AI Project

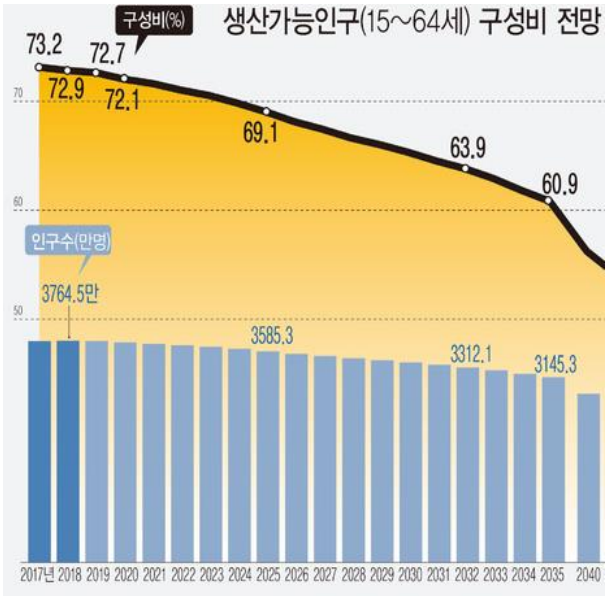
AI Bulldozer PODO

A2

김정원, 김주연, 이무동, 이우철, 인바다

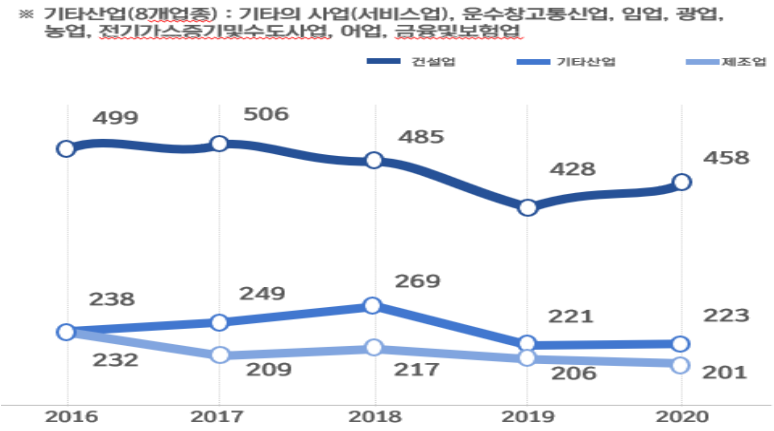
목차

1. 프로젝트 개요
    a. 추진 배경
        i. 생산가능인구 감소



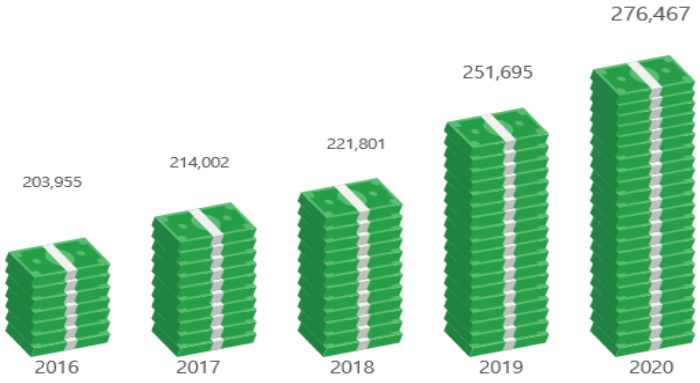
- 경제 활동에 참여하는 주된 연령층인 15~64 세 사이 인구가 2 년 전 정점을 찍은 후 감소세를 지속하고 있다.
- 65 세 이상 고령인구가 급속하게 늘어나는 반면 40 대 인구는 급감하고 있는 상황이다. 미국 경제학자 해리 덴트는 관련 저서에서 소비를 많이 하는 45~49 세 인구가 급속도로 줄어들면 경제 활동의 위축으로 심각한 경제 위기가 발생한다고 경고했던 바 있다.
- 생산연령인구는 경제활동을 하는 주 연령대로 이 인구의 감소는 소비 패턴과 산업 구조 등에서의 변화를 가져와 경제 성장 전체에 영향을 준다고 볼 수 있다

