

도로 안전 시스템

C104 - 박건국, 박민철, 정우영, 정하림, 지민경



로드 가 디 언 즈



01

로고 의미



02

배경 및 목표



03

데이터 흐름



04

역할 분배



05

진행 상황

01 WHAT IS 로드가디언즈 ?



01 WHAT IS 로드가디언즈 ?

ROAD
LOAD



로드
가디언즈

02 배경 및 목표

(서울=연합뉴스) 최윤선 기자 = 서울시는 4월 한 달간 대형 공사장 등 과적 근원지와 과적 차량이 많은 시내 주요 지점 등에서 24시간 집중 단속을 한다고 7일 밝혔다.

시가 도로안전을 위협하는 포트홀 등을 줄이기 위해 다양한 노력을 기울이는 가운데 주요 원인 중 하나인 과적 차량을 단속하고 나선 것이다.

단속 대상은 차량 축하중 10t 또는 총 중량 40t을 초과하는 과적 차와 적재물 포함 길이 16.7m·폭 2.5m·높이 4m가 넘는 차이다. 하나의 기준이라도 초과하면 단속 대상이다.

포트홀의 주요 원인 = 과적 차량

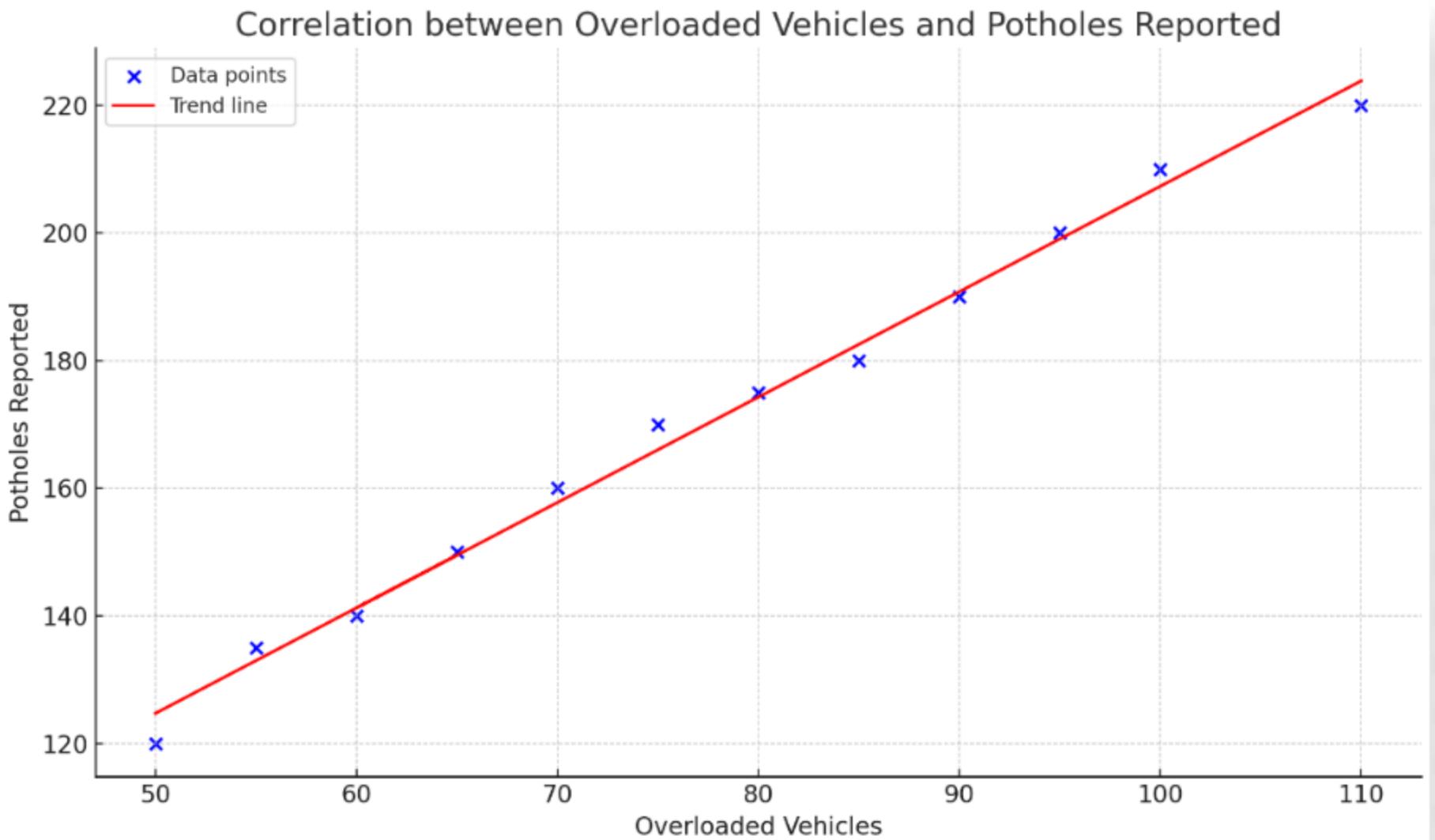
02 배경 및 목표

경기도내 등록 화물차가 해마다 증가하고 있는 가운데 싱크홀 등 도로 파손과 대형 교통사고의 주범으로 지목되고 있는 ‘과적 차량’ 운행이 되풀이되고 있는 것으로 나타났다. 더욱이 과적 자당 단속 인력이 세안석인 데나 최근 시행된 과적 자당에 대한 책임과 처벌을 강화하는 내용의 도로법 개정안마저 제역할을 못하고 있어 대책이 절실하다는 지적이다.

6일 국토교통부와 경기도 등에 따르면 최근 3년간 경기지역 등록 화물차는 2020년 80만2천851대, 2021년 81만8천406대, 지난해 82만8천787대 등으로 매년 증가세다. 평균적으로 1년에 1만2천968대씩 늘어나고 있는 셈이다. 올해(11월 기준)는 지난해보다 2만1천325대 늘어난 85만112대로, 이미 평균 증가치를 2배 가까이 넘어섰다.

