

딥러닝을 활용한 **야간 케어봇**

김규빈 김한울 박성우 이민준

I N D E X

팀&서비스
소개

로봇
구동

넘어짐
감지

알람
기능

TEAM ATOMIC



김규빈

DETECTION

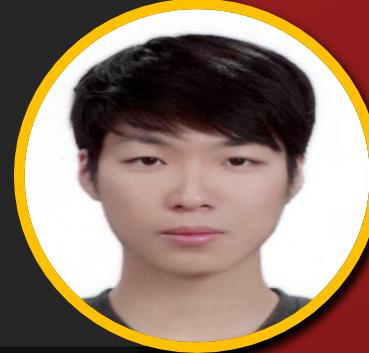


박성우



김한울

OPERATION



이민준
(조장)



SERVICE

- 요양원 야간 순찰을 통해 쓰러진 환자 파악



요양원 야간 근무자





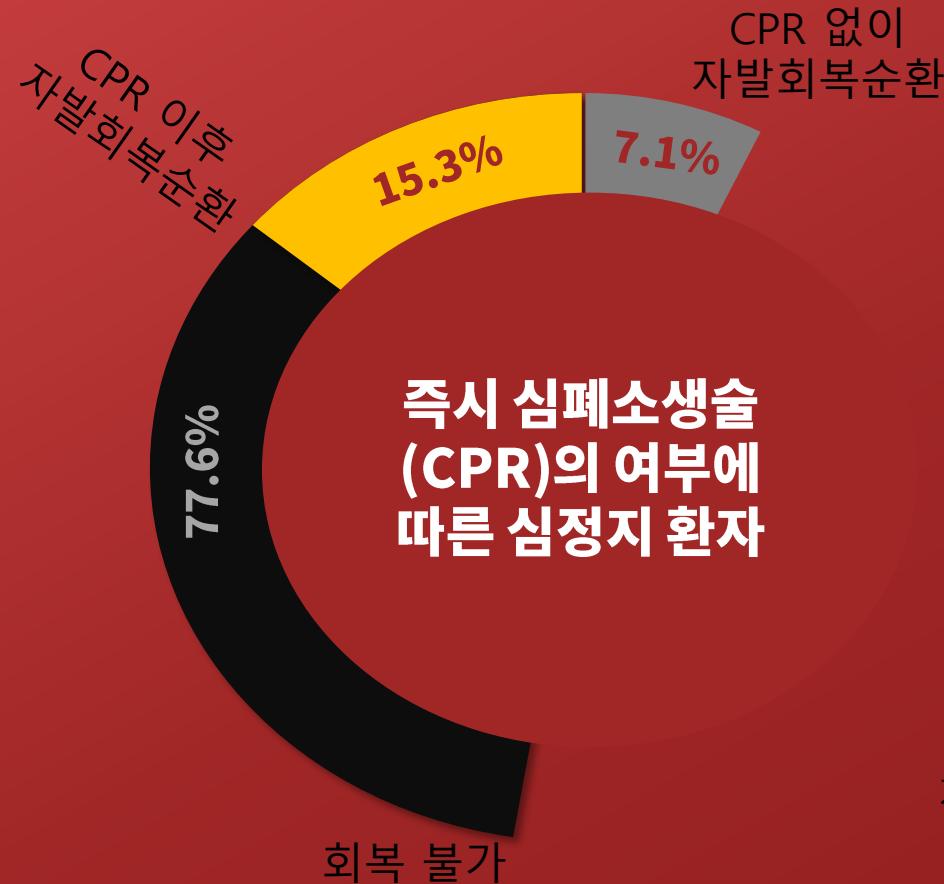
고령자 안전사고



이 중
67% 가
낙상사고!!

출처: YTN news – 고령자 안전사고

☰ 심정지와 골든타임(GOLDEN TIME)



자료 출처: 광주 소방청

생존율 2배 !

☰ 야간 순찰 터틀봇 소개



배려하고 지키는 터틀봇

최대 이동 속도: 0.26m/s
무게: 1.8kg
임베디드 컨트롤러: Open CR
센서: 라즈베리 파이 카메라, Lidar
CPU: Raspberry pi 3 B+
RAM: 1GB

☰ 서비스 프로세스



쓰러진 환자 발생



배지터가 상황 파악



요양 보호사에게 알람 전송

☰ 서비스 프로세스



쓰러진 환자 발생



배지터가 상황 파악



요양 보호사에게 알람 전송

☰ 서비스 프로세스



쓰러진 환자 발생



배지터가 상황 파악



요양 보호사에게 알람 전송



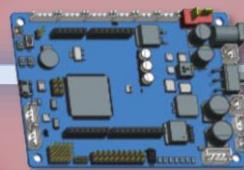
OPERATION

- Virtual Box & ROS를 이용한 로봇 구동

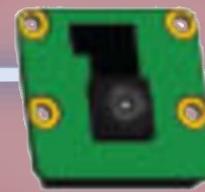


하드웨어 구성

Raspberry pi 3 B+
: 리눅스 기반의 소형 CPU



OpenCR
: 오픈소스를 기반으로 하는 로봇 제어기



Pi camera
: 카메라를 통한 객체 인식

Lidar
: 레이저를 사용하여 로봇 주위에 대한 정보를 수집

☰ Ubuntu & ROS를 사용한 이유



WINDOW

- 원도우 명령어
- 유료
- 제한적 정보



Ubuntu

- 리눅스 명령어
- 무료
- 많은 오픈 패키지

☰ Ubuntu & ROS를 사용한 이유



Virtual Box

우분투를 돌리기
위해서 사용한
가상환경



Ubuntu

터틀봇에 사용되는
ROS와 호환성이
좋아서 사용



ROS

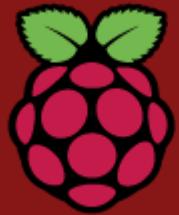
터틀봇과 컴퓨터에
서 사용하는 운영
체제



사용하는게 낫설어 많은 실수와 오
류를 겪음!!



