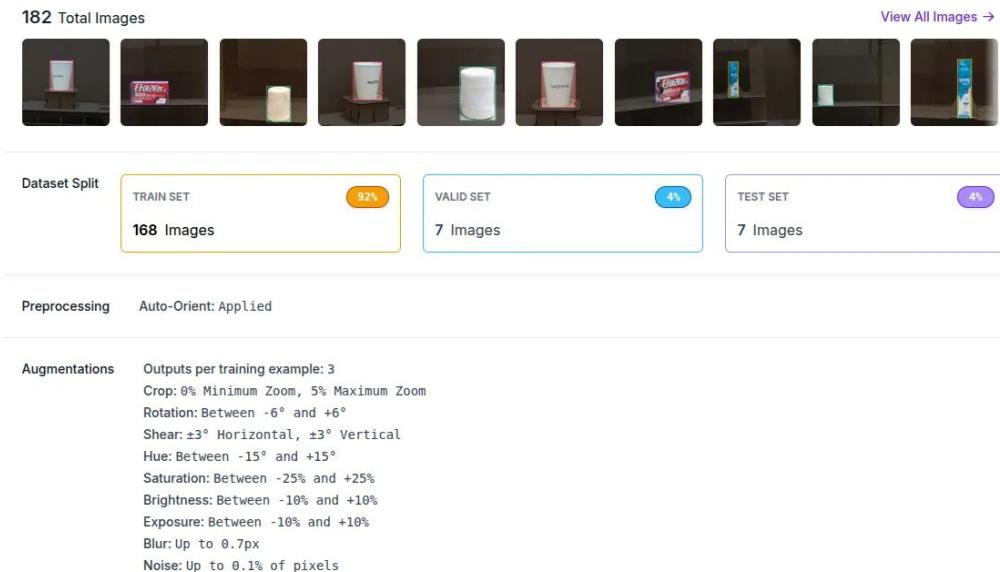
▶ YOLO Detect

YOLO Dataset 제작

/robot4/okad/rgb/image_raw 토픽을 받아
이미지 데이터 70장 촬영
Roboflow에서 라벨링 진행

증강을 활용 총 182장 데이터셋 구성

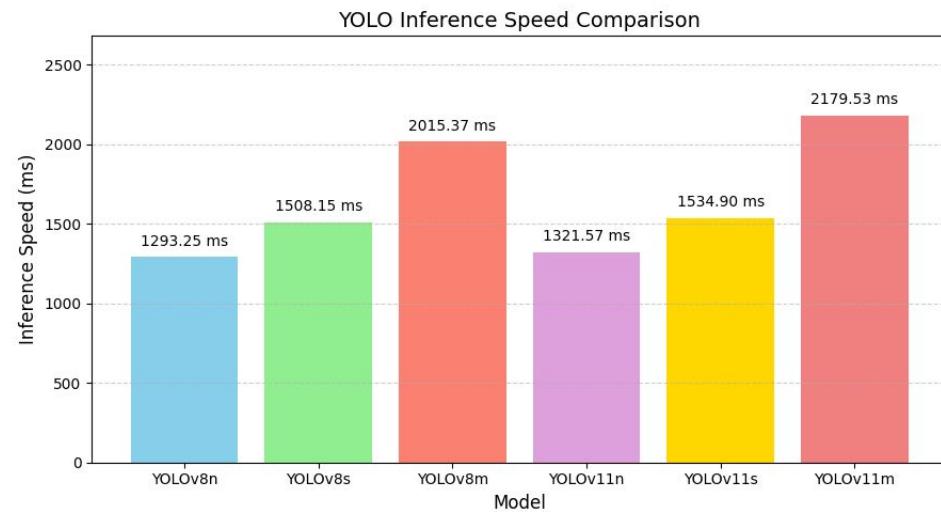
class: 봉대, 컵, 콜대원, 타이레놀



04 프로젝트 수행 경과 - 조제실 로봇

▶ YOLO Detect

Model	신뢰도 평균	오인식	Inference time(70장)
yolov8n	0.952	O	1293.25 ms
yolov8s	0.930	X	1508.15 ms
yolov8m	0.935	O	2015.37 ms
yolov11n	0.957	O	1321.57 ms
yolov11s	0.895	O	1534.90 ms
yolov11m	0.895	X	2179.53 ms



04 프로젝트 수행 경과 - 조제실 로봇

▶ YOLO Detect

1. 빠른 연산 시간
2. 적절한 신뢰도
3. 모델 크기와 속도 균형의 적절한 타협
4. 오인식이 없음

→ 위와 같은 요인을 고려하여

yolov8s 모델 사용 결정



yolov8m



yolov8s

04 프로젝트 수행 경과 - 조제실 로봇

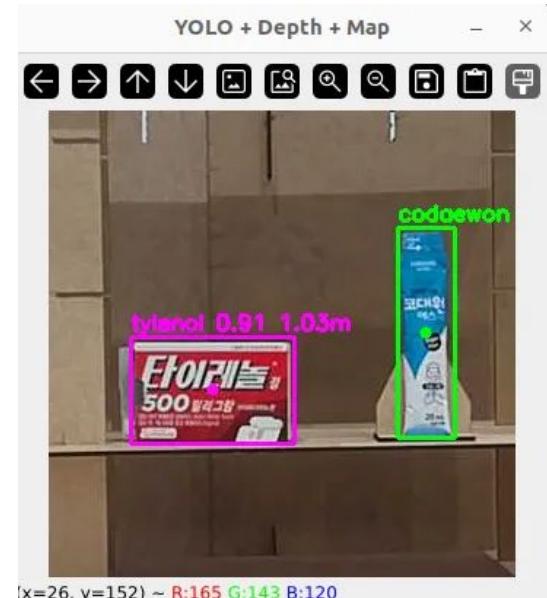
▶ YOLO Detect

물품 탐지

선반 위치로 이동 후
nav → vision request 전송
vision 5초간 객체탐지 수행

depth 거리 계산 수행

이후 해당 객체 map 위치 추정



04 프로젝트 수행 경과 - 조제실 로봇

▶ Depth 측정 / TF 변환

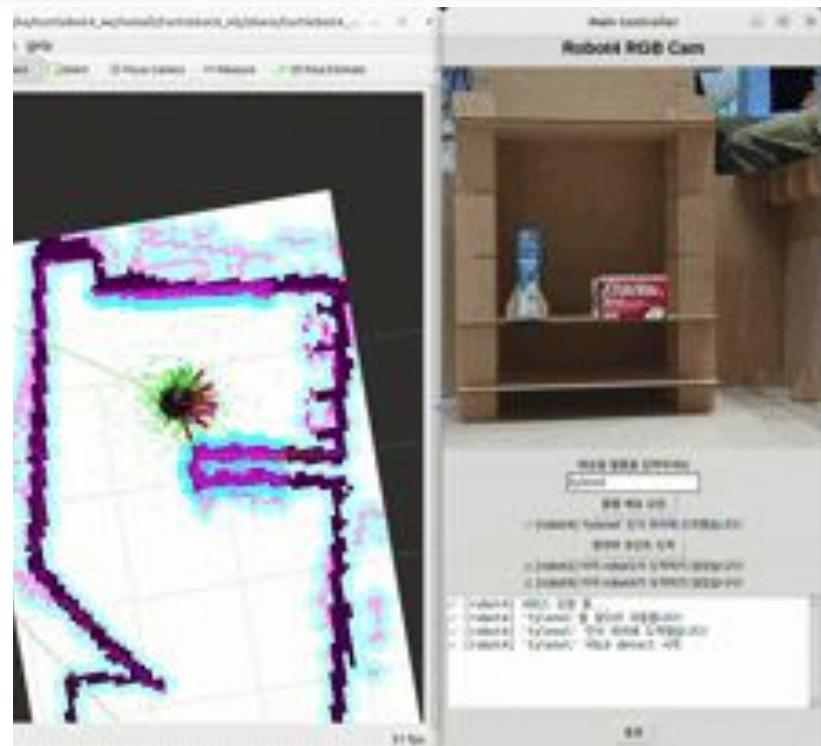
물품 위치 추정

YOLO Bounding Box 중심 좌표의 depth 추출

TF2 변환(camera → map)