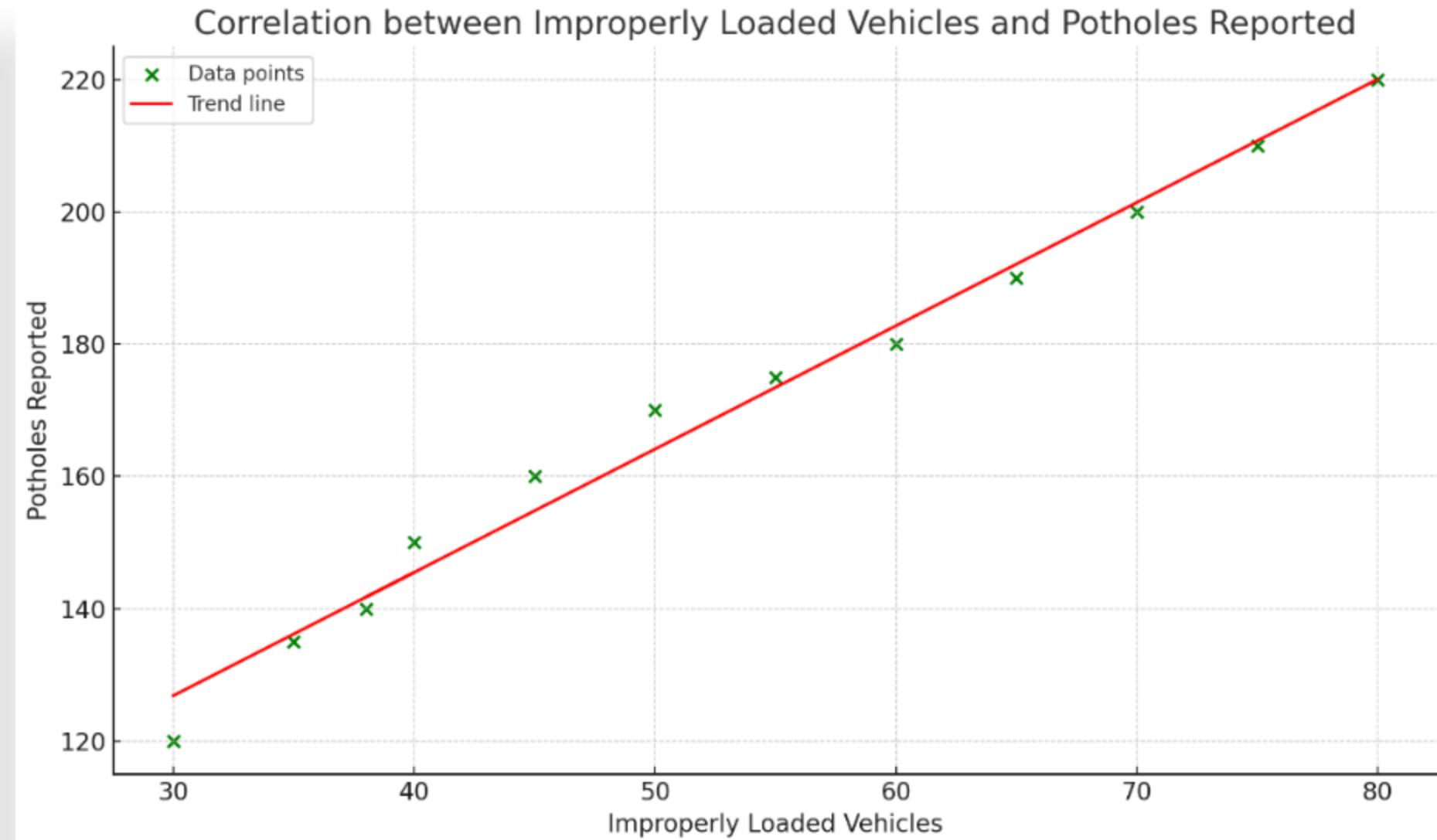
대형 교통사고의 주범 = 과적 차량

02 배경 및 목표



포트홀과 과적 차량 사이의 높은 상관관계

상관계수 : 0.996



포트홀과 적재 불량 차량 사이의 높은 상관관계

상관계수 : 0.993

02 배경 및 목표

2-3. 도로유지관리 인력의 부족

그래 록하드인가 뭔가로 일단은 떼우면 된다면서
열심히 떼워야할 것 아니야

일단 우리 지자체 같은 경우는
도로를 담당하는 일반직 공무원이 구별로 1명이야
1명이 도로 전화민원접수, 국민신문고 처리, 영조물 보상 접수, 도로공사 설계, 도로공사 감독을 하고 있어

보통은 전화든 국민신문고든 포트홀 민원이 접수되면
도로보수원이라고 불리는 공무직 분들에게 보수를 지시해
가끔 가다보면 포터에 화살경광등켜고 형광안전복입고 도로보수작업하시는 분들이야