Raspbian & ROS



Raspbian OS

: 터틀봇에 있는 메인 OS



ROS

: 터틀봇과 PC를 연결해
서 구동하는 OS



ROS를 통해 구동시키기

TELEOPERATION



PC에서 전송한 값
으로 조종



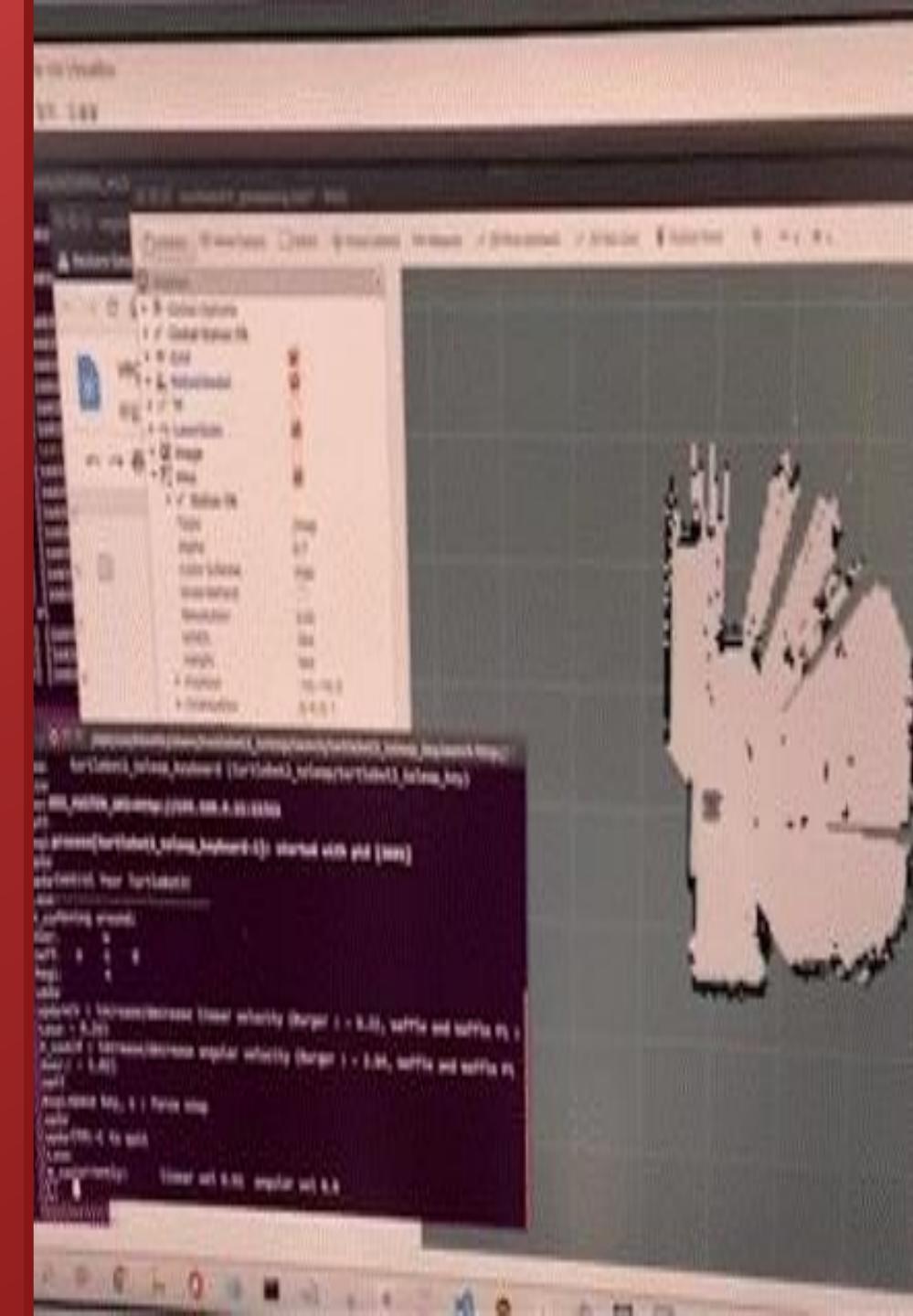


ROS를 통해 구동시키기

SLAM



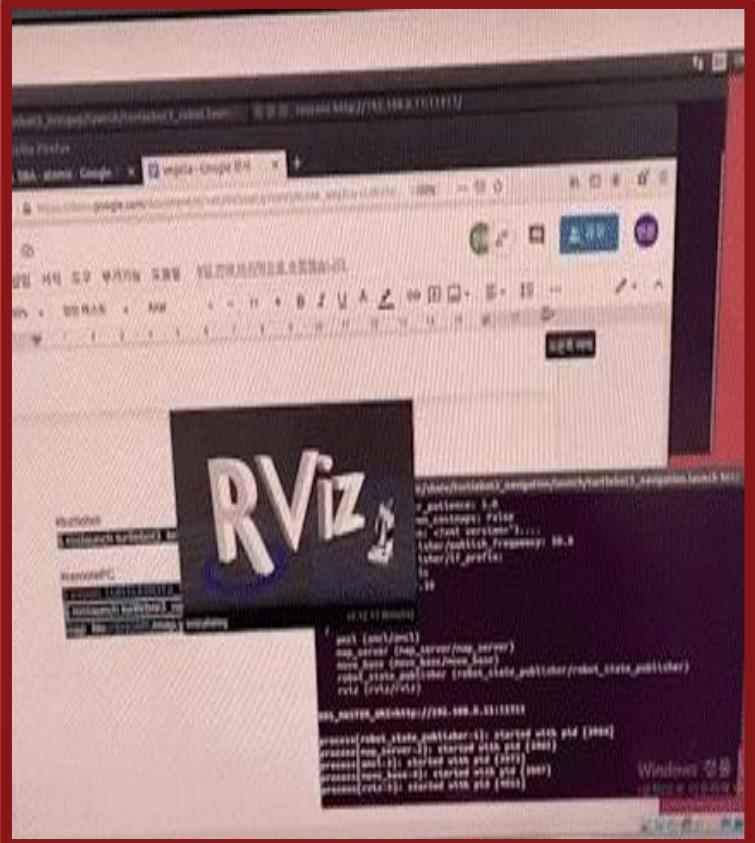
Lidar를 통해 맵을 파악
후 PC로 전송



NAVIGATION



ROS를 통해 구동시키기



<Remote PC>



<Turtlebot>

A photograph of a man lying on his back on a dark floor in what appears to be a hospital hallway. He is wearing a white short-sleeved shirt and dark trousers. His eyes are closed, and he has a pained or unconscious expression. His right arm is bent, with his hand near his head, and his left arm is extended towards the camera. In the background, there are wooden railings, a window, and a green exit sign. The overall atmosphere is somber and suggests a medical emergency.