map 좌표 (x,y,z) 추정 및 RVIZ 마커 시각화

물품 위치로 nav2를 사용하여 이동



04 프로젝트 수행 경과 - 조제실 로봇

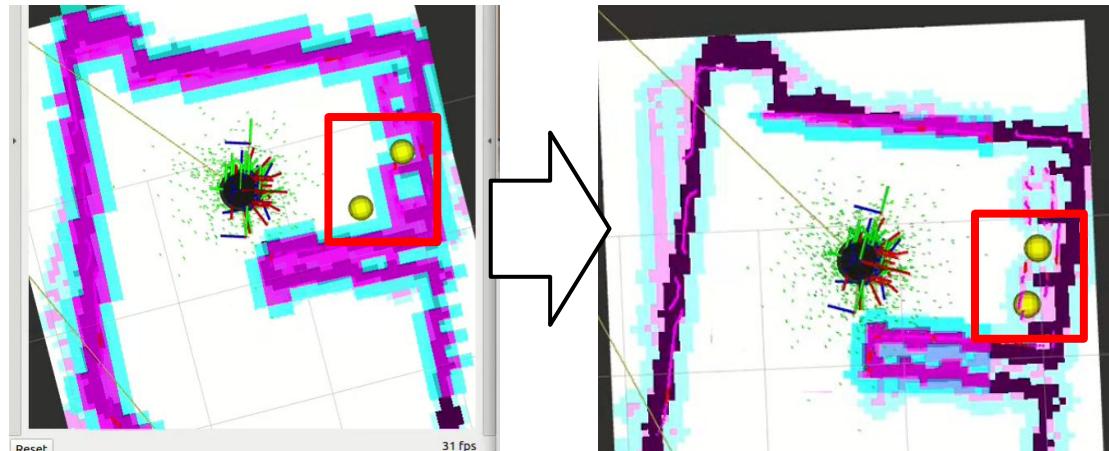
▶ Depth 측정 / TF 변환

카메라 해상도 변경

기존 해상도인 240픽셀에서는
depth 추정의 오차가 커,
물품의 위치를 정확히 추정할 수 없었음

depth값의 정확성을 향상시키고자
카메라의 해상도를 480픽셀로 변경

변경 후 안정적으로 depth값을 계산

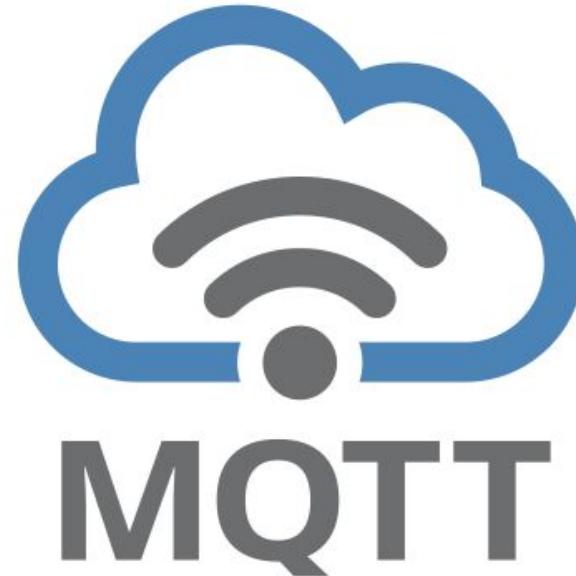


04 프로젝트 수행 경과 - 조제실 로봇

▶ MQTT cloud

두 대의 터틀봇이 단일 공유기 사용 시
트래픽 과부하로 통신 병합에 문제 발생

네트워크를 분리하고 클라우드 기반 통신으로
안정성을 확보하고자 함



04 프로젝트 수행 경과 - 조제실 로봇

▶ MQTT cloud

```
Curr: 844.07 kBit/s  
Avg: 1.19 MBit/s  
Min: 30.71 kBit/s  
Max: 23.13 MBit/s  
.... Ttl: 13.00 MByte  
  
#####  
##### Curr: 94.74 MBit/s  
##### Avg: 39.52 MBit/s  
##### Min: 615.46 kBit/s
```

MQTT Cloud 사용전
(평균 네트워크 사용량 **102MB**)



```
Curr: 259.47 kBit/s  
Avg: 208.81 kBit/s  
Min: 14.25 kBit/s  
Max: 1.01 MBit/s  
Ttl: 23.91 MByte  
  
#####  
##### Curr: 83.52 MBit/s  
##### Avg: 66.19 MBit/s  
##### Min: 315.58 kBit/s
```

MQTT Cloud 사용 후
(평균 네트워크 사용량 **67MB**)

네트워크 대역폭 분리로 쾌적한 실행환경 확보
리소스 ↓ 안정성 ↑

04 프로젝트 수행 경과 - 조제실 로봇

▶ MQTT cloud 아키텍처

-TurtleBot1 (병동 로봇)

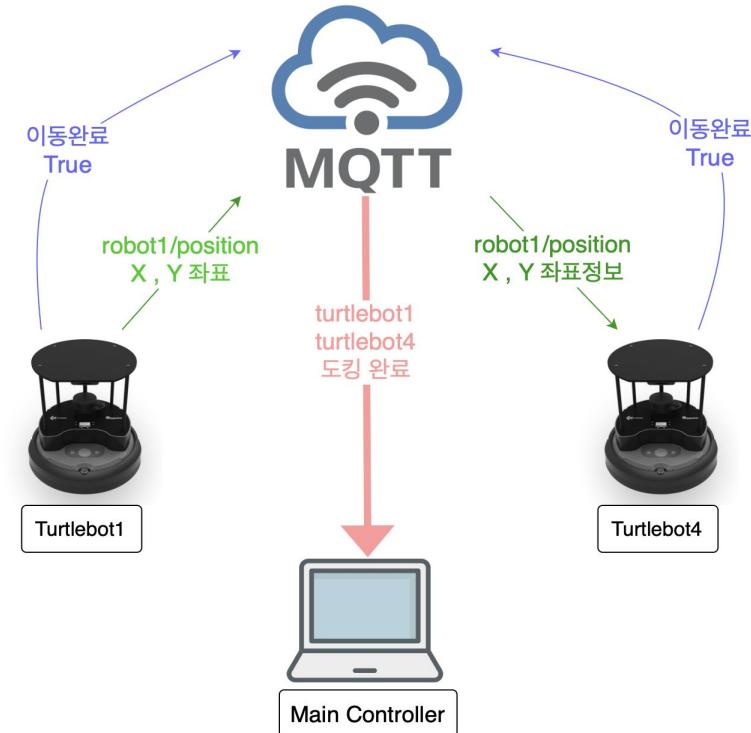
랑데부 포인트 이동 → 좌표 클라우드 Publish

-TurtleBot4 (조제실 로봇)

좌표 Subscribe → 이동 → 이동 완료 신호 Publish

-Main Controller

양쪽 로봇 상태 수신 후 다음 단계 진행



04 프로젝트 수행 경과 - 조제실 로봇

▶ MQTT cloud

```
publisher.py
/home/rokey/rokey_ws/.src/rokey_pjt/rokey_pjt/cloud_publisher.py:22: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
  client = mqtt_client.Client(client_id=client_id, protocol=mqtt_client.MQTTv311)
>
Sent `4.0459012,0.8092611` to topic `robot1/position`
```

turtlebot1

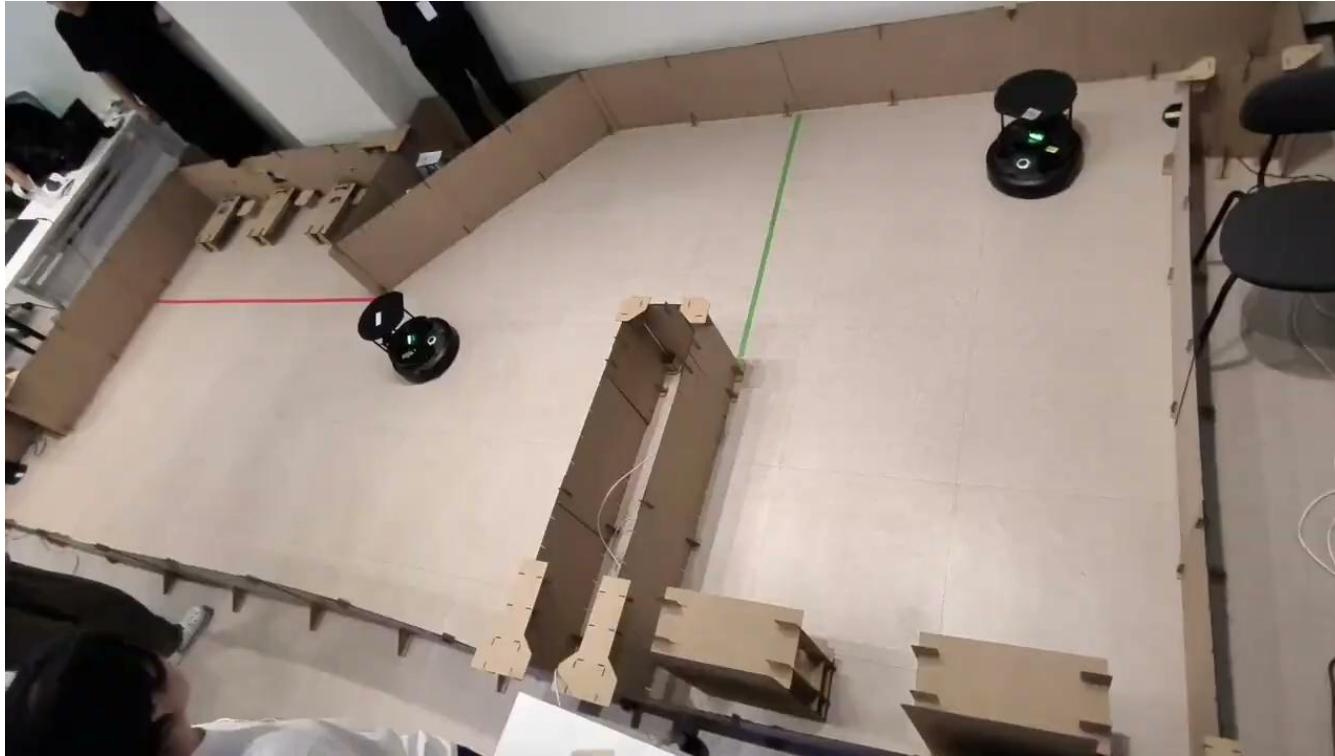
turtlebot1 의 좌표값을
클라우드에서 turtlebot4 가 받아옴

```
/home/rokey/rokey_ws/.src/rokey_pjt/rokey_pjt/cloud_subscriber.py:26: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
  client = mqtt_client.Client(client_id=client_id, protocol=mqtt_client.MQTTv311)
>
Connected to MQTT Broker!
Received `4.0459012,0.8092611` from `robot1/position` topic
```