도로보수원은 2~3명이 1개조인데(운전/교통통제/도로보수)
이 분들이 많지가 않아

차량1대로 1개조가 2개의 구를 커버하고 있어
민원대비 담당면적이 넓기때문에

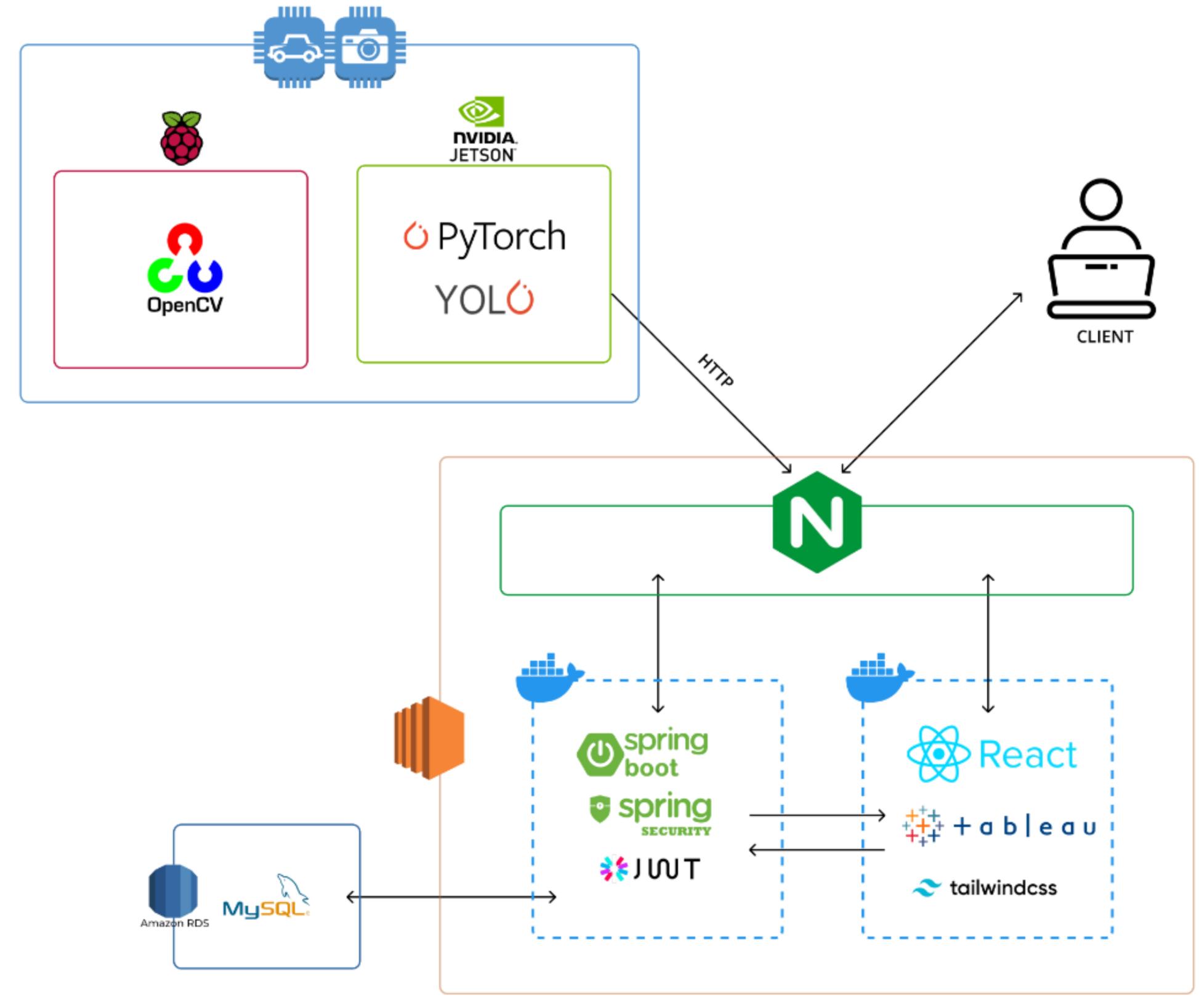
두번째, 하루에 보수할 수 있는 포트홀의 갯수가 많지 않아

도로보수원이 더블캡(포터)에 긴급보수재를 25kg짜리 30포정도씩을 싣고 나가
근데 30cm정도되는 작은 포트홀 하나에 한 포대정도 들어가고 큰건 8~10포대도 들어가
결국 5~10곳정도 보수하면 록하드 싣으러 다시 사무실에 들어와야해
그럼 점심시간이고
점심먹고 나가서 그정도 보수하면 퇴근시간인거지

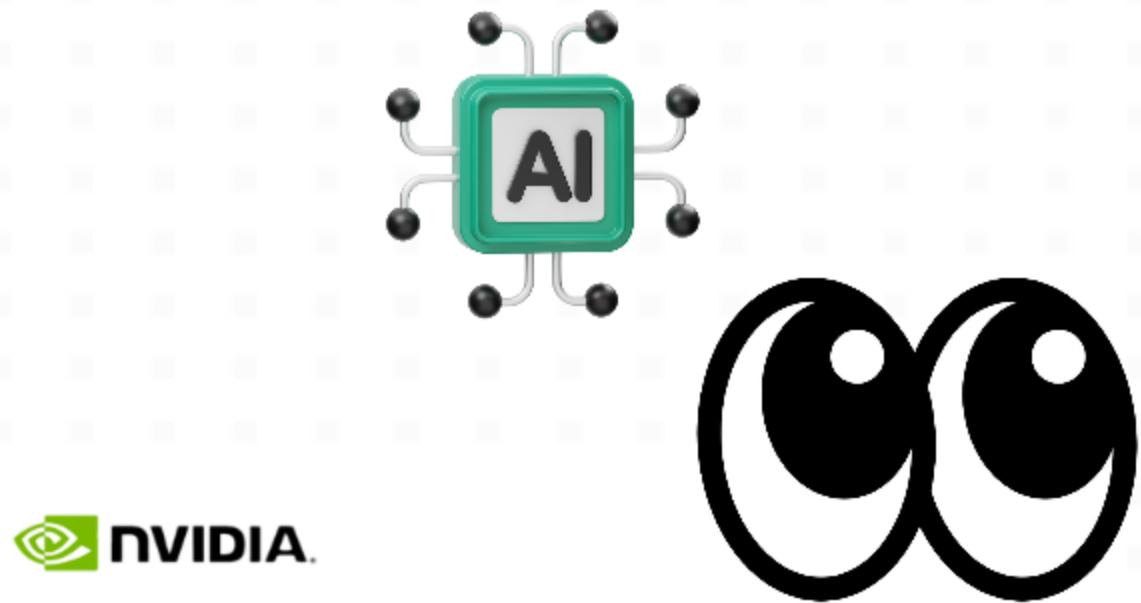
그러면, 일반직 공무원 담당자는 애가 타
민원답변기한이 있으니깐
도로보수원에게 보수됐냐고 물어보고, 안됐으면 빨리 부탁좀 드린다고 하고