FALL DETECTION

- 쓰러진 환자를 배지터가 인식



OBJECT_DETECTION :핵심 모듈

```
-- 모델링 가중치값
# object detection에 사용할 tensorflow그래프 파일
PATH_TO_CKPT =
os.path.join(CWD_PATH,MODEL_NAME,'frozen_inference_graph.pb')

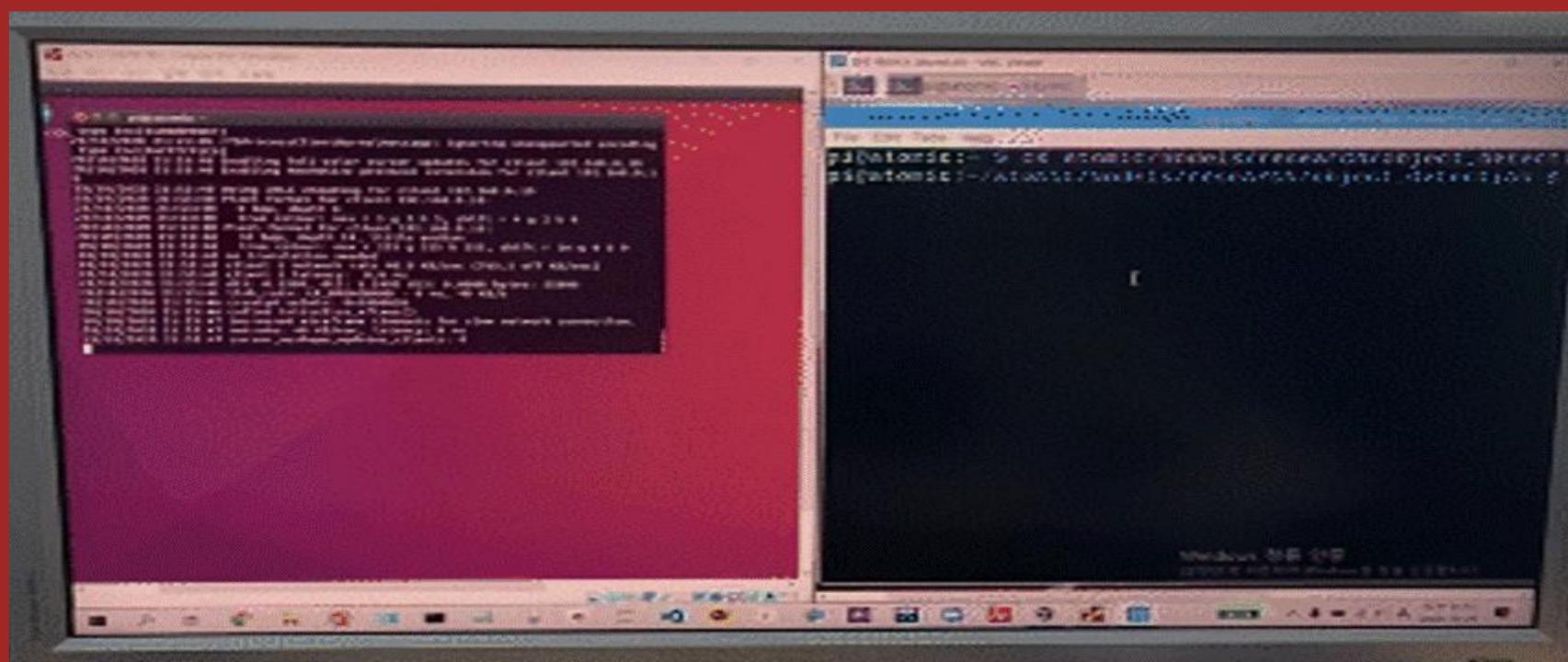
-- 클래스 레이블링값
# object detection에 클래스 분류에 사용되는 label파일
PATH_TO_LABELS = os.path.join(CWD_PATH,'data','mscoco_label_map.pbtxt')

-- 파이카메라에서 찍힌 이미지가 입력값으로 들어감
# input은 image가 들어가고
image_tensor = detection_graph.get_tensor_by_name('image_tensor:0')

-- 모델을 거쳐서 나온 결과값
## ouput은 detection box, score, classes이다
detection_boxes = detection_graph.get_tensor_by_name('detection_boxes:0')
detection_scores = detection_graph.get_tensor_by_name('detection_scores:0')
detection_classes = detection_graph.get_tensor_by_name('detection_classes:0')
```