        ii. 안전사고 증가와 그에 따른 경제적 손실 증가



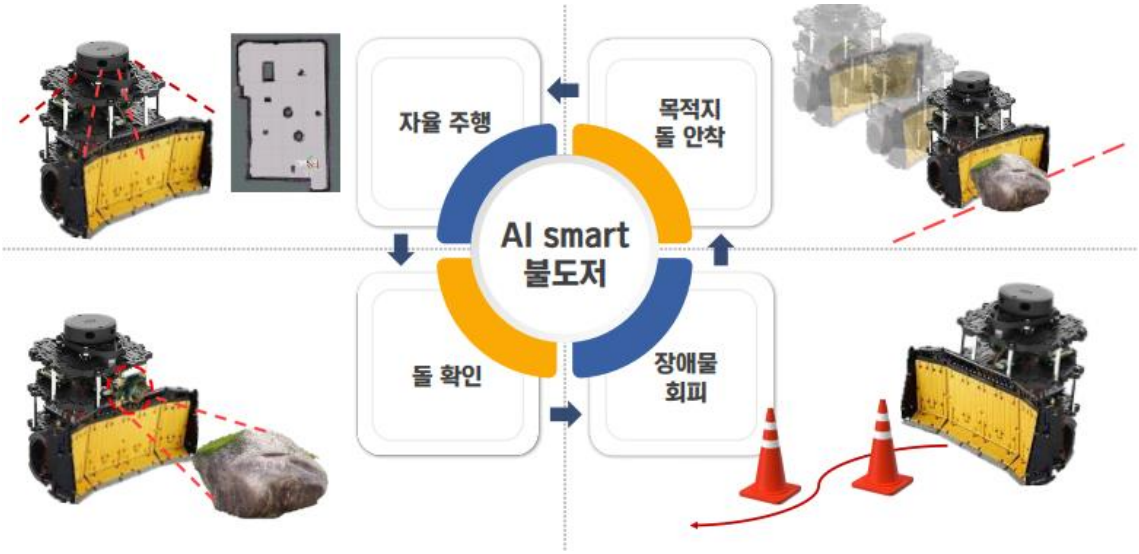
- 고용노동부는 올해 산재예방 지원사업에 1 조 1000 억원의 예산을 투입해 소규모 사업장의 안전관리 역량 향상을 위한 재정·기술지원을 확대할 방침이다.
- 산업재해로 인한 경제적손실 추정액은 2015 년 20 조원에서 2016 년 21 조원, 2017 년 22 조원, 2018 년 25 조원, 2019 년 29 조원, 2020 년 30 조원, 2021 년(5 월말) 13 조원으로 최근 7 년간 꾸준한 늘어나고 있다.
- 사고사망자를 산업별로 분석하면 최근 7 년간 건설업이 2,813 명으로 절반 가까이를 차지하고 있으며 제조업이 1,393 명, 서비스업 876 명 등이 뒤를 이었다.

b. 관련 이전 기수 프로젝트
- 5 기 B 반 4 조 CAR_GO platooning 기능을 포함하고 있는 택배 배달 자율주행 자동차
- 6 기 B 반 3 조 PODEX 2D 지도 작성과 영상 & 음성 인식 기능을 갖춘 무인 택배로봇
- 8 기 A 반 3 조 A.S.CAR Object Tracking 을 활용한 사용자 추적 쇼핑카트
- 13 기 B 반 2 조 chaGAN 재고관리 효율화를 위한 무인 재고관리 로봇



- 14 기 B 반 2 조 CARD 자율주행 차량을 이용한 드론 배송 서비스
- 16 기 A 반 1 조 포팡맨 실내자율주행 배송로봇
- 16 기 A 반 2 조 PORY User Tracking AI Cart
- 16 기 B 반 4 조 Poloro Smart Warehouse(자율주행 운반 로봇, 스마트 CCTV)
- 18 기 A 반 3 조 CamPOS GO(자율주행 캠퍼스 안내 로봇)
- 18 기 A 반 4 조 PO 가요(물류 자동화 로봇)

c. 프로젝트 소개



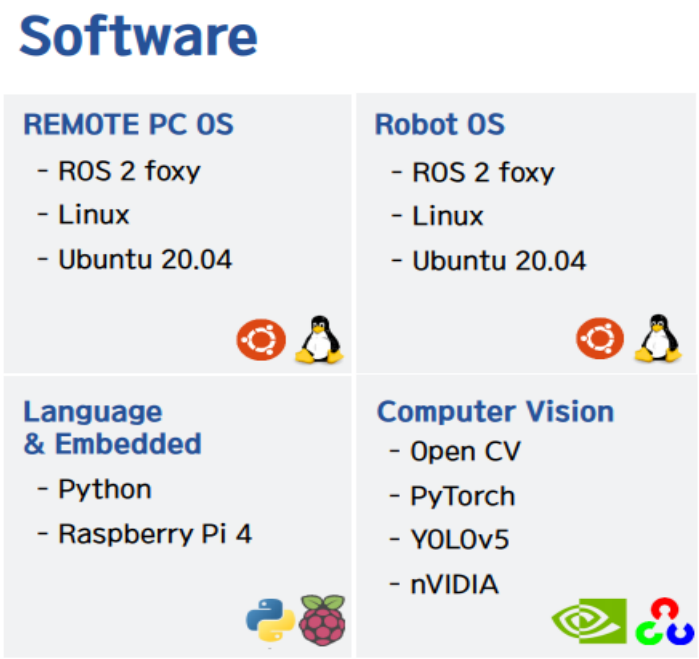
불도저는 흙, 돌, 모래 등을 앞에 달린 bucket 을 이용해 밀어서 땅을 평탄하게 하는 용도로 사용된다. 본 프로젝트에서는 평탄하게 하고자 하는 구역을 인지한 후 장애물을 회피하며 자율주행을 진행하고, 주행 중 돌을 발견하게 되면 정해진 구역으로 해당 돌을 옮기는 작업을 수행한다.