그러다 답답하니 결국 보물찾기 하러 나가면서, 록하드를 싣고 나가
더블캡이 아니니 10포정도 뿐이 못싣고 나가는거지

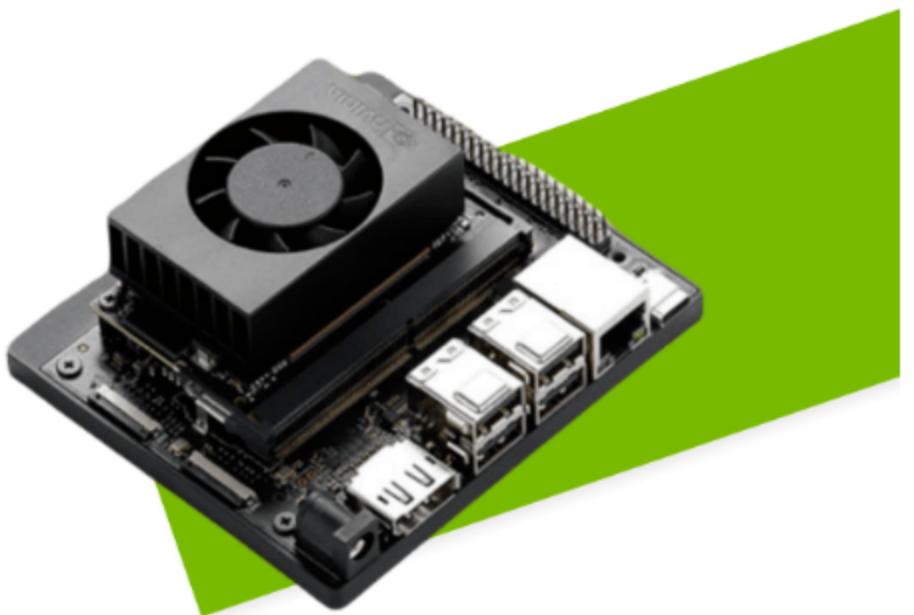
03 데이터 구조



03 데이터 흐름 (임베디드)



**Jetson Orin Nano
Dev-Kit**

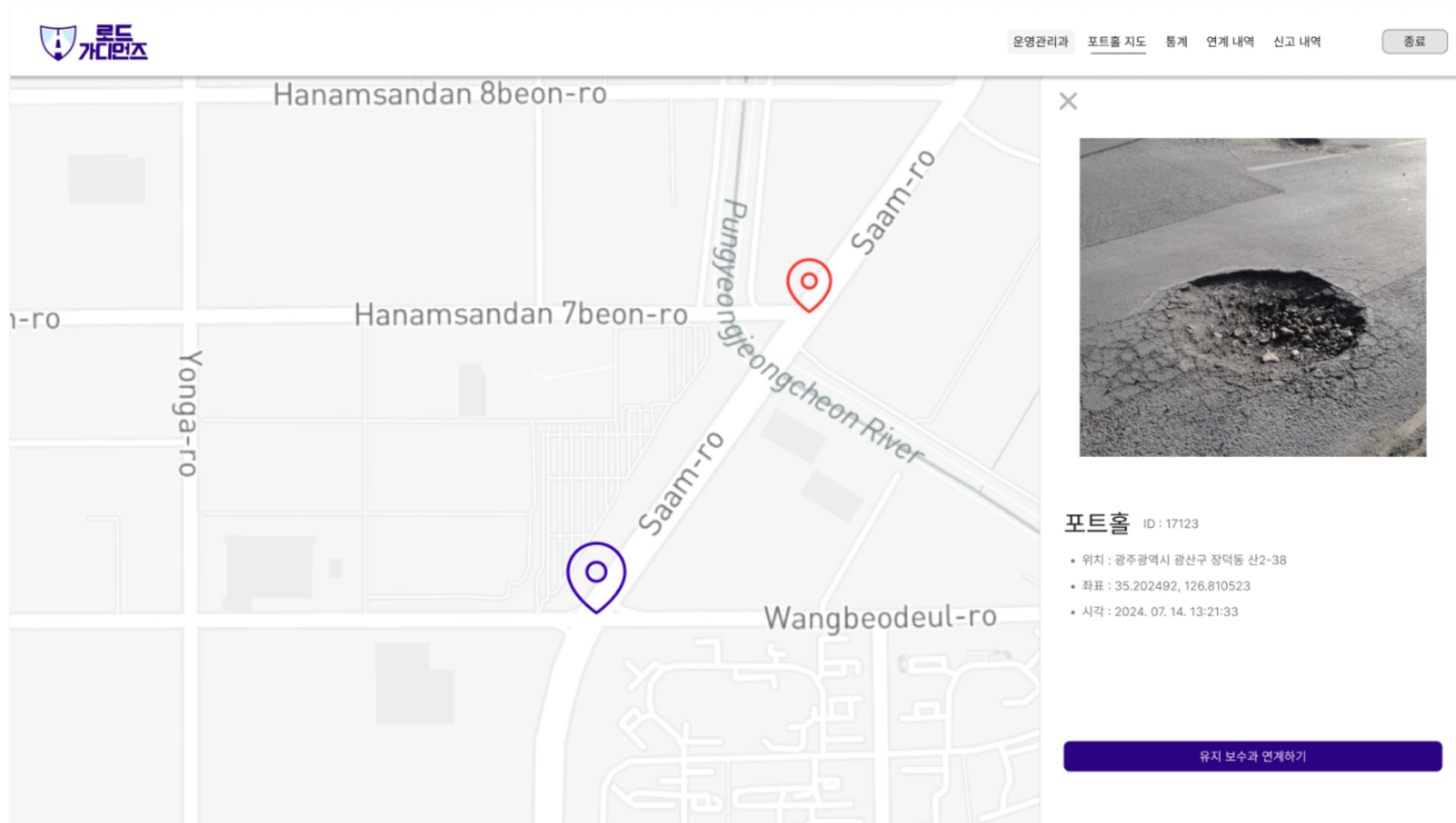


Object Detection



Amazon EC2

03 데이터 흐름 (WEB - 운영관리팀)