OBJECT_DETECTION



사람과 사물을 카메라로 파악한 후 바운딩 박스
(Bounding Box)를 처리하여 인지한다.

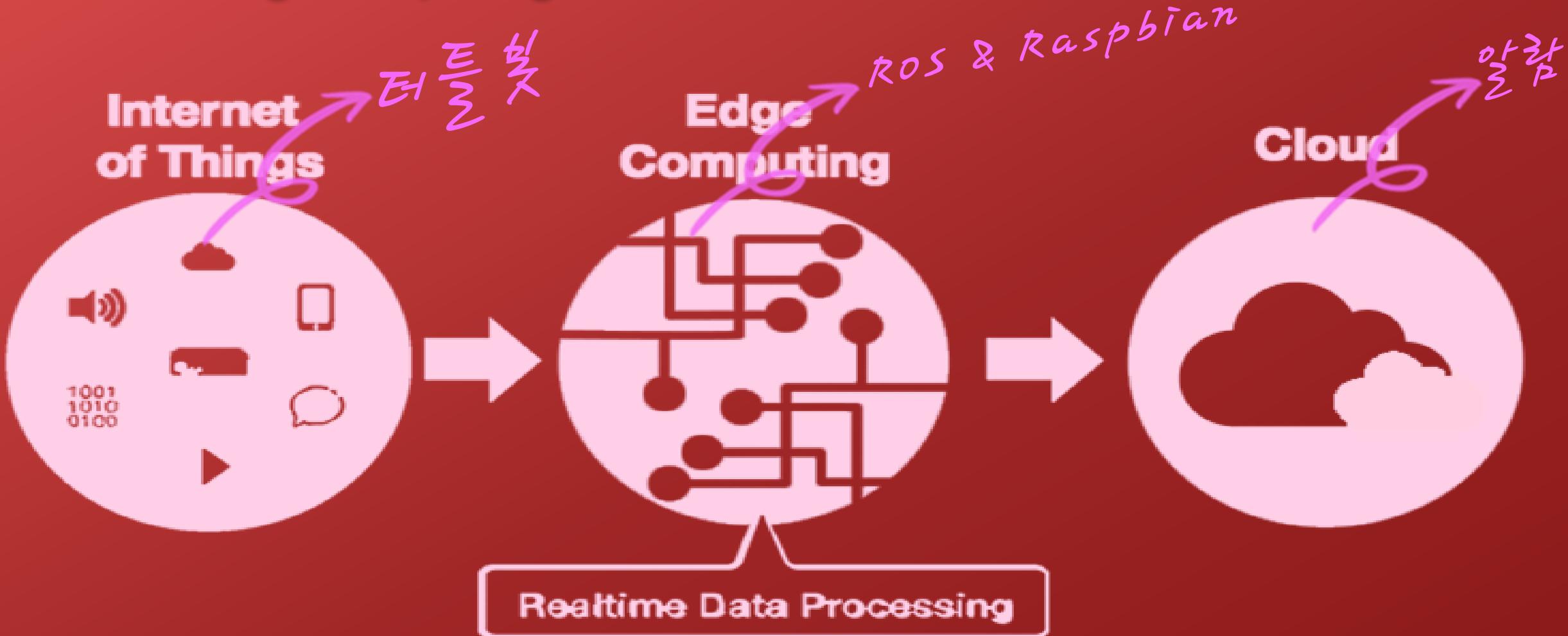


PROBLEMS

“기존에 존재하는 모델들은 터틀
봇에서 **호환**이 안되고, 새로운
모델을 적용시키기에는 터틀봇의
사양이 너무 낮아서 불가능.”