2. 구조도 및 프로세스

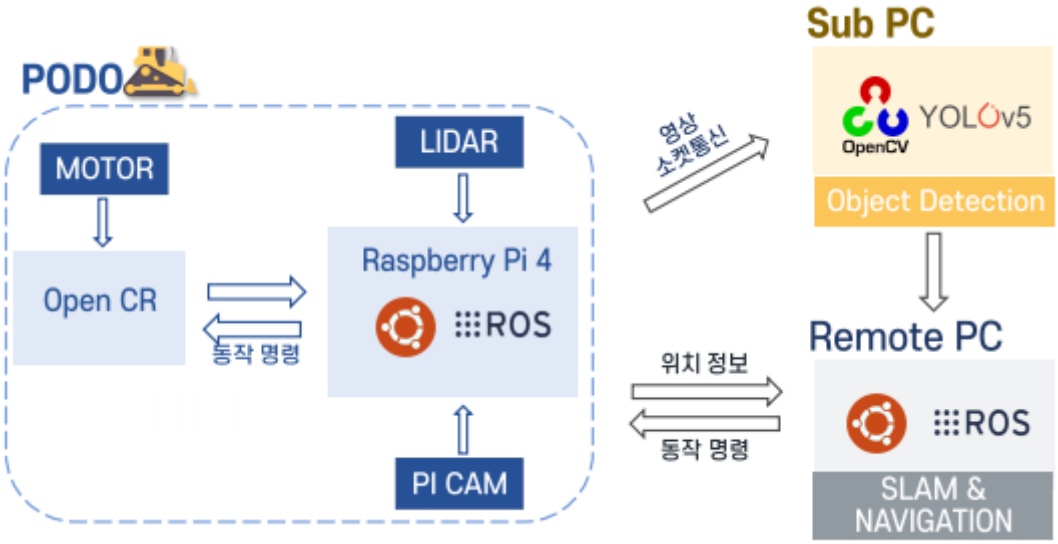
a. HW

1) LIDAR: LIDAR 는 빛 감지 및 거리측정의 준말로, 감지 센서를 통해 특정 무체의 거리를 측정하는 기법을 말한다. 광학 펄스로 목표물을 비춘 후 반사된 반송 신호의 특징을 측정한다.
2) Raspberry Pi4: 초소형 컴퓨터로 Remote PC 로 부터 받은 명령을 바탕으로 OpenCR 에 동작 명령을 내린다. 또한 Pi camera 를 통해 얻은 영상을 Object Detection 을 위한 Sub PC 에 소켓 통신을 이용해 frame 단위로 전송한다.
3) OpenCR: ROS 를 지원하는 보드이며 Turtlebot3 의 메인 제어기로 쓰인다.

b. SW



c. 프로세스



1) Open CR setup

- Micro USB 케이블을 사용하여 OpenCR 을 Rasbperry Pi 에 연결한다.
- 필요한 패키지를 Rasberry Pi 에 설치하여 OpenCR 펌웨어를 업로드한다.
- 플랫폼에 따라 OPENCR_MODEL 이름에 버거 또는 와플을 사용한다.
- 펌웨어와 로더를 다운로드한 다음 파일을 추출한다.
- OpenCR 에 펌웨어 업로드한다.

2) Remote PC setup

- PC 에 적합한 Ubuntu 20.04 LTS 를 다운로드한다.
- ROS 2 를 설치한다.
- Gazebo11, Cartographer, Navigation2 을 설치한다.
- Turtlebot3 패키지를 설치한다.
- PC 의 ROS 환경설정을 맞춘다.

3) Raspberry pi 4 setup

- Raspberry pi 버전에 맞는 ROS2 FOXY image 를 다운로드한다.
- 다운로드한 이미지 파일의 압축을 푼다.
- 이미지 파일을 굽는다. 이때, Raspberry pi imager 혹은 Linux Disks 를 이용할 수 있다.
- 이후, 파티션 크기를 조정한다.(할당되지 않은 공간을 사용하기 위해)
- WiFi 네트워크 설정을 구성한다.

```
network:
  version: 2
  renderer: networkd
  ethernets:
    eth0:
      dhcp4: yes
      dhcp6: yes
      optional: true
  wifis:
    wlan0:
      dhcp4: yes
      dhcp6: yes
      access-points:
        WIFI_SSID:
          password: WIFI_PASSWORD
```

- 마지막으로,ROS2 네트워크를 구성한다.