turtlebot4

04 프로젝트 수행 경과 - Turtle4

▶ MQTT cloud



04 프로젝트 수행 경과 - 조제실 로봇

▶ 물품 전달 위치 이동

Turtle1 옆 위치로 이동

클라우드 서버를 거쳐 **turtle1의 좌표 subscribe**

nav2를 사용하여 수신한 **turtle1의 위치로 이동**

충돌 방지하기 위해 **좌표에 offset**

이동 완료 후, **물품 전달 수행**



04 프로젝트 수행 경과 - 조제실 로봇

▶ 물품 전달 위치 이동

동적 위치 이동

하드코딩된 좌표로 이동 X

subscribe한 **turtle1**의 위치로 이동

turtle1의 위치가 달라져도
해당 위치로 유기적으로 이동



초록선 멀리에 위치할 때



초록선 가까이에 위치할 때

04 프로젝트 수행 경과 - 조제실 로봇

▶ 랑데부 포인트 도착 Flag 전송

물품 전달 완료 Flag

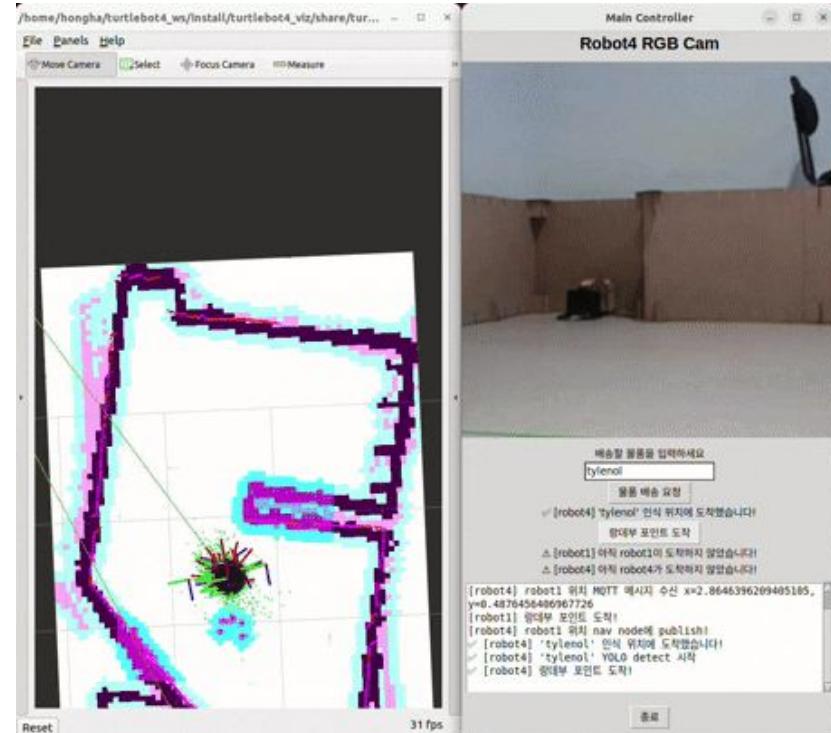
랑데부 포인트로 이동 완료 후

Cloud 통해 도착 여부 turtle1에 publish

랑데부 포인트 도착 버튼 클릭 시 다음 동작 수행

turtle1: 병동으로 이동

turtle4: 초기 Docking 상태로 이동



04 프로젝트 수행 경과 - 조제실 로봇

- ▶ 랑데부 포인트 도착 Flag 전송

물품 전달 완료 Flag

랑데부 포인트로 이동 완료 후

Cloud 통해 도착 여부 turtle1에 publish

랑데부 포인트 도착 버튼 클릭 시 다음 동작 수행

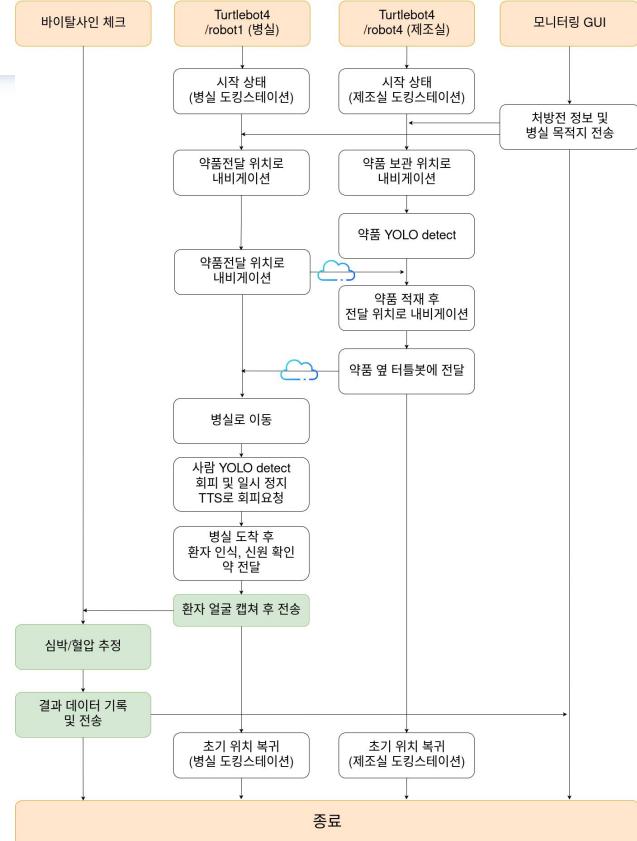
turtle1: 병동으로 이동

turtle4: 초기 Docking 상태로 이동



04 프로젝트 수행 경과 - 병동 로봇

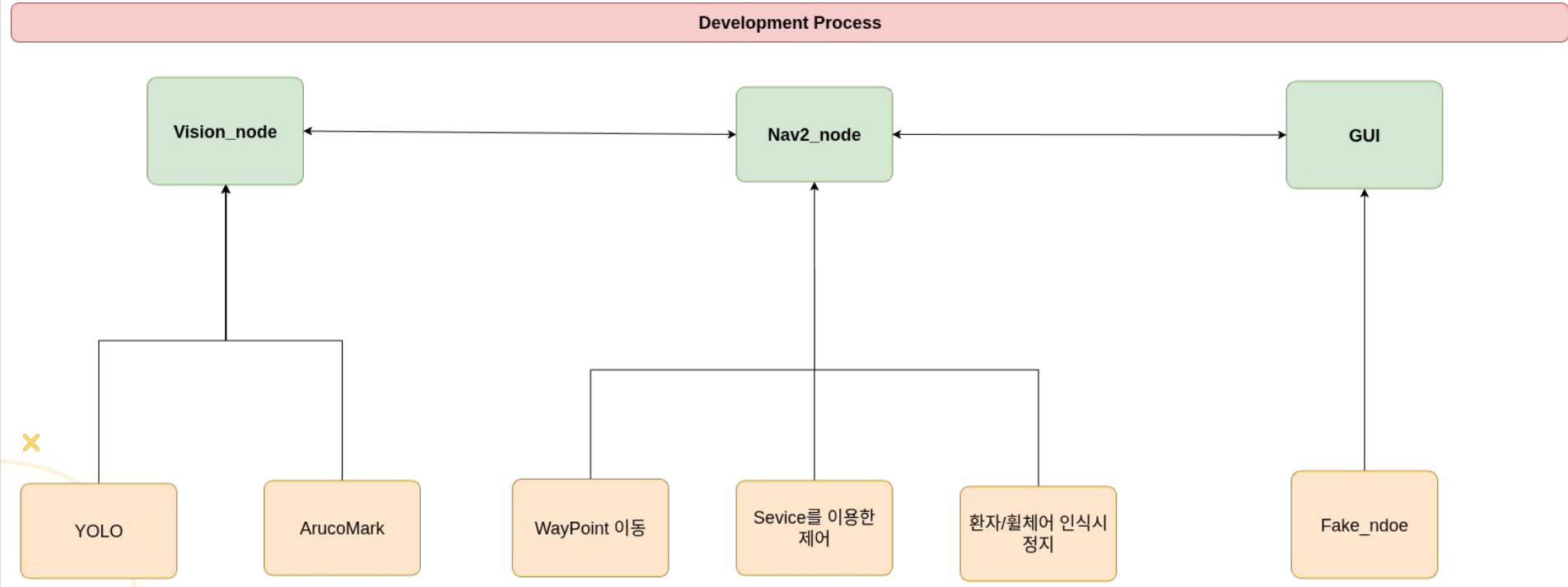
▶ 시스템 플로우



04

프로젝트 수행 경과 - 병동 로봇

▶ 개발 프로세스



04

프로젝트 수행 경과 - 병동 로봇

▶ Nav2 node 단위 테스트

- NAV 주행중 일시정지 주행 TEST

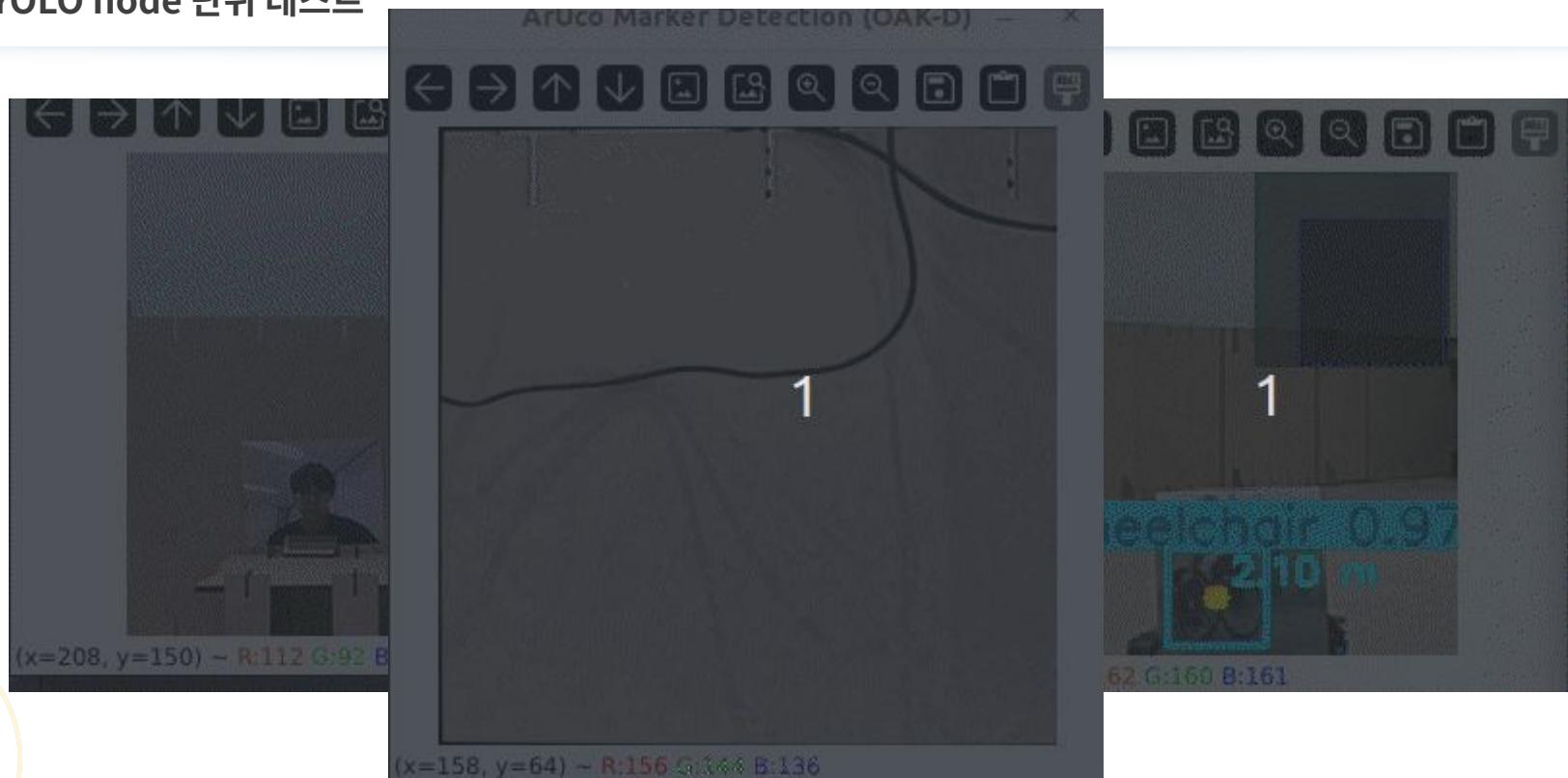
/person_detected (정지 토픽)

/person_cleared (재개 토픽)



04 프로젝트 수행 경과 - 병동 로봇

▶ YOLO node 단위 테스트



04 프로젝트 수행 경과 - 병동 로봇

▶ GUI 단위 테스트

Navigation 없이 GUI와 ROBOT

상태 전이 Service Test