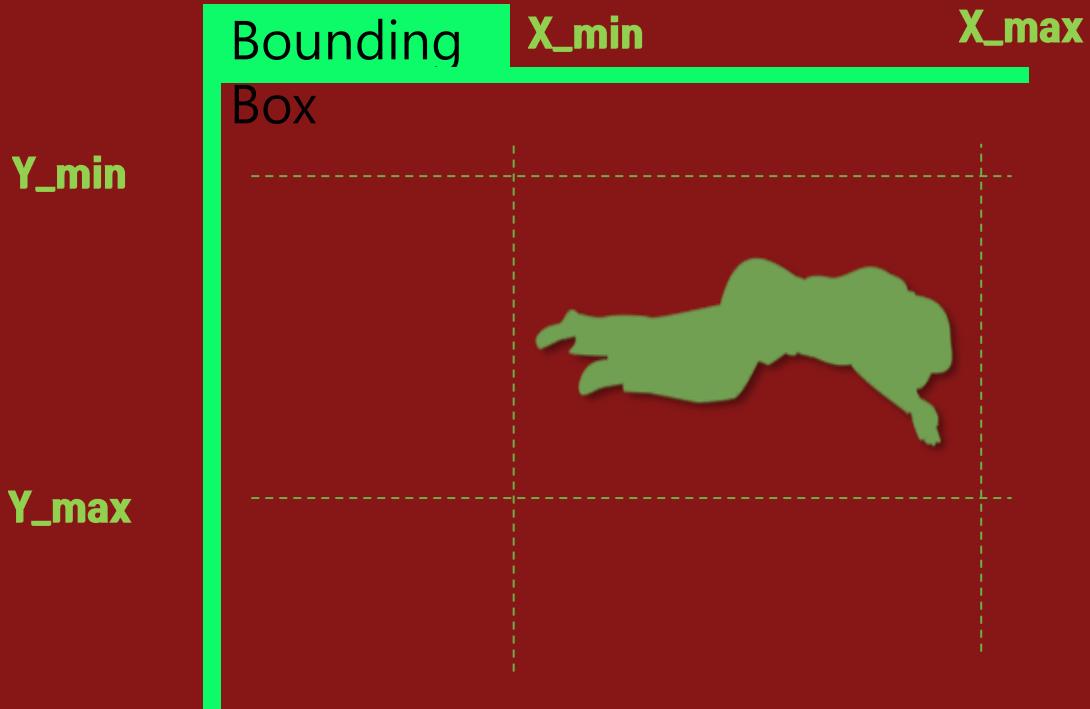


Edge Computing

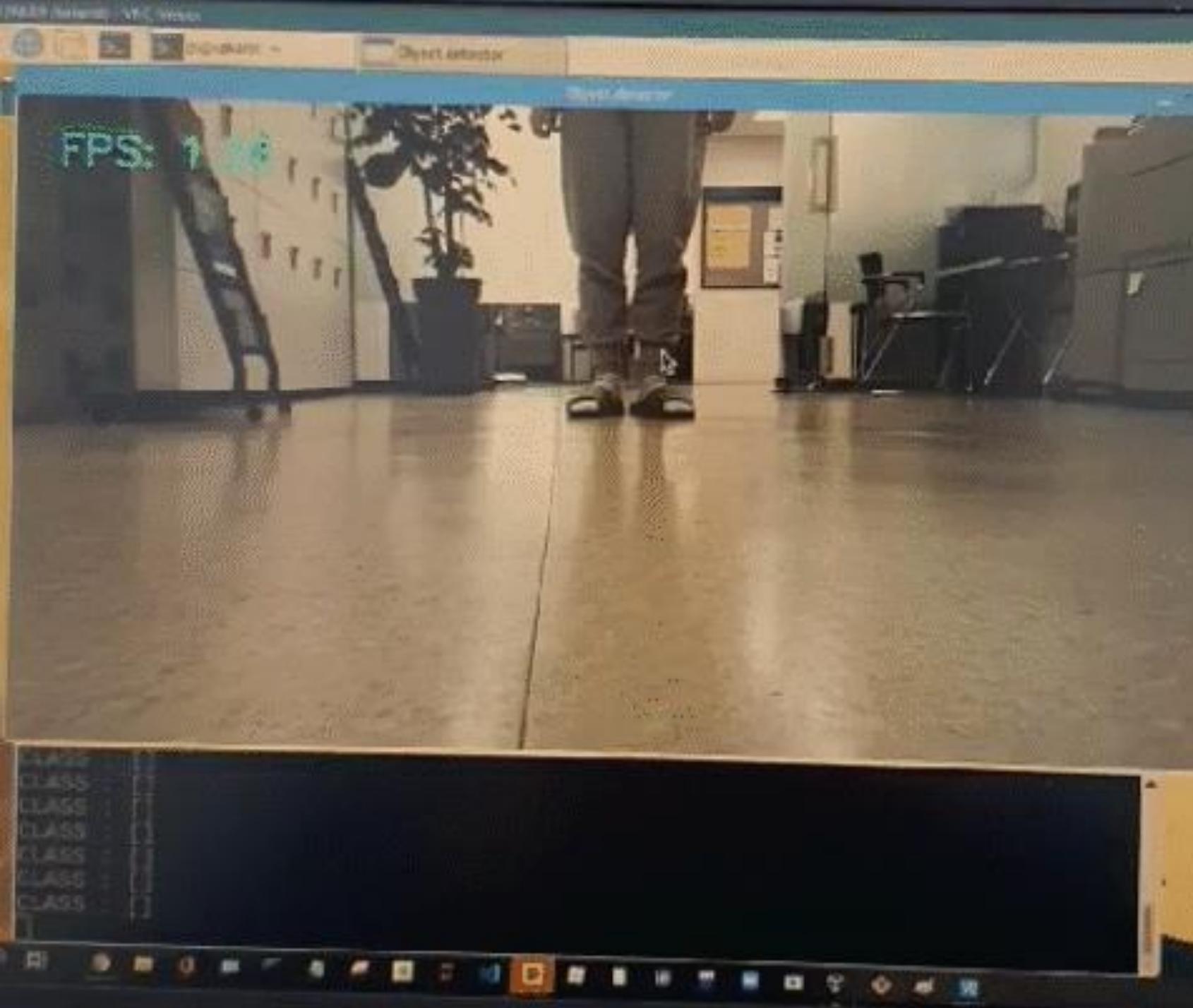




FALL_DETECTION



바운딩 박스(Bounding Box) 크기의 일정 비율을 통해 파악

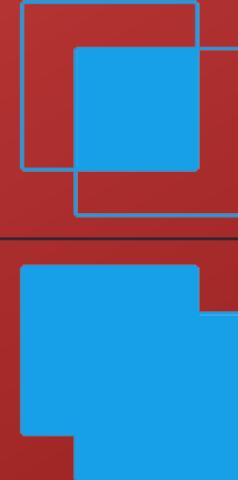


ATOMIC



정확도

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



정확도는 80%



ALARM

- 쓰러진 환자 인식 후 알람 기능



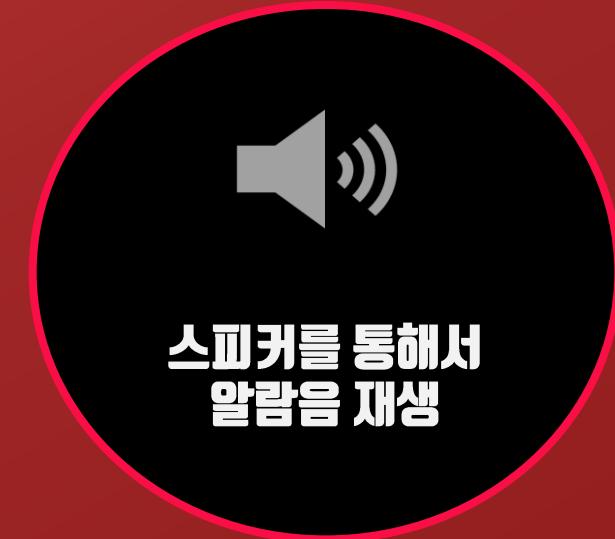
ALARM



쓰레짐 인식 후
알림을 재생 신호를
전송



터틀봇과 스피커를