4) Turtlebot connecting

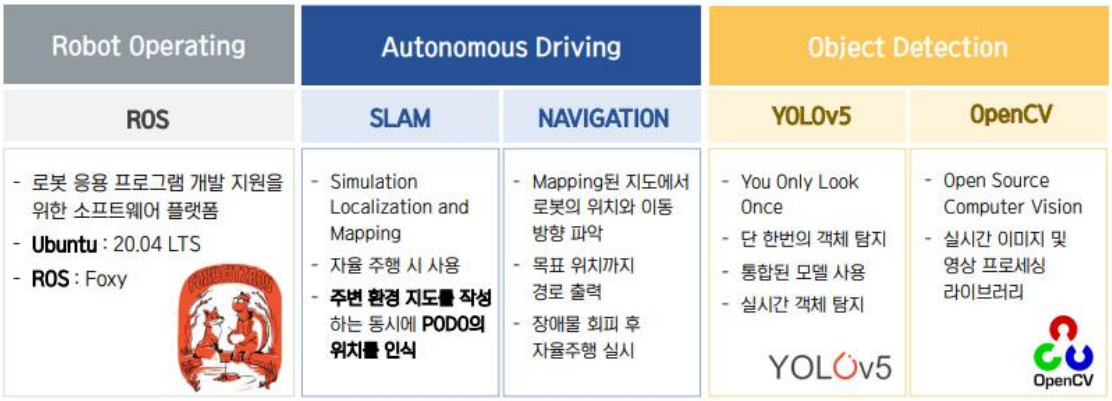
- Remote PC 와 Raspberry Pi 를 IP address 로 연결한다. (default password 는 turtlebot 이다.)
- TurtleBot3 애플리케이션을 시작하기 위한 기본 패키지를 불러온다. 이때, TULLBOT3_MODEL 파라미터는 버거, 와플, 와플_pi 중 적절한 키워드를 사용해야 한다.
- TurtleBot3 모델이 아래와 같이 버거인 경우, 단말기는 아래 메시지를 출력한다.

```
$export TURTLEBOT3_MODEL=burger
$ros2 launch turtlebot3_bringup robot.launch.py
```

- Rviz 를 사용하고자 한다면 아래의 메시지를 출력한다.

```
$ ros2 launch turtlebot3_bringup rviz2.launch.py
```

3. 적용 기술



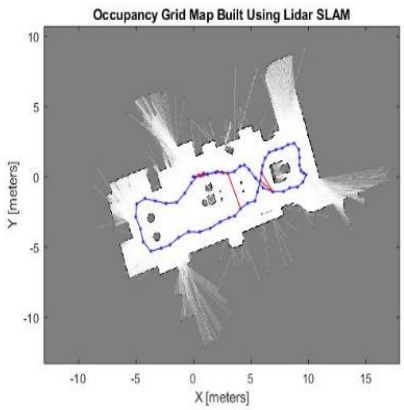
a. Autonomous Driving(SLAM & NAVIGATION)

1) SLAM (Simultaneous Localization And Mapping)

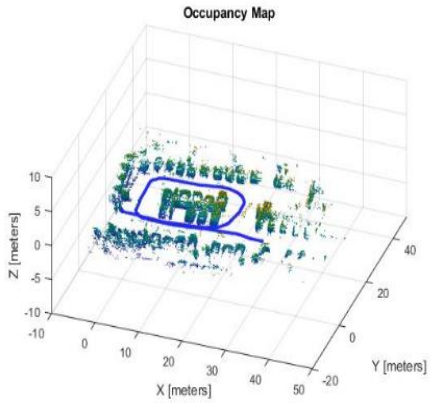
SLAM (Simultaneous Localization And Mapping)은 우리말로는 동시적 위치 추정 및 지도 작성이라고 한다. SLAM 은 로봇 공학에서 주행이 가능한 로봇을 이용해 임의의 공간에서 주변을 탐색해 지도를 작성하면서 동시에 현재 로봇의 위치를 추정하는 방식이다. 향상된 컴퓨팅 성능과 다양한 센서들에 의해 SLAM 개발은 활발해 지고 있다. SLAM 의 종류는 크게 LIDAR 센서를 이용한 LIDAR SLAM 과 Depth 카메라를 이용한 Visual SLAM 이 있다. LIDAR SLAM 에 사용되는 LIDAR 센서는 360 도 전방향의 물체를 인식 가능하여 제한된

Fov(Field of View)를 가진 Visual SLAM 에 더 넓은 범위의 물체를 인식 가능하다. SLAM 을 위해서는 두 축의 구동 모터로 이루어진 형태로 좌/우측 바퀴가 따로 구동 가능한 차등 구동형 모바일 로봇(Differential Drive Mobile Robot) 또는 옴니 휠 3 개 이상의 구동축을 가지고 있는 전 방향 이동 로봇(Omni-wheel Robot)과 같이 X, Y 축 병진속도 명령어와 theta 회전속도 명령어로 동작할 수 있어야 한다. 또한 오드메트리 정보를 얻을 수 있어야 한다.
ROS 에서 사용할 수 있는 SLAM 알고리즘은 gmapping, hector, cartographer, karto 등이 있다. frontier_exploration 방식은 1997 년, Gmapping 방식은 2007 년, Hector 방식은 2011. Cartographer 방식은 2016 년에 제안되었다.

2) LIDAR SLAM
LIDAR (Light Detection and Ranging)는 레이저 센서 (또는 거리 센서)를 주로 사용하는 방법이다. 레이저는 카메라와 ToF 등의 센서보다 훨씬 더 정밀하여 자율주행 차량과 드론처럼 빠르게 이동하는 물체와 관련된 응용 사례에 사용된다. 레이저 센서에서 얻어지는 출력값은 일반적으로 2 차원(x, y) 또는 3 차원(x, y, z) 포인트 클라우드 데이터이다. 레이저 센서 포인트 클라우드를 사용하면 고정밀 거리 측정이 가능하며, SLAM 을 적용한 지도 생성에도 매우 효과적이다.