The screenshot shows a software interface for a nursing assistance robot. At the top, there are three tabs: '환자 ID' (Patient ID), '환자 배정 (AssignPatient)', and '병실 이동 허가 (GoToRoom)'. The '환자 배정 (AssignPatient)' tab is currently selected. Below it is a table titled '환자 목록' (Patient List) with columns '환자 번호' (Patient Number), '이름' (Name), and '전화번호' (Phone Number). The table contains five entries:

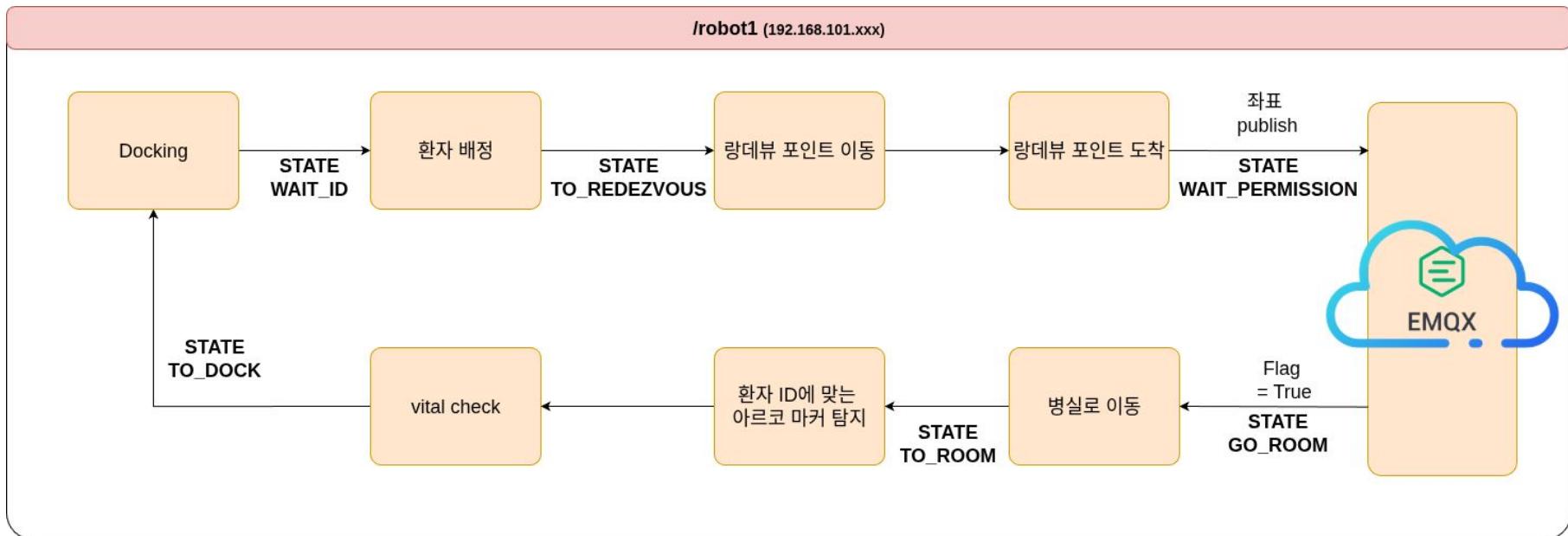
환자 번호	이름	전화번호
1	정민섭	010-1234-5678
2	이영희	010-2345-6666
3	최정호	010-0000-1111
4	문준운	010-2222-1111
5	이경민	010-3333-4444

Below the table is a section labeled '현재 로봇 위치' (Current Robot Position) with a dropdown menu showing '대기중..'. At the bottom left, there is a 'Robot Battery' indicator showing '100%'. On the right side of the screen, there is a terminal window displaying a command-line session:

```
bash: /home/moon/nursing_assistance/install/setup.bash: 그런 파일이나 디렉터리  
가 없습니다.  
moon@moon-ThinkPad-T460: ~$ ros2 run nursing_assistance_robot nav_node3 --ros2  
args -r _ns:=/robot1  
[WARN] [1751586508.42228643] [rcl.logging_rosout]: Publisher already registered  
for provided node name. If this is due to multiple nodes with the same name the  
in all logs for that logger name will go out over the existing publisher. As soon  
as any node with that name is destructed it will unregister the publisher, pre-  
venting any further logs for that name from being published on the rosout topic.
```