블루투스를 통해 연결



스피커를 통해서
알람을 재생



ALARM



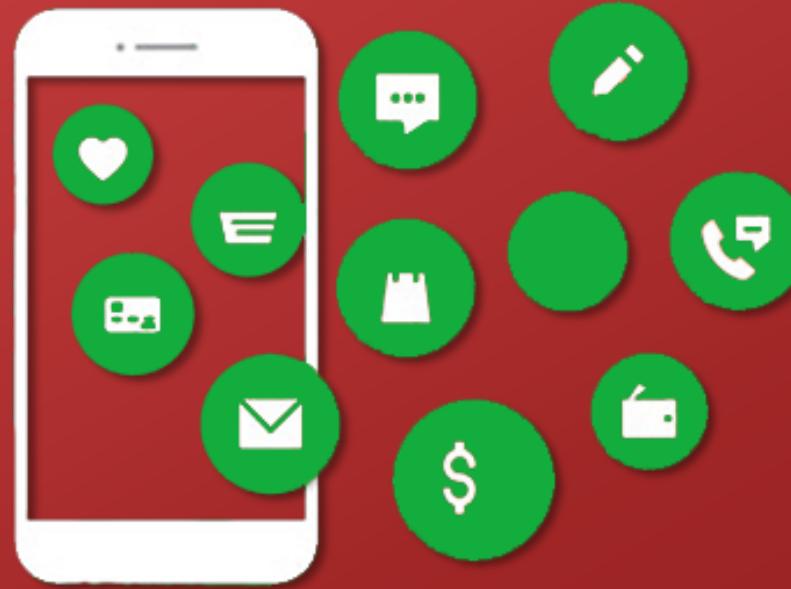
Bluetooth는 있지만
되지 않는다…



별도의 패키지
설치 필요



ALARM



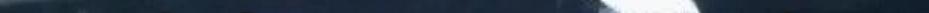
모바일 앱을 사용하여 요양 보호사에게 메시지를 전송



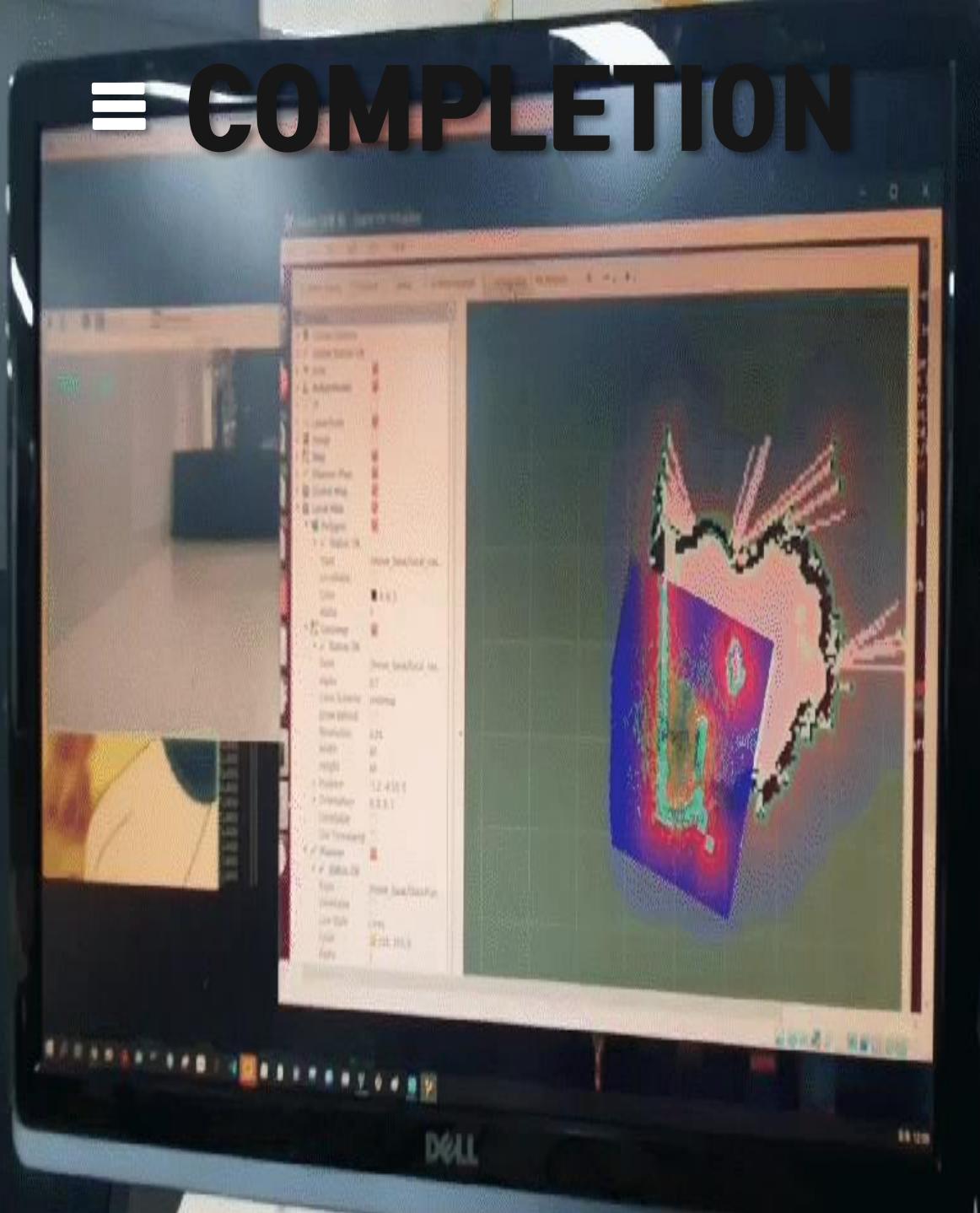
ALARM



메시지 전송 기능 대신
자체 알림음 기능 탑재



≡ COMPLETION





PLANS

- 향후 계획



요양원 내에서의 서비스

요양원 특성상 바닥에 장애물들이 없음

조명 밝기는 카메라가 인식할 수 있을 정도

어르신들의 걸음속도에 맞춰 속도 조절이 가능