포트홀 지도 => 상세 정보 확인 + 유지부서와 연계 가능

03 데이터 흐름 (WEB - 운영관리팀)

로드
가디언즈

운영관리
포트홀 지도
통계
연계 내역
신고 내역
종료

2024.07.14
신고 전 과적 차량

신고 전 신고 후

ID : 12312
위치 : 광주광역
좌표 : 35.202
시각 : 2024. 07. 14. 13:21:33
번호판 : 123도

ID : 12312
위치 : 광주광역
좌표 : 35.502
시각 : 2024. 07. 14. 13:21:33
번호판 : 156다

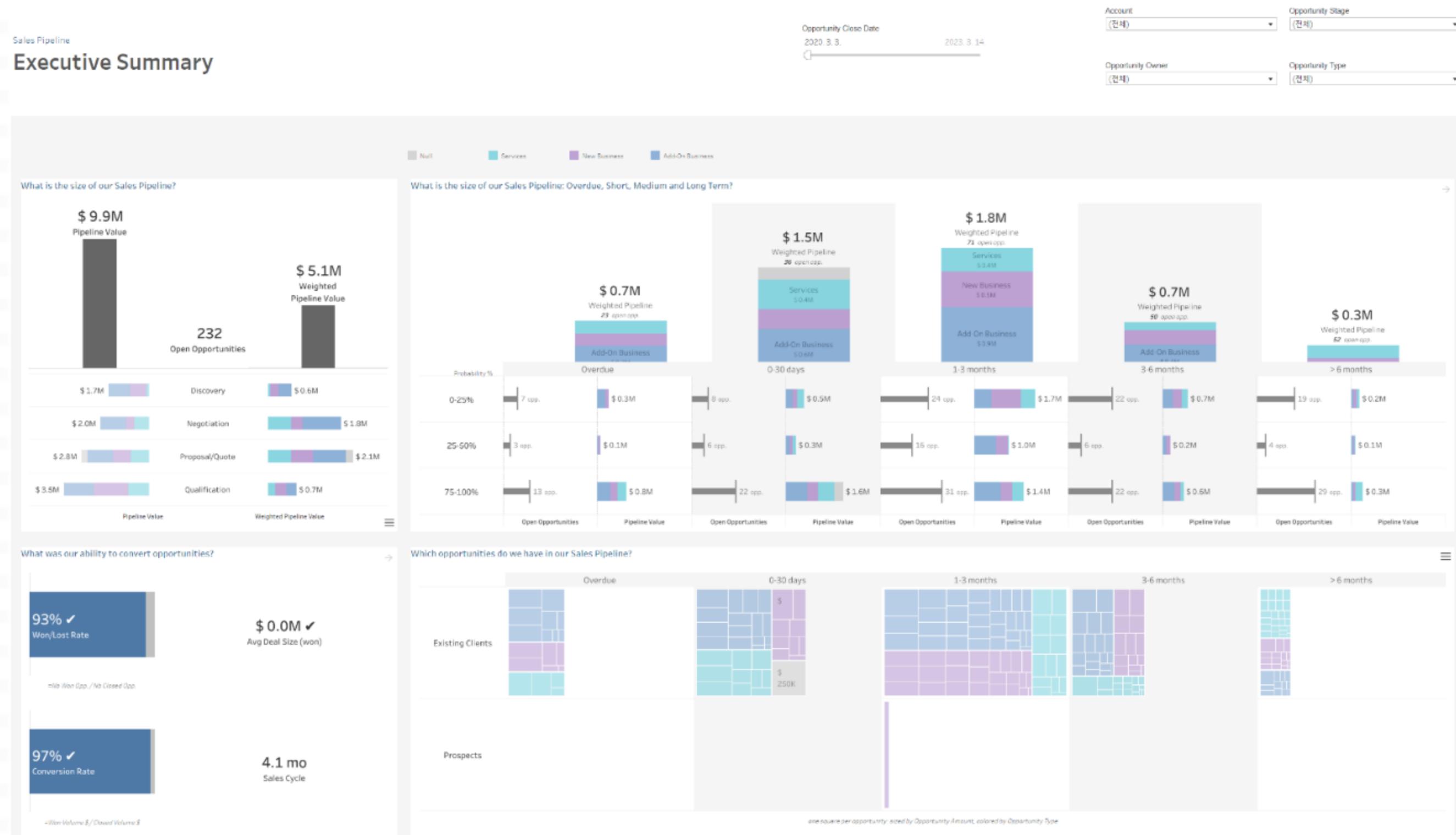
ID : 123123 / 높이 불량 123123
위치 : 광주광역시 광산구 장덕동 산2-38
좌표 : 35.202492, 126.810523
시각 : 2024. 07. 14. 13:21:33
번호판 : 123도 4567

ID : 12312
위치 : 광
좌표 : 35
시각 : 20
번호판 : 452도 9995

신고하기

신고 내역 中 신고 전 => 적발 과적 차량 신고 가능

03 데이터 흐름 (WEB - 운영관리팀)



통계 (과적 차량, 포트홀 상관성 분석 + 구별 포트홀 분포)

워드 문서로 구성한 대시보드입니다.