04 프로젝트 수행 경과 - 병동 로봇

▶ TurtleBot 1 State Flowchart

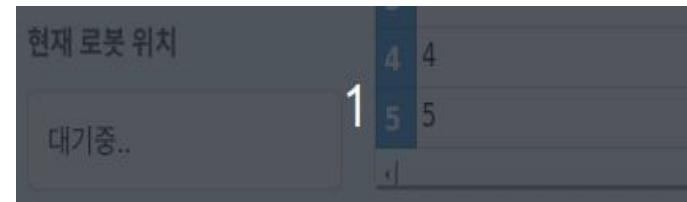


04 프로젝트 수행 경과 - 병동 로봇

▶ 로봇 위치 - GUI

- 현재 로봇 위치 알림

상태에 따라 ROBOT이 GUI에게 위치 송신



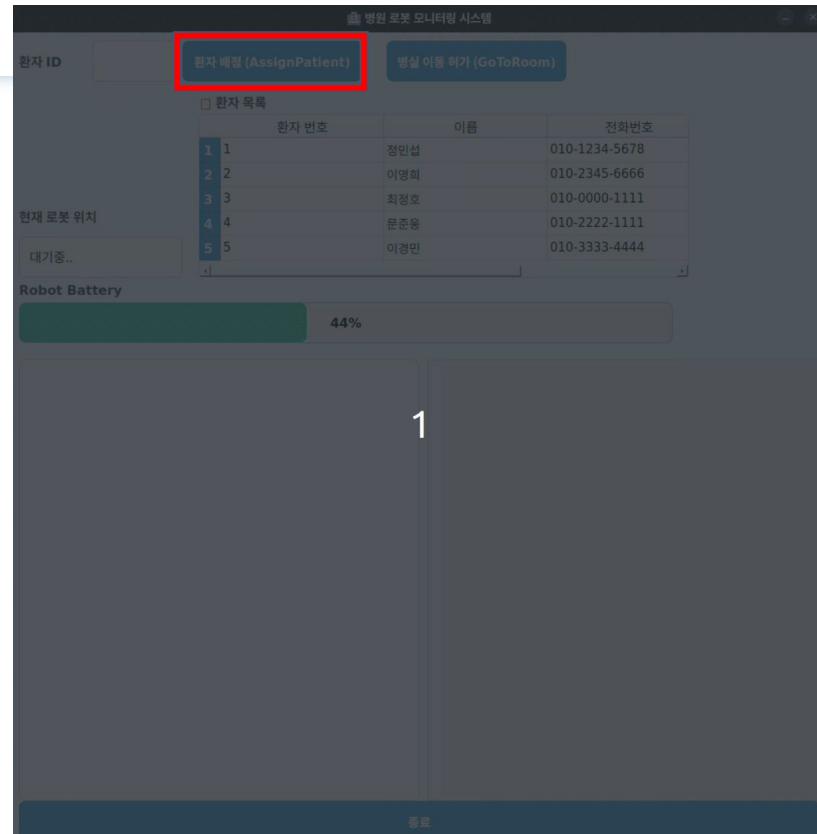
04 프로젝트 수행 경과 - 병동 로봇

▶ 환자 ID 배정 - GUI

1. 환자 ID 배정

GUI에서 테이블 환자 데이터 선택 또는 직접 입력

환자 배정 성공시 log 출력



04 프로젝트 수행 경과 - 병동 로봇

- ▶ 랑데부 포인트까지 이동 - Nav2_node

2. 랑데부 포인트로 이동

환자 ID 배정시 지정된 랑데부 포인트로 이동

이동 완료시 flag 수신 대기



04 프로젝트 수행 경과 - 병동 로봇

▶ 랑데부 포인트까지 이동 - 환자 감지시

2-1. 랑데부 포인트로 이동

이동 중 사람 감지시 STOP



```
rokey@rokey-550XBE-350XBE: ~/rokey_ws
rokey@rokey-550XBE-350XBE: ~/rokey_ws $ rokey2
anything else : stop
q/z : increase/decrease max speeds by 10%
w/x : increase/decrease only linear speed by 10%
e/c : increase/decrease only angular speed by 10%
CTRL-C to quit
currently: speed 0.5 turn 0.
7549729318046625
currently: speed 0.45 turn 0.
7549729318046625
currently: speed 0.485 turn 0.
7549729318046625
currently: speed 0.3645000000000000
05 turn 0.6794756386241962
currently: speed 0.4009500000000000
1 turn 0.747423202486016
currently: speed 0.3688550000000000
1 turn 0.6726000022379544
```

04 프로젝트 수행 경과 - 병동 로봇

- ▶ YOLO Dataset - 사용한 클래스



04 프로젝트 수행 경과 - 병동 로봇

▶ YOLO Dataset

데이터 라벨링

이미지 데이터 총 92장을 촬영하여
Roboflow에서 라벨링 진행

증강을 활용하여 총 276장의 사진으로
데이터셋 구성

Outputs per training example: 3
 Crop: 0% Minimum Zoom, 5% Maximum Zoom
 Rotation: Between -6° and $+6^{\circ}$
 Shear: $\pm 3^{\circ}$ Horizontal, $\pm 3^{\circ}$ Vertical
 Hue: Between -15° and $+15^{\circ}$
 Saturation: Between -25% and +25%
 Brightness: Between -10% and +10%
 Exposure: Between -10% and +10%
 Blur: Up to 0.7px
 Noise: Up to 0.1% of pixels

[View All Images →](#)

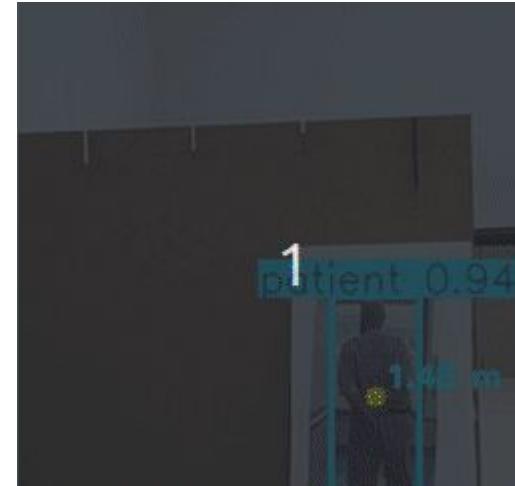


04 프로젝트 수행 경과 - 병동 로봇

▶ YOLO Dataset

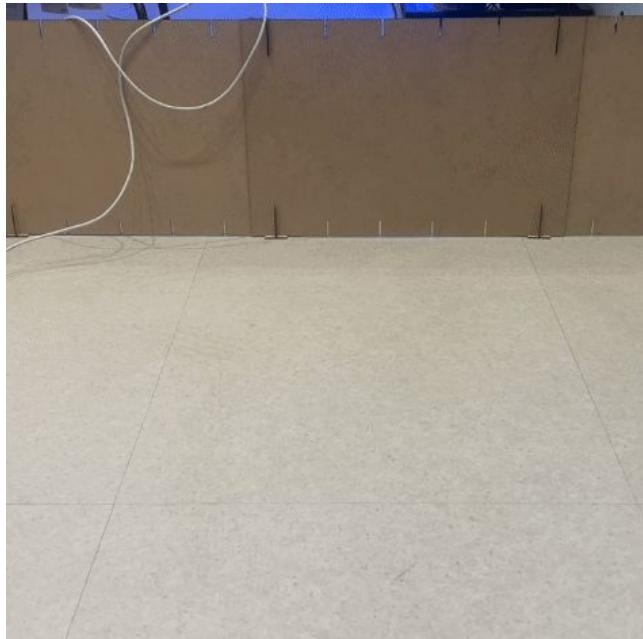
모델 학습

epochs: 100, imgsz: 640, batch: 16



04 프로젝트 수행 경과 - 병동 로봇

- ▶ 보는 사람의 가시성을 높인 물체들



04 프로젝트 수행 경과 - 병동 로봇

▶ OAK-D 카메라 설정 파일

```
/oakd:  
ros__parameters:  
camera:  
i_enable_imu: false # IMU 데이터 사용 (필요시)  
i_enable_ir: false # IR 프로젝터 사용 (필요시)  
i_floodlight_brightness: 0 # IR 투광기 밝기  
i_laser_dot_brightness: 100 # IR 도트 프로젝터 밝기  
i_nn_type: none  
i_pipeline_type: RGBD # 파이프라인: RGB+D (Depth 활성화)  
i_usb_speed: SUPER_PLUS  
rgb:  
i_board_socket_id: 0  
i_fps: 10.0  
i_height: 320  
i_interleaved: false  
i_max_q_size: 1  
i_preview_size: 250  
i_enable_preview: true  
i_low_bandwidth: false  
i_keep_preview_asct_ratio: true  
i_publish_topic: true #전체 해상도 영상 비활성 (프리뷰만 퍼블리시)  
i_resolution: '480P'  
i_width: 320  
stereo:  
i_publish_topic: true  
i_align_depth: true  
use_sim_time: false
```