2차원 라이다를 사용한 SLAM



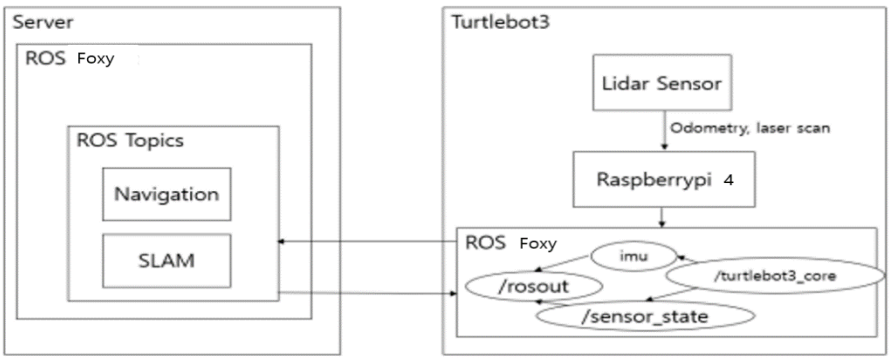
3차원 라이다를 사용한 SLAM

ROS 에서 SLAM 으로 작성한 지도는 위와 같이 그리드 기반의 Map 으로 저장된다. 지도에서 흰색은 로봇이 이동 가능한 자유 영역(Free area), 검은색은 이동 불가능한 점유 영역(Occupied Area), 회색은 확인되지 않은 미지의 영역(Unknown Area) 이다.

3) Navigation

Navigation 은 로봇이 특정 위치로 이동하기 위한 로봇의 방향과 속력을 결정하여 로봇을 원하는 위치로 이동시키는 과정이다. Navigation 을 위해서 로봇은 현재 자신의 위치를 알 수 있어야 한다. 현재 로봇의 자신의 위치를 추정하는 과정을 위치 추정(Localiztion)이라고 한다. 위치 추정은 센서로부터 장애물과의 거리 정보 등을 바탕으로 SLAM 등으로 미리 작성된 지도에서 로봇의 위치를 추정한다. Navigation 이 가능한 로봇은 박물관 투어가이드, 물류 운반등에 활용된다.

4) ROS2 기반 SLAM&NAVIGATION 작동 방법

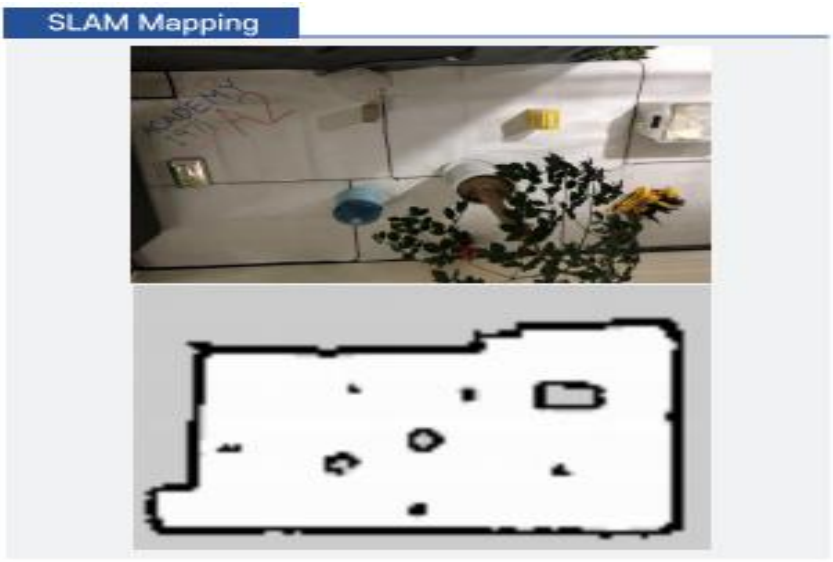


서버 역할을 하는 Linux Ubuntu 20.04.5 와 ROS2 의 Foxy 버전이 설치된 컴퓨터와 ROS2 Foxy 버전이 설치된 Turtlebot3 Burger 를 사용한다.