한계점

골든타임을 지킬 수 있도록 도와주지만 예방은 할 수 없다.

하드웨어 사양이 지금 서비스를 구현하는 기준에서 현저히 낫다.



쓰러진 사람을 구분하는 정확도가 서비스화 하기에 낫다.



SOLUTION



TELEOP

SLAM

NAVIGATION

FALL_DETECTION

ALARM

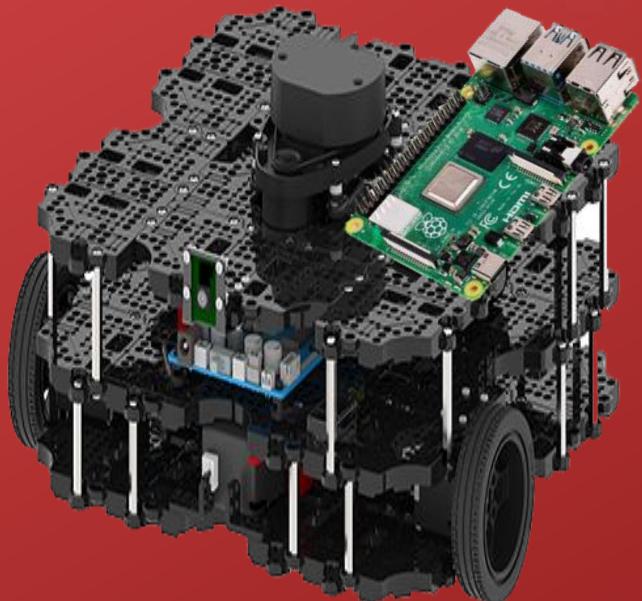
전체적인 성능이 좋아짐

FALL_DETECTION을 업그레이드하는 불가능

BUILT



SOLUTION



Raspberry pi 4

CPU: ARM Cortex-A72 **1.5GHz**

RAM: **4GB**

USB: USB 3.0

가격: **₩80,000**



더 많은 기능 구현 가능!!