GPU가 아닌 노트북에서 비전 프로젝트 진행



카메라와 yolo, depth를 같이 실행



데이터 처리가 늦어져 멈추거나 과거의 데이터를 가져오는 상황 발생

04 프로젝트 수행 경과 - 병동 로봇

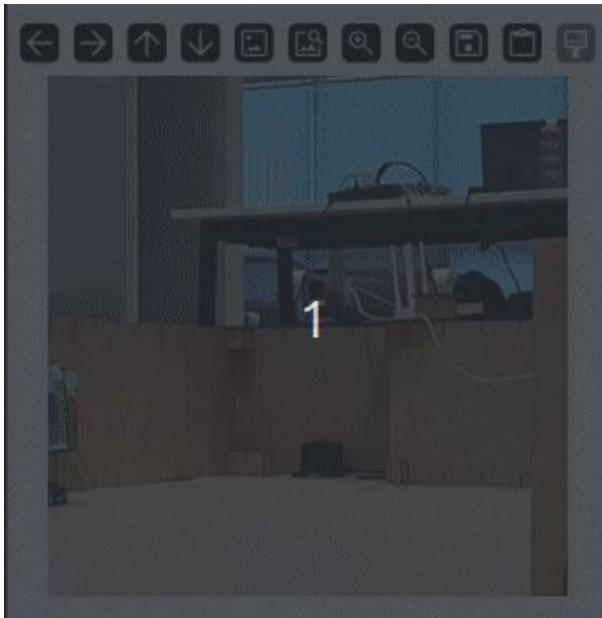
▶ OAK-D 카메라 설정 파일

CPU 노트북에서 실행하기 위해 변경한 값

파라미터	기능 설명	값	목적
i_low_bandwidth	압축된 프리뷰 활성화. 고화질 이미지 전송	False	대역폭 줄이기 위한 최적화
i_max_q_size	ROS 메시지 큐 크기(10이면 항상 최신 프레임만 유지)	1	실시간성 증가
i_fps	초당 프레임 수(너무 높으면 부하 증가)	15	저부하, 실시간 추론 용도
i_width, i_height	출력 프리뷰 영상의 해상도 (정사각형)	320x320	추론 모델 입력에 맞춰 인식률 증가

04 프로젝트 수행 경과 - 병동 로봇

▶ FPS 변경



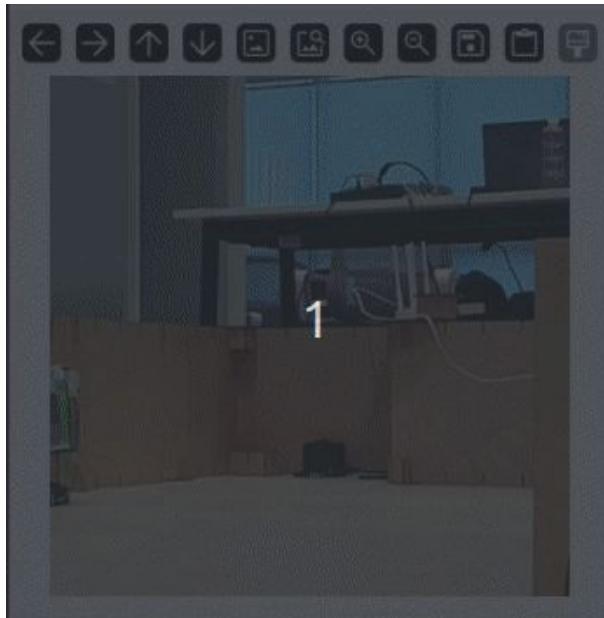
YOLO와 데프스 동시 실행 (15 FPS)

아직도 부족한 실시간성을 해결하고자 비교

FPS 설정	카메라만 실행	YOLO만 실행 (RGB)	YOLO와 데프스 동시 실행 (RGB + Depth)
30 FPS	지연 발생	카메라 멈춤	실행 불가 (X)
20 FPS	일반적인 영상	끊김 발생	5초 후 원본 영상 나옴
15 FPS	일반적인 영상	자주 끊김	4초 후 원본 영상 나옴
10 FPS	일반적인 영상	일반적인 영상	한번 끊긴 후 정상 작동

04 프로젝트 수행 경과 - 병동 로봇

▶ FPS 변경



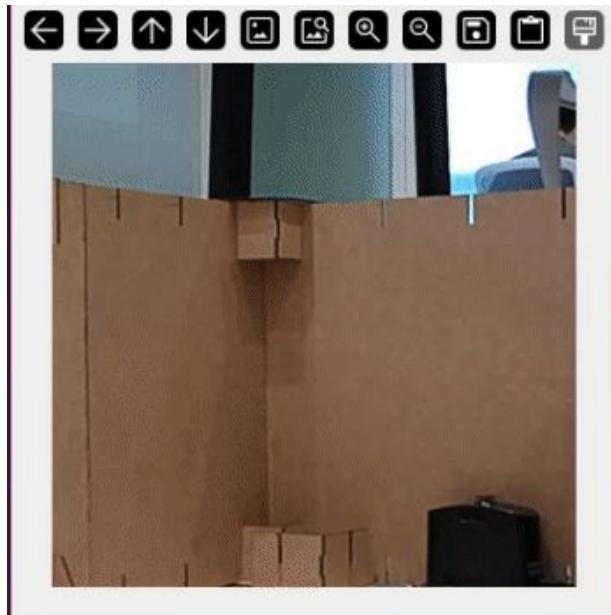
YOLO와 데프스 동시 실행 (15 FPS)

아직도 부족한 실시간성을 해결하고자 비교

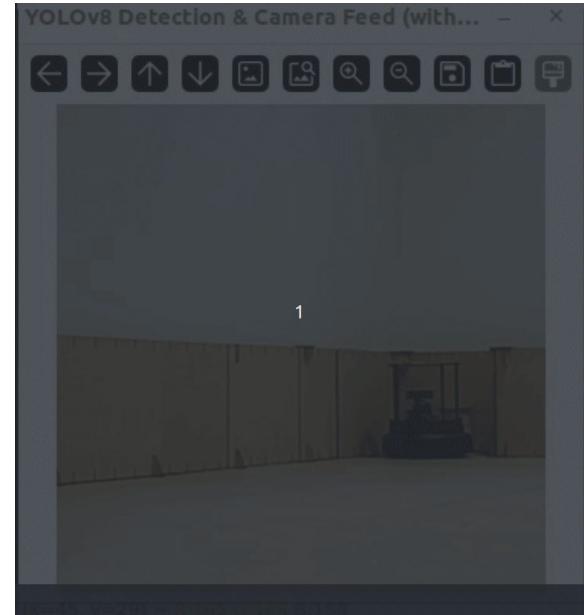
FPS 설정	카메라만 실행	YOLO만 실행 (RGB)	YOLO와 데프스 동시 실행 (RGB + Depth)
10 FPS	일반적인 영상	미세한 지연	1초 정도 프레임 지연
5 FPS	일반적인 영상	미세한 지연	1초 정도 프레임 지연
3 FPS	일반적인 영상	엄청 미세한 지연	0.5초 정도 프레임 지연
2 FPS	일반적인 영상	엄청 미세한 지연	미세한 지연

04 프로젝트 수행 경과 - 병동 로봇

▶ 세팅 변경 적용 변화



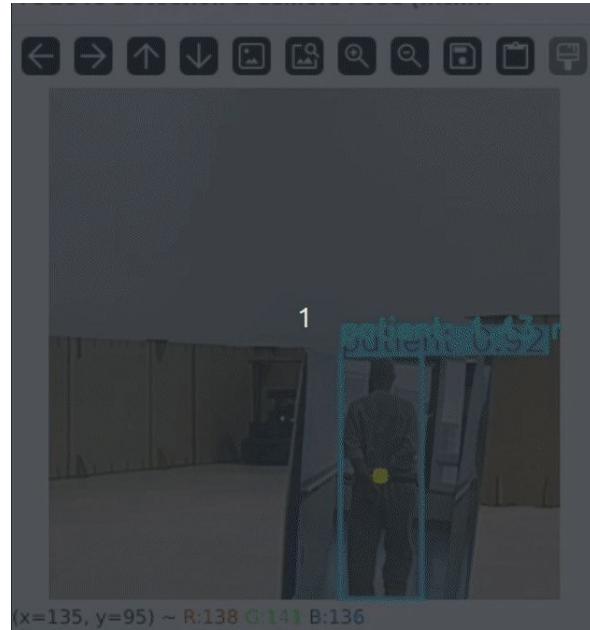
YOLO와 데ps 동시 실행 (5 FPS)



YOLO와 데ps 동시 실행 (2 FPS)

04 프로젝트 수행 경과 - 병동 로봇

▶ 최종 카메라 화질

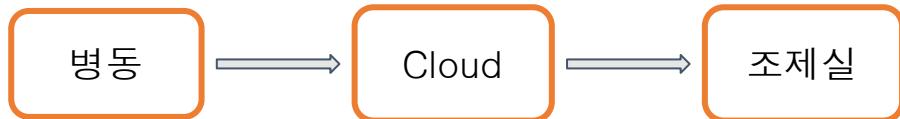


04 프로젝트 수행 경과 - 병동 로봇

▶ 도착 후 현재 좌표 publish

3. 도착 후 현재 좌표 publish

도착 완료시 현재 좌표를 Cloud에 publish



```
moon@moon-Thin-GF63-12UCX: ~ 109x43
moon@moon-Thin-GF63-12UCX: $ ros2 run nursing_assistance_robot cloud --ros-args -r __ns:=/robot1
[INFO] [1751586820.735635774] [robot1.cloud_node]: ✓ MQTT 브로커 연결 성공
```