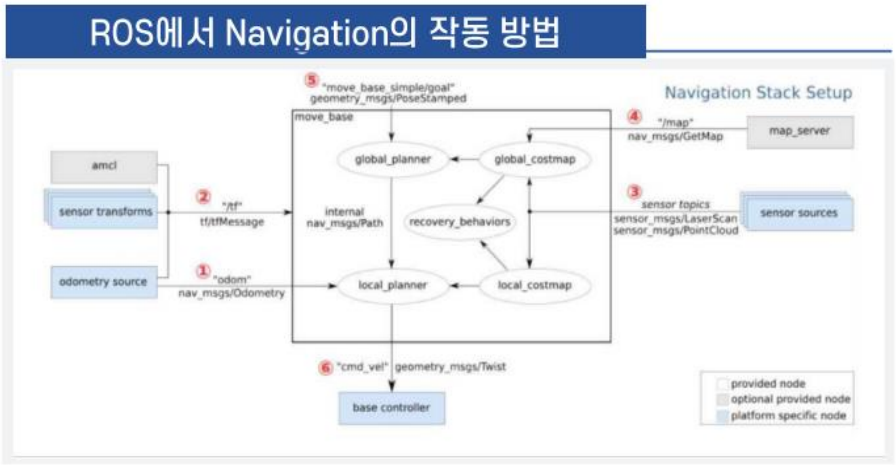
Turtlebot3 는 서버 역할을 하는 Remote PC 와 같은 공유기에 접속하여 서로 통신하는데, 본 프로젝트에서는 Mobile Hotspot 을 이용하여 통신을 가능케 하였다. (ex. IP_OF_REMOTE_PC = 192.168.0.0 / IP_OF_TURTLEBOT = 192.168.0.1)

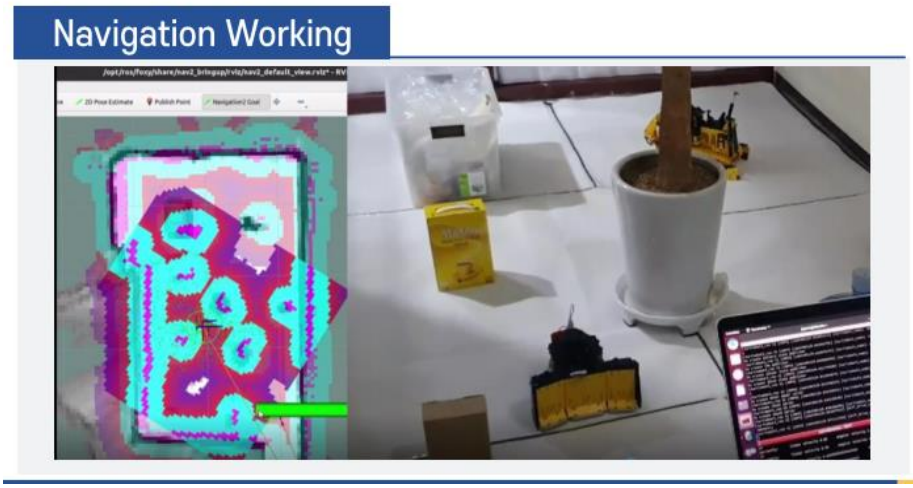
Turtlebot3 의 Lidar 센서 데이터를 Raspberrypi4 를 통해 서버로 전달하여 Remote PC 에서 SLAM 과 Map Building, Navigation 과정을 수행한다. Remote PC 를 이용하여 Turtlebot3 를 제어할 수 있다.

Turtlebot3_core 는 센서 데이터들을 전달 하도록 하여 Turtlebot3 의 Lidar 센서의 데이터를 Remote PC 에서 Navigation 과 Map Building 을 위해 사용할 수 있게 해준다.



SLAM Mapping 결과물에서 볼 수 있듯이, 위의 이미지는 A2 조원들이 구축한 가상의 공사 현장이고, 아래의 이미지는 'AI 불도저 PODO'가 SLAM 을 통해 구현한 공사 현장의 Mapping 결과물이다. 장애물의 형태와 크기를 모두 정확하게 반영한 것을 알 수 있다. SLAM 의 정확도 향상을 위해 로봇의 속도를 너무 급하게 바꾸거나 너무 빠른 속도로 전진과 후진 그리고 회전하지 않도록 주의해야 한다. 또한, 로봇을 이동시킬 때는 계측할 환경의 구석구석을 로봇이 돌아다니며 스캔하도록 하고 똑같은 장소를 여러 번 지나도록 한다.





Navigation 을 처음 실행했을 때 'AI 불도저 PODO'는 지도에서 자신의 위치를 찾지 못한다. 이 때, RViz 창에서 2D Pose Estimate 를 클릭한 후 지도에서의 'AI 불도저 PODO'의 실제 위치에 화살표를 생성한다. 화살표의 방향은 'AI 불도저 PODO'의 정면을 의미한다. 이는 초기에 'AI 불도저 PODO'의 위치를 추정하기 위한 방법이다.
'AI 불도저 PODO'를 이동시키면 녹색 화살표로 지정된 위치와 방향을 초기 자세로 삼아 스스로 자신의 위치와 방향을 추정한다. Remote

PC 에서 "turtlebot3_teleop turtlebot3_teleop_key.launch"를 실행하여 전후좌우로 움직여 주다 보면 'AI 불도저 PODO'가 스스로 주변 환경 정보를 수집하여 지도상에서 현재 위치를 알아낸다.
'AI 불도저 PODO'가 목적지를 정하고 이동시키기 위해서 RViz 의 메뉴에서 2D Nav Goal 을 누른다. 상기 이미지와 같이 큰 녹색 화살표가 나타나는데, 이는 'AI 불도저 PODO'의 목적지를 지정할 수 있는 Marker 로 화살표의 시작점이 'AI 불도저 PODO'의 x, y 위치, 화살표가 가리키는 방향이 'AI 불도저 PODO'의 방향이다. 'AI 불도저 PODO'는 작성된 지도를 기반으로 목적지까지 장애물은 피하며 이동하지만, 처리해야 하는 돌은 Bucket 에 Load 하여 목적지로 이송한다.