03 데이터 흐름 (WEB - 유지보수팀)

로드
가드ianz

유지보수 1팀 | 작업 정보 | 작업 지도 | 종료

작업 현황

- 보수 작업 11321
127.123456, 36.123456
- 보수 작업 22422
127.125555, 36.125555
- 보수 작업 61823
127.123456, 36.543210
- 보수 작업 45431
127.543210, 36.123456
- 보수 작업 23451
128.123456, 36.123456

Gwangju

HEUKSEOK-DONG

DUAM-DONG

WOLSAN-DONG

RKJJ

성촌마을

녹동마을

선택한 포트홀을 기준으로 작업 경로 설정 가능

05 진행 상황



AI

- AI 학습을 위한 라벨링 데이터 확인
- 소규모 라벨링 데이터로 테스트 학습 및 모델 테스트 확인

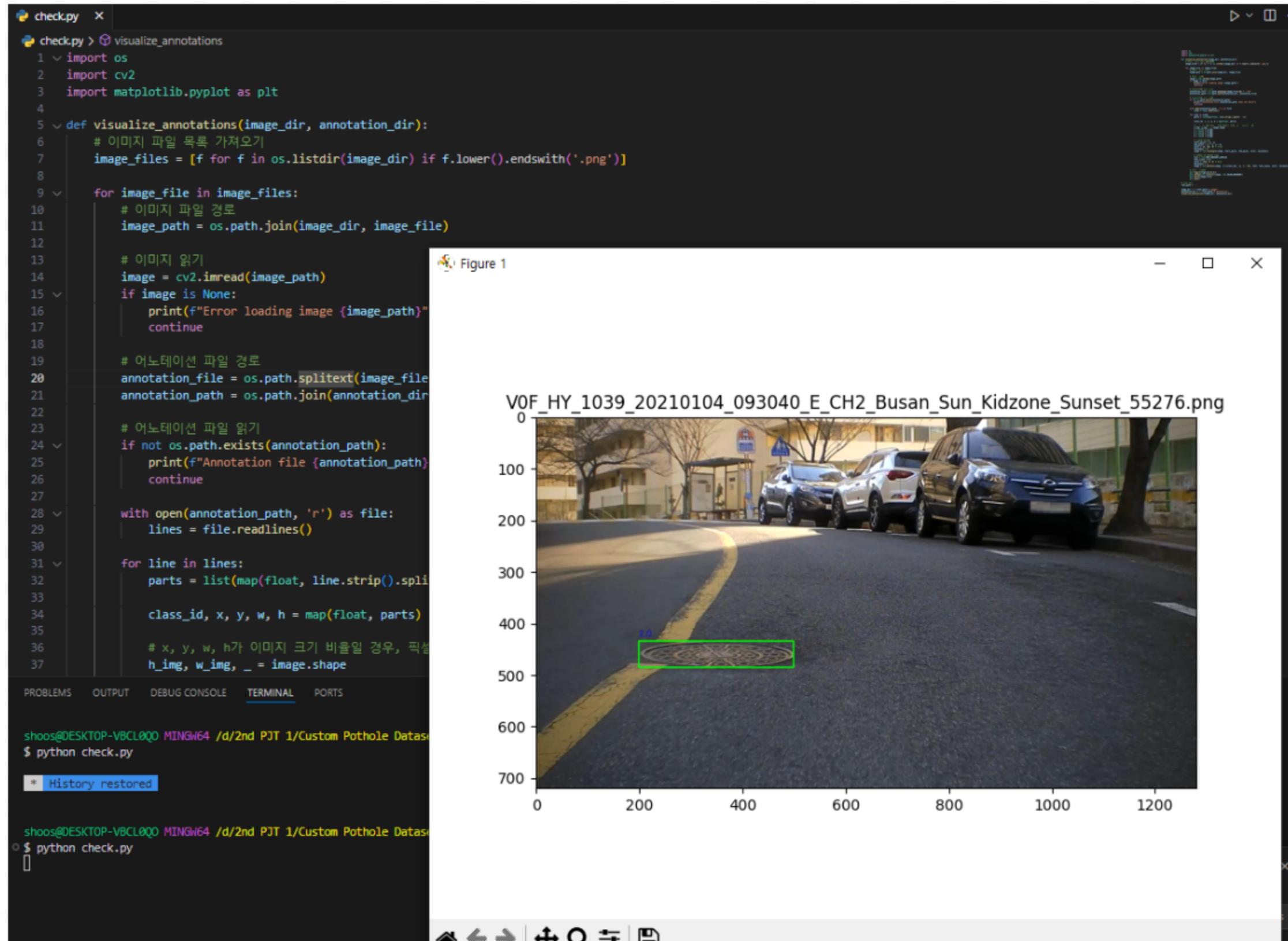
The screenshot shows a code editor window with Python code named 'check.py'. The code reads annotation files from a directory, checks if corresponding image files exist, loads the images, and processes their annotations. A terminal window below shows a list of file paths. To the right is a photograph of a paved road with a green bounding box drawn around a specific area, labeled 'Draw 2 00001'.

```
check.py x
check.py > process_annotations
1 import os
2 import cv2
3 import matplotlib.pyplot as plt
4
5 def process_annotations(annotation_dir, image_dir):
6     annotation_files = [f for f in os.listdir(annotation_dir) if f.endswith('.txt')]
7
8     for file in annotation_files:
9         with open(os.path.join(annotation_dir, file), 'r') as f:
10             lines = f.readlines()
11
12             image_name = os.path.splitext(file)[0] + '.png'
13             image_path = os.path.join(image_dir, image_name)
14
15             # Check if image file exists
16             if not os.path.exists(image_path):
17                 print(f"Image file '{image_name}' does not exist. Skipping...")
18                 continue
19
20             img = cv2.imread(image_path)
21
22             if img is None:
23                 print(f"Failed to load image '{image_name}'")
24                 continue
25
26             height, width, _ = img.shape
27
28             temp = []
29             for line in lines:
30                 label = list(map(float, line.strip().split()))
31                 class_id = int(label[0])
32                 x = float(label[1]) * width
33                 y = float(label[2]) * height
34                 w = float(label[3]) * width
35                 h = float(label[4]) * height
36
37                 x1 = int(x)
```

V4F_HY_1077_20210126_145052_N_CH1_Busan_Rain_Mainroad_Day_456
V4F_HY_1083_20201230_102724_E_CH1_Seoul_Sun_Mainroad_Day_93362
V4F_HY_1093_20210126_144452_N_CH2_Busan_Rain_Mainroad_Day_1075
V4F_HY_1095_20210115_122338_E_CH0_Seoul_Sun_Mainroad_Day_85058
V4F_HY_1095_20210126_142715_N_CH1_Busan_Rain_Mainroad_Day_9026
V4F_HY_1110_20210114_115158_N_CH1_Seoul_Sun_Mainroad_Day_94052
V4F_HY_1110_20210126_142913_E_CH1_Busan_Rain_Mainroad_Day_9797
V4F_HY_1111_20210129_134330_N_CH2_Busan_Sun_Mainroad_Day_61977
V4F_HY_1113_20210120_164336_E_CH1_Seoul_Sun_Mainroad_Day_92513
V4F_HY_1114_20210114_114821_E_CH1_Seoul_Sun_Mainroad_Day_25664
V4F_HY_1127_20201007_140035_N_CH1_Seoul_Sun_Mainroad_Day_78603
V4F_HY_1127_20210126_144152_E_CH1_Busan_Rain_Mainroad_Day_2895
V4F_HY_1133_20210126_105105_E_CH1_Busan_Rain_Mainroad_Day_9112
V4F_HY_7015_20210119_100647_N_CH0_Busan_Sun_Kidzone_Day_42636.png

05 진행 상황

AI => 라벨링 커스터마이징



The screenshot shows a Python code editor with a script named `check.py`. The code is used to visualize annotations for images. It imports os, cv2, and matplotlib.pyplot. The `visualize_annotations` function takes two parameters: `image_dir` and `annotation_dir`. It lists all image files in `image_dir` and reads each image. It then checks if the corresponding annotation file exists. If it does, it reads the annotation file, which contains class ID, x, y, width, and height values. These values are used to draw a bounding box around objects in the image. The image shown is a street scene with several cars parked on the side. A pothole on the road surface is highlighted with a green bounding box.

```
checkpy x
checkpy > ⌂ visualize_annotations
  1 import os
  2 import cv2
  3 import matplotlib.pyplot as plt
  4
  5 def visualize_annotations(image_dir, annotation_dir):
  6     # 이미지 파일 목록 가져오기
  7     image_files = [f for f in os.listdir(image_dir) if f.lower().endswith('.png')]
  8
  9     for image_file in image_files:
 10         # 이미지 파일 경로
 11         image_path = os.path.join(image_dir, image_file)
 12
 13         # 이미지 읽기
 14         image = cv2.imread(image_path)
 15         if image is None:
 16             print(f"Error loading image {image_path}")
 17             continue
 18
 19         # 어노테이션 파일 경로
 20         annotation_file = os.path.splitext(image_file)[0] + ".txt"
 21         annotation_path = os.path.join(annotation_dir, annotation_file)
 22
 23         # 어노테이션 파일 읽기
 24         if not os.path.exists(annotation_path):
 25             print(f"Annotation file {annotation_path} does not exist")
 26             continue
 27
 28         with open(annotation_path, 'r') as file:
 29             lines = file.readlines()
 30
 31         for line in lines:
 32             parts = list(map(float, line.strip().split()))
 33
 34             class_id, x, y, w, h = map(float, parts)
 35
 36             # x, y, w, h가 이미지 크기 비율일 경우, 픽셀
 37             h_img, w_img, _ = image.shape
 38
 39             # 이미지에 bounding box 그리기
 40             cv2.rectangle(image, (int(x), int(y)), (int(x+w), int(y+h)), (0, 255, 0), 2)
 41
 42             # 이미지에 클래스 ID 표시
 43             cv2.putText(image, str(class_id), (int(x), int(y)), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
 44
 45     # 이미지 출력
 46     plt.imshow(image)
 47     plt.title(annotation_file)
 48     plt.show()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
shoos@DESKTOP-V8CL0QQ MINGW64 /d/2nd PJT 1/Custom Pothole Dataset
$ python check.py
* History restored

shoos@DESKTOP-V8CL0QQ MINGW64 /d/2nd PJT 1/Custom Pothole Dataset
$ python check.py
```