04 프로젝트 수행 경과 - 병동 로봇

▶ 약 전달 flag subscribe - GUI

4. 약 전달 flag subscribe

조제실 GUI에서 publish한 출발 허가 flag
수신시, 병동 앞으로 출발



```
moon@moon-Thin-CF63-12UCX: ~ 109x43
moon@moon-Thin-CF63-12UCX: $ ros2 run nursing_assistance_robot cloud_sub --ros-args -r __ns:=/robot1
[INFO] [1751587140.623353054] [robot1.mqtt_receiver_node]: MQTT 브로커 연결 성공
```

04 프로젝트 수행 경과 - 병동 로봇

- ▶ 병동 이동 - Nav2_node.py

4. 병동으로 이동



04 프로젝트 수행 경과 - 병동 로봇

- ▶ 아르코 마커 탐지 - Nav2_node.py

5. 환자 ID 탐지

병동 앞 도착시 회전하며 환자 ID에
해당하는 아르코 마커 탐지, 아르코 마커의
거리가 1m 이상일시 정지



04 프로젝트 수행 경과 - 병동 로봇

▶ 아르코 마커



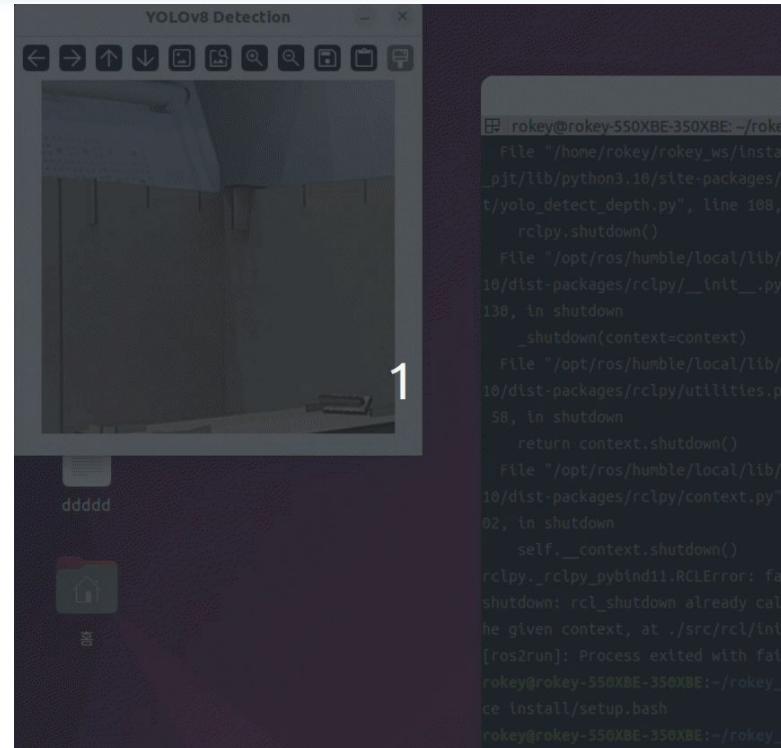
04 프로젝트 수행 경과 - 병동 로봇

▶ face detect - vision_detect_node

6. 환자 face 탐지

아르코 마커의 거리가 1m이내일때,
face detect 시작

detect된 face의 confidence가
0.7 이상일때 vital_check_node 실행

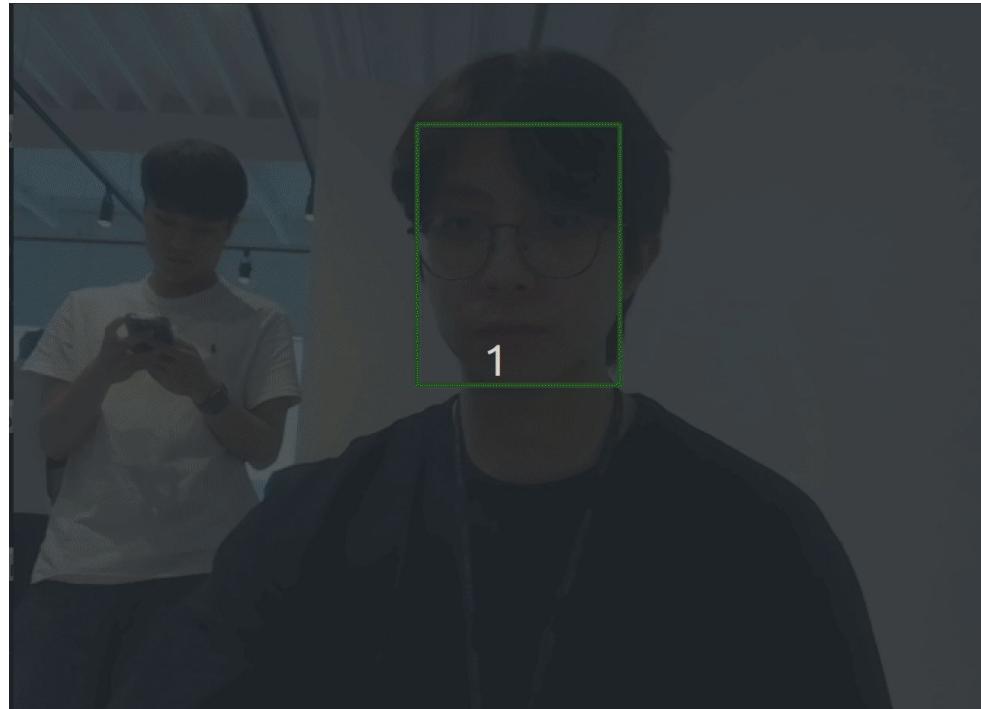


04 프로젝트 수행 경과 - 병동 로봇

▶ Vital Check - vital check node

7. Vital Check

환자의 얼굴을 감지하여 rPPG 추출 및
전처리를 통해 심박수, 산소포화도, 혈압 추정



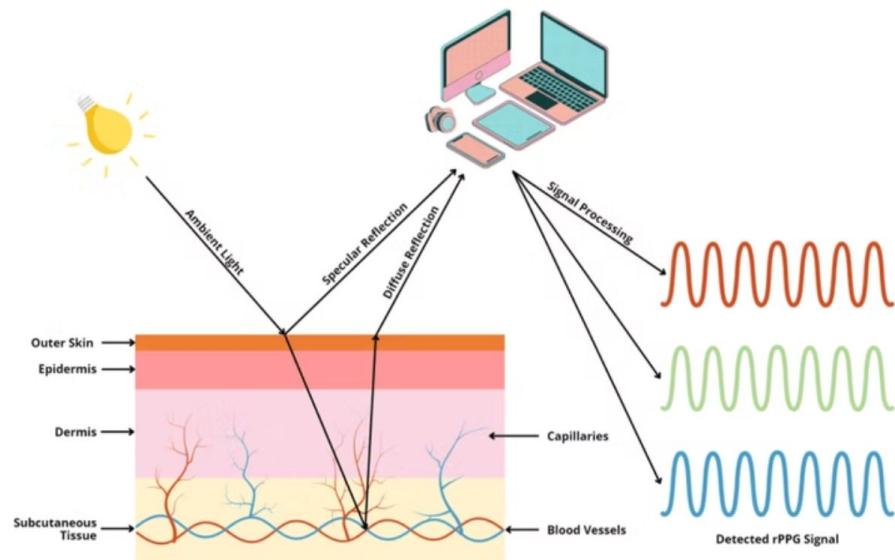
04 프로젝트 수행 경과 - 병동 로봇

▶ Vital Check - vital_check_node

7. rPPG란?

rPPG: Remote + PPG = 비접촉식 PPG

실시간으로 얼굴인식을 진행하여 츄득한
안면이미지를 데이터로 사용하여 피부색의
미묘한 변화를 분석함으로써 활력 징후를
측정.