b. Object Detection

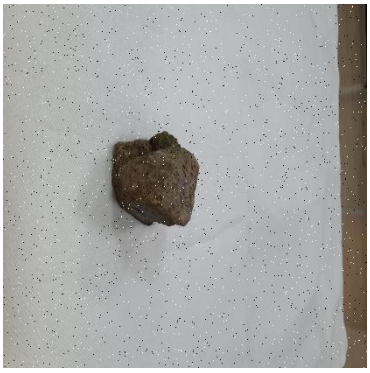
Object Detection 이란 이미지 내에서 물체의 위치와 종류를 찾아내는 기법이다. 이미지 기반의 문제를 풀기 위해서 다양한 곳에서 사용되는데, 대표적으로는 자율 주행을 위해 차량이나 사람을 찾아내는 경우, 얼굴 인식을 위해 사람 얼굴을 찾아내는 경우 등 다양한 방식으로 이용되고 있다.

Object Detection 은 사물의 클래스를 Classification 하고 localization 을 함께 수행한다.
Classification 이란, 주어진 이미지 안에 어떤 object 가 있는지 label 을 구분하는 행위이다.
Localization 은 이미지 안의 object 가 어느 위치에 있는지 정보를 출력해주는 것이다.

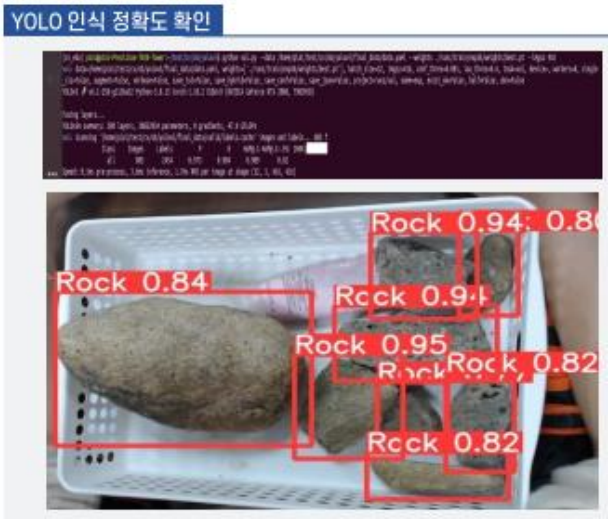
YOLOv5 를 사용했는데, V4 에 비해서 용량이 낮고 속도는 빠르지만 성능이 비슷하여 많이 쓰이기 때문이다. 또한 PyTorch 구현이 가능하다.

프로젝트에서 기존 실습시간에 활용한 cv_edu 내에서 YOLOv5 PyTorch 구현했다. roboflow 사이트에 이미 라벨링된 3000 장의 Rock 데이터셋을 찾아 활용했다. Epochs 100, batch size 64 로 학습을 시키고 준비한 돌을 구분하는지 확인했으나 정확도가 0.85 임을 확인했다.

그래서 정확도를 낮추는 데이터로 보이는 돌의 질감만 보이는 사진, 여러 개의 돌이 모인 사진, 돌이 아닌 사진을 찾아서 라벨과 함께 삭제했고 준비한 돌의 사진을 1000 장 준비했다.

또한 충분한 데이터를 위해 다른 돌 데이터셋을 찾았다. 라벨이 rock 이 아닌 'basalt', 'pebble', 'quartz'로 지정되어 있었고 nc 역시 3 이었다. 해당 데이터 라벨을 모두 rock 으로 변경하고 nc 를 1 로 변경하여 총 5000 장의 돌 데이터를 학습에 사용했다.



학습 결과 정확도를 0.989 로 향상시켰으며 인식 역시 잘되는 것을 볼 수 있다.

PODO 에 연결된 캠에서 돌을 인식했을 때 0.88 이라는 높은 정확도를 확인할 수 있다.

c. Socket communication
소켓통신을 이용하여 불도저 bucket 에 달린 picamera 를 통해 얻은 영상을 frame 단위로 object detection 을 위한 sub PC 에 전달한다. Client 는 picamera 가 연결된 raspberrypi 이고 server 는 object detection 을 위한 sub PC 이다.