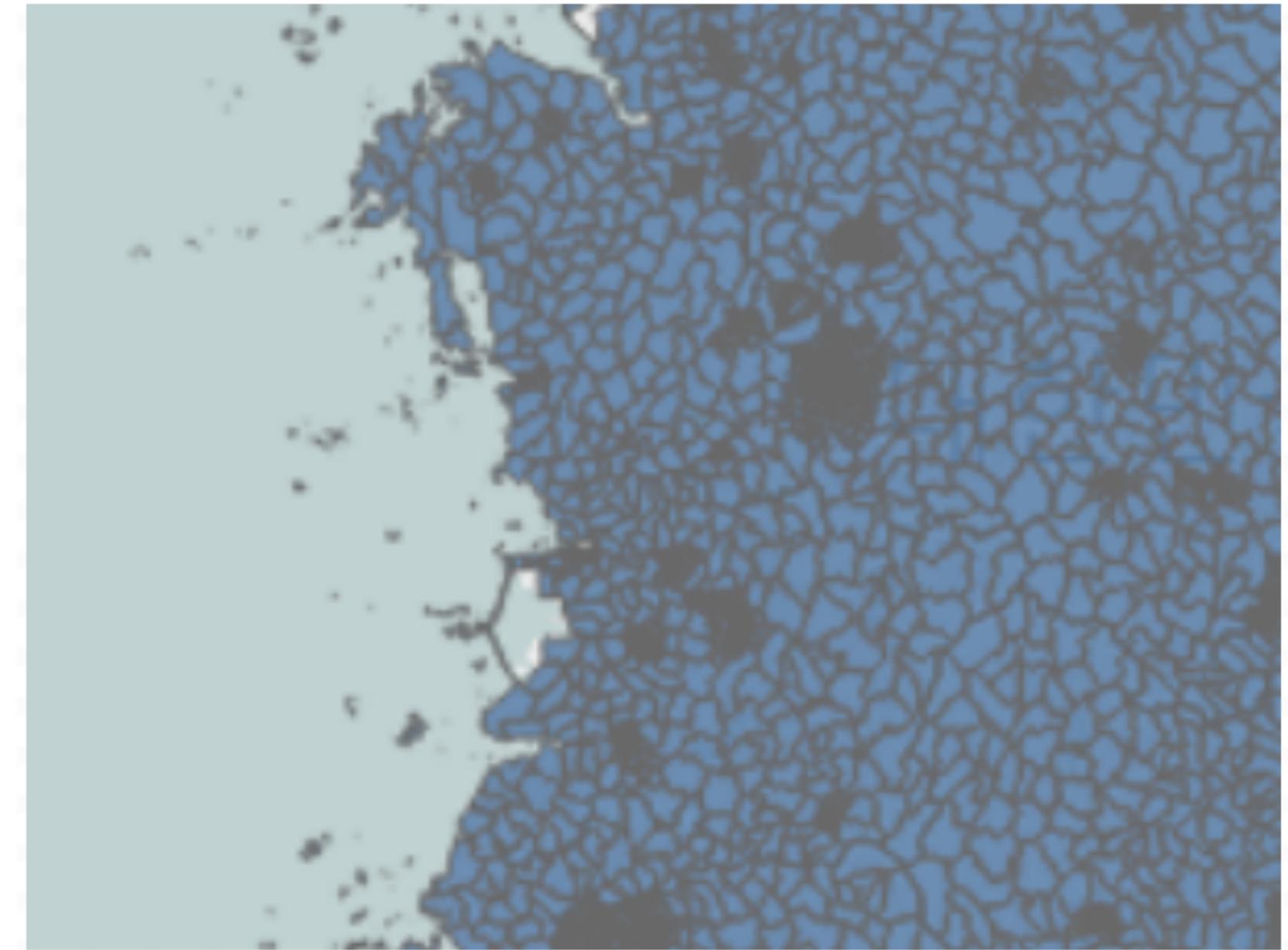
AI-HUB의 데이터셋
활용을 위한
라벨링 커스터마이징

05 진행 상황



통계

- 시각화 대시보드 프론트로 불러오기
- 기하도형 지도 구현



05 진행 상황

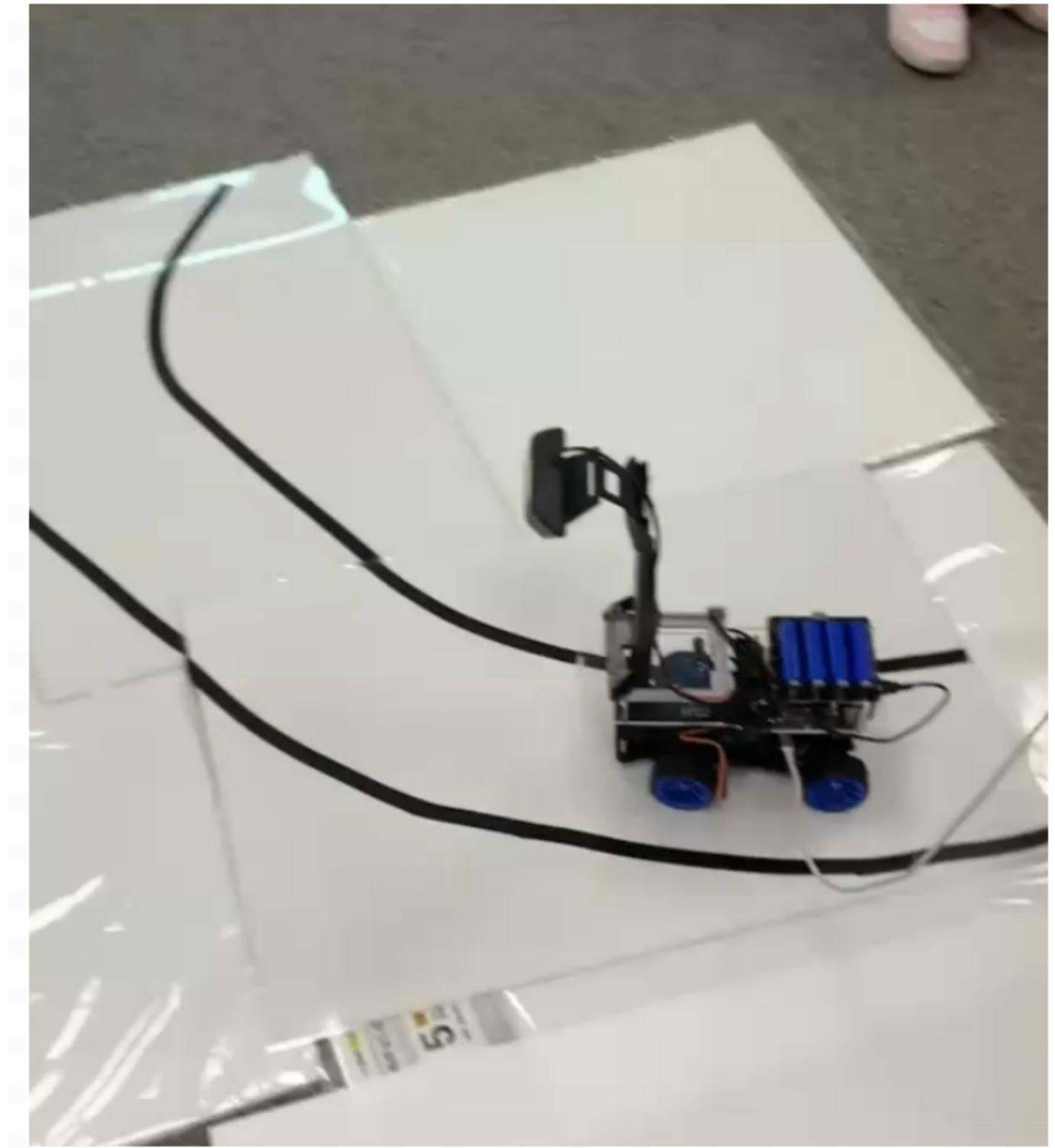