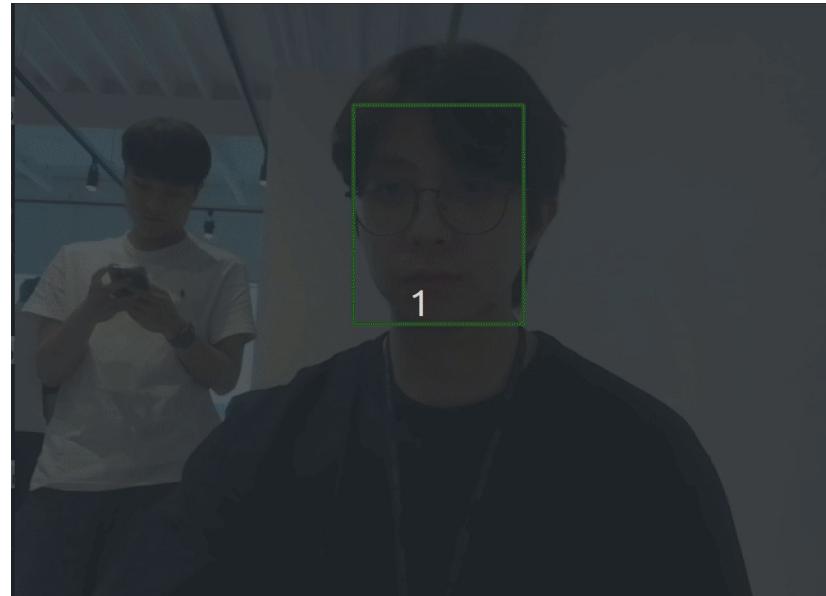
04 프로젝트 수행 경과 - 병동 로봇

▶ Vital Check - vital check node

rppg 특성상 움직임에 대한 노이즈 영향 多

KCF Tracker (Kernelized Correlation Filters)를 이용하여 얼굴 추적

사전 학습된 Face_detector 모델을 이용하여 ROI 영역 생성

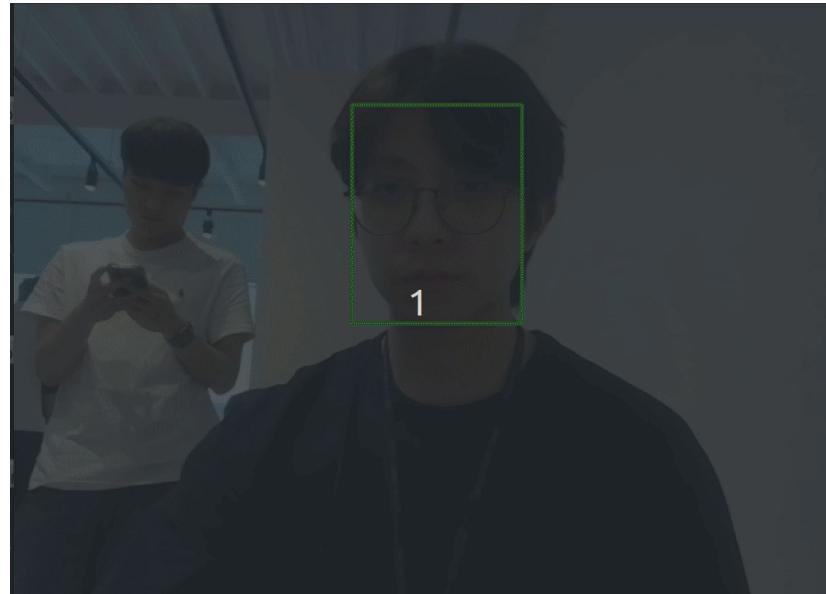


04 프로젝트 수행 경과 - 병동 로봇

▶ Vital Check - vital check node

또한 빛에 따른 노이즈 영향 多

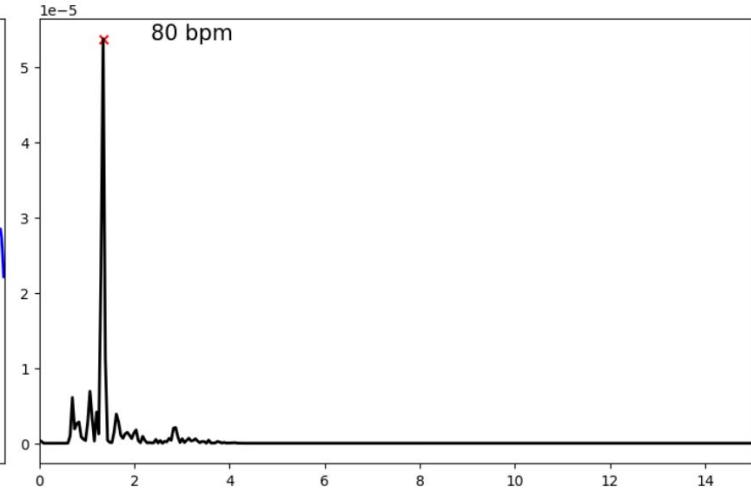
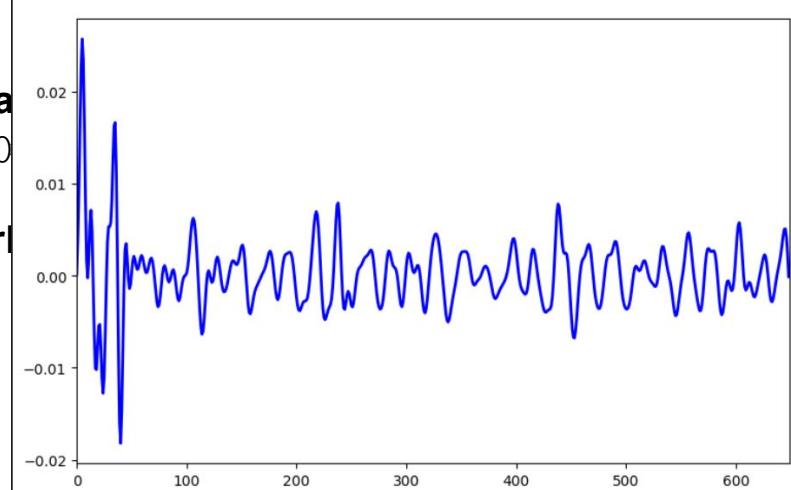
RGB의 평탄화를 사용하여 빛의 영향 최소화



04 프로젝트 수행 경과 - 병동 로봇

▶ Vital Check - vital check node

Bandpass
이용하는
추출한 rI
추정



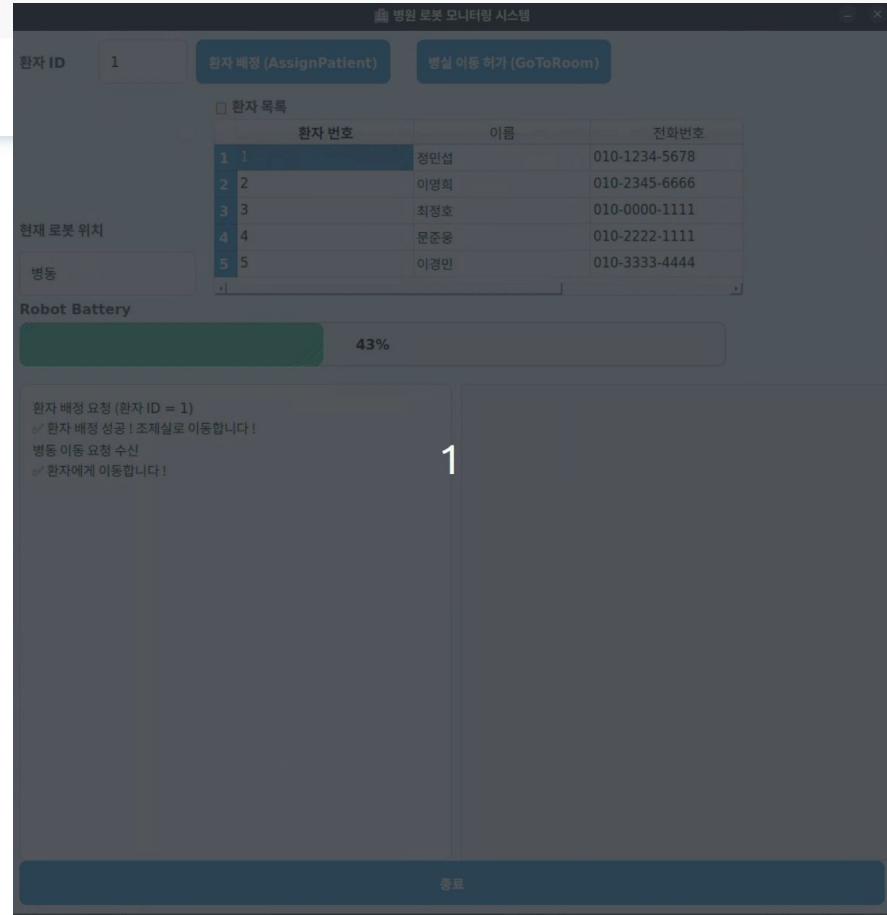
04

프로젝트 수행 경과 - 병동 로봇

▶ Vital Check - GUI

7. Vital Check

측정된 Vital signal은 GUI상에서 확인 가능



04 프로젝트 수행 경과 - 병동 로봇

▶ Docking

7. Docking

약 배송 후 docker에서 다음 명령 대기



04 프로젝트 수행 경과 - 병동 로봇

▶ 서브 시나리오



05

K-Digital Training

자체 평가 의견

▶ 자체 평가 의견

프로젝트 결과물에 대한 완성도 평가

9.5/10

잘한 부분과 아쉬운 점

기능별 노드를 분리하여 작업하여
프로젝트 일정을 준수할 수 있었음.
터틀봇의 리소스 경량화를 충분히 달성하지 못한 부분이
아쉬웠다.

추후 개선점이나 보완할 점

추후 터틀봇 이동을 정교화하기 위한
PID제어 추가 및
카메라의 실시간성을 위한
토픽 경량화 + 이미지 FPS 증가

느낀 점이나 경험한 성과

제한된 환경에서 터틀봇을 동시에 구동하는 데 어려움이
있었지만, 이를 해결하는 과정에서 네트워크 트래픽 관리와
리소스 분배에 대해 심도 있게 고민하고 해결 역량을 키움

05 자체 평가 의견 - 기대효과

▶ 기대 효과

- 자율주행 로봇 -
- 클라우드 기반 -
- 반복 업무 자동화 -
- 실제 의료 현장에



05 K-Digital Training 자체 평가 의견

▶ GitHub

GitHub 링크: https://github.com/mjw3723/nursing_assistance_robot