client.py

```
4. # -*- coding: utf8 -*-
5. import cv2
6. import socket
7. import numpy as np
8.
9. ## TCP 사용
10.s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
11.## server ip, port
12.s.connect(('192.168.1.83', 8485))
13.
14.
```

```
15.## webcam 이미지 capture
16.cam = cv2.VideoCapture(0)
17.
18.## 이미지 속성 변경 3 = width, 4 = height
19.cam.set(3, 320);
20.cam.set(4, 240);
21.
22.## 0~100 에서 90 의 이미지 품질로 설정 (default = 95)
23.encode_param = [int(cv2.IMWRITE_JPEG_QUALITY), 90]
24.
25.while True:
26.    # 비디오의 한 프레임씩 읽는다.
27.    # 제대로 읽으면 ret = True, 실패면 ret = False, frame 에는 읽은 프
   레임
28.    ret, frame = cam.read()
29.    # cv2. imencode(ext, img [, params])
30.    # encode_param 의 형식으로 frame 을 jpg 로 이미지를 인코딩한다.
31.    result, frame = cv2.imencode('.jpg', frame, encode_param)
32.    # frame 을 String 형태로 변환
33.    data = numpy.array(frame)
34.    stringData = data.tostring()
35.
36.    #서버에 데이터 전송
37.    #(str(len(stringData))).encode().ljust(16)
38.    s.sendall((str(len(stringData))).encode().ljust(16) + stringDat
   a)
39.
40.cam.release()
```

server.py

```
1  import socket
2  import cv2
3  import numpy as np
4
5  #socket 에서 수신한 버퍼를 반환하는 함수
6  def recvall(sock, count):
7      # 바이트 문자열
8      buf = b''
9      while count:
10
```

```
        newbuf = sock.recv(count)
        if not newbuf: return None
        buf += newbuf
        count -= len(newbuf)
    return buf

HOST=''
PORT=8485

#TCP 사용
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
print('Socket created')

#서버의 아이피와 포트번호 지정
s.bind((HOST,PORT))
print('Socket bind complete')
# 클라이언트의 접속을 기다린다. (클라이언트 연결을 10 개까지 받는다)
s.listen(10)
print('Socket now listening')

#연결, conn 에는 소켓 객체, addr 은 소켓에 바인드 된 주소
conn,addr=s.accept()

while True:
    # client 에서 받은 stringData 의 크기 (==(str(len(stringData))).encode().ljust(16))
    length = recvall(conn, 16)
    stringData = recvall(conn, int(length))
    data = np.fromstring(stringData, dtype = 'uint8')

    #data 를 디코딩한다.
    frame = cv2.imdecode(data, cv2.IMREAD_COLOR)
    cv2.imshow('ImageWindow',frame)
    cv2.waitKey(1)
```

4. 결론
   a. 기대효과 및 활용방안
      i. 기대효과
         - 공사현장 작업 효율성 향상 : 실제로 현대건설에서 AI 기반 스마트 굴삭기를 자체 개발하여 상용화 시키고 있는데, 공사기간 및 비용을 약 20% 이상 줄일 수 있을 것으로 기대함.

- 안전사고 위험 감소 : 실제 건설업 공사현장에서의 사망사고 5 명중 1 명은 건설기계 및 장비가 원인이라고 함. 스마트 불도저를 통해 반복되는 재래형 사고를 단절할 수 있을 것으로 기대해볼 수 있다.
- 건설현장 인부 부족 문제 해결 : 기초 공사 작업 대부분을 스마트 중장비로 대체함으로써 건설현장 자동화를 실현시킬 수 있다.

ii. 활용방안

- PODO 의 앞 blade 모형을 다양하게 변형시켜 여러 종류의 장애물을 이동시키거나, 다른 공사자재들을 옮겨주는 시스템을 구성하여 공사현장에서 유용하게 사용될 수 있다.
- 대기질 측정 기술을 추가하여 공사현장에서의 대기 환경 개선에 긍정적인 영향을 미칠 수 있다.

b. 한계점 및 개선 기회

i. 한계점

- Socket communication 을 통해 Frame 단위로 전달하다 보니 delay 가 생겨 실시간 object Detection 을 수행하지 못했다.
- 돌의 무게가 너무 무거운 경우 속도가 느려지거나 움직이지 않았다.
- Pi camera 의 화질이 좋지 않았다.
- Pi camera 부착 위치로 인해 bucket 의 전면을 다 담지 못하여 사각지대가 존재하였다.

ii. 개선 기회

- Multithreaded Programming 을 통해 Socket communication 에서 발생한 delay 를 해결할 수 있을 것으로 보인다.
- 출력이 높은 turtlebot3 모터를 사용하여 무거운 돌을 옮길 수 있도록 한다.
- Pi camera 가 아닌 웹캠을 사용하여 화질을 개선한다.
- 시야각이 넓은 cam 을 사용하거나 높은 위치에 cam 을 거치하여 사각지대를 없앤다.

c. 참조 자료

https://newsis.com/view/?id=NISX20190328_0000602246
생산가능인구 내년부터 연 33 만명씩 급감… '인구절벽' 가속

https://www.electimes.com/news/articleView.html?idxno=227729
고용부, 중대재해처벌법 시행 맞춰 산재 사망사고 감축 추진방향 발표

http://www.todayenergy.kr/news/articleView.html?idxno=240395
7 년간 산업재해 경제적 손실 159 조원 추정

https://www.onsemi.com/site/pdf/ONSAR3048_Semiconductor%20Network_Using_dToF_in_LiDAR_applications_pg_76-80.pdf
라이다(LiDAR) 애플리케이션의 dToF 적용사례