발전 방향



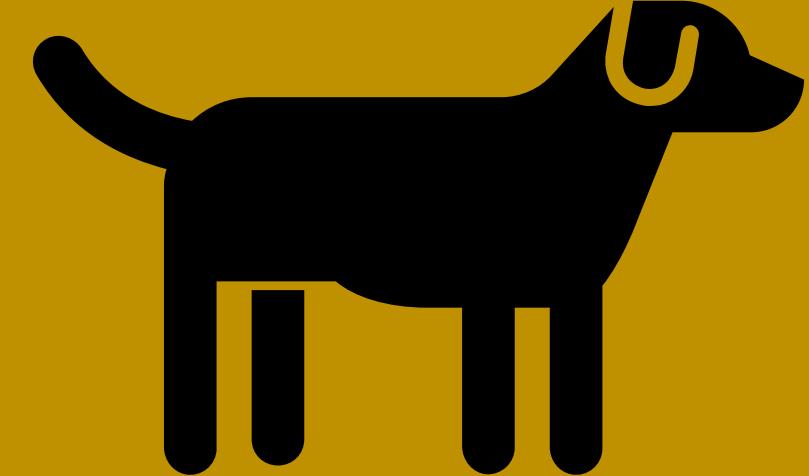
시각 장애인 안내로봇

- 자율주행 기능 추가 필요
- 안내견과 달리 법적 규제 없음
- 시각 장애인 시설이 없어도 판단 가능



보안용 경비 로봇

- 묘양원 케어봇과 가장 비슷한 구조
- 동시에 서비스화 가능



반려 로봇

- 자율주행 기능 추가 필요
- 하드웨어 추가 필요
(EX. 간식 공간)
- 주인과의 통신 추가 기능 필요



PROJECT SCHEDULE

10/16 :
프로젝트
시작

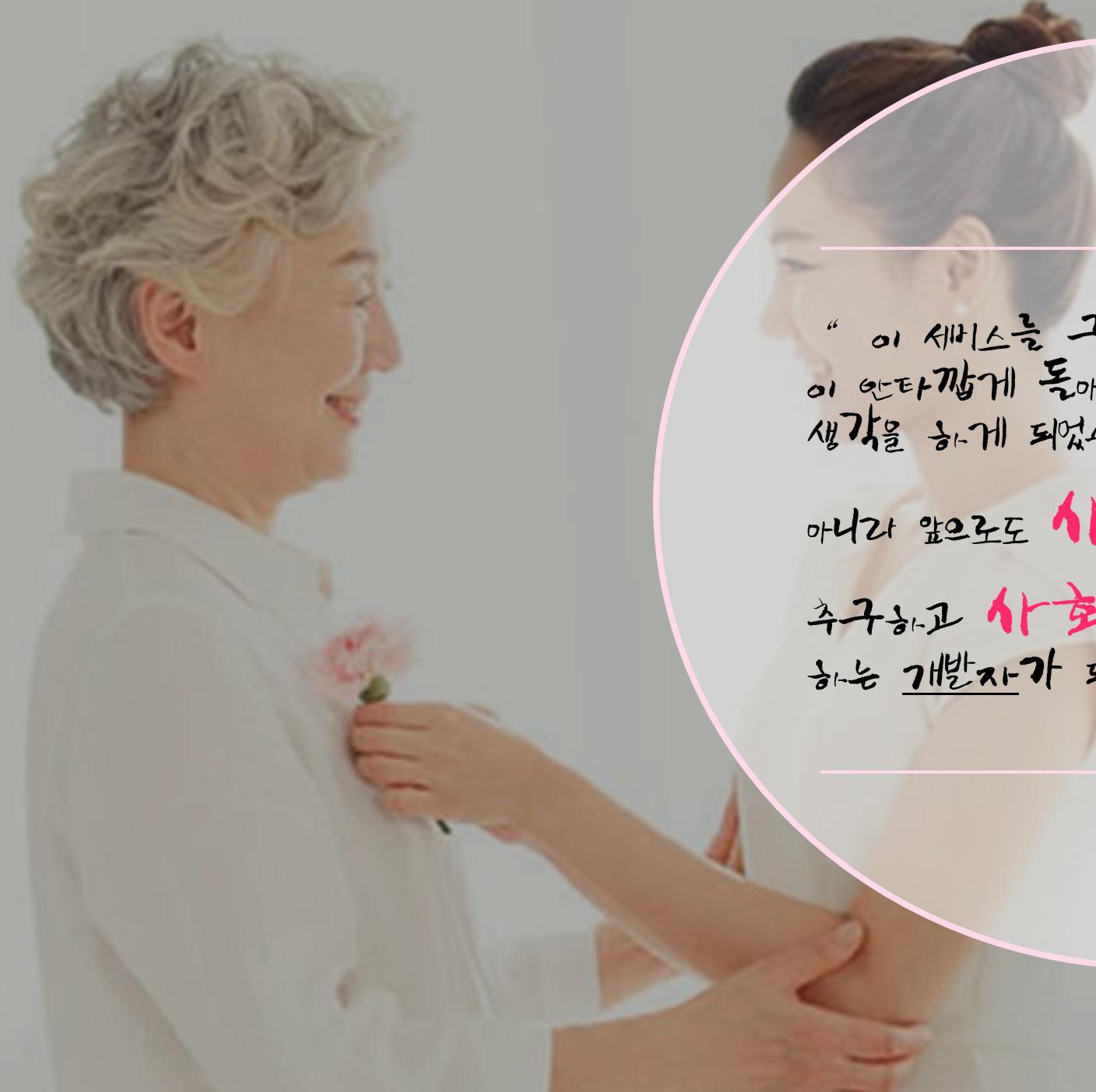
10/23 :
로봇 구동
마무리

11/09 :
메인 서비
스 완성

10/30 :
중간
발표

11/13 :
최종 발표

R
E
V
I
E
W



“이 서비스를 구상하면서 많은 어르신들이
이 안타깝게 돌아가시는 경우를 보고 깊은
생각을 하게 되었습니다. 이 서비스가 끝이

아니라 앞으로도 **사회적 가치**를
추구하고 **사회 복지**에 큰 기여를
하는 개발자가 되고 싶습니다!”

감사합니다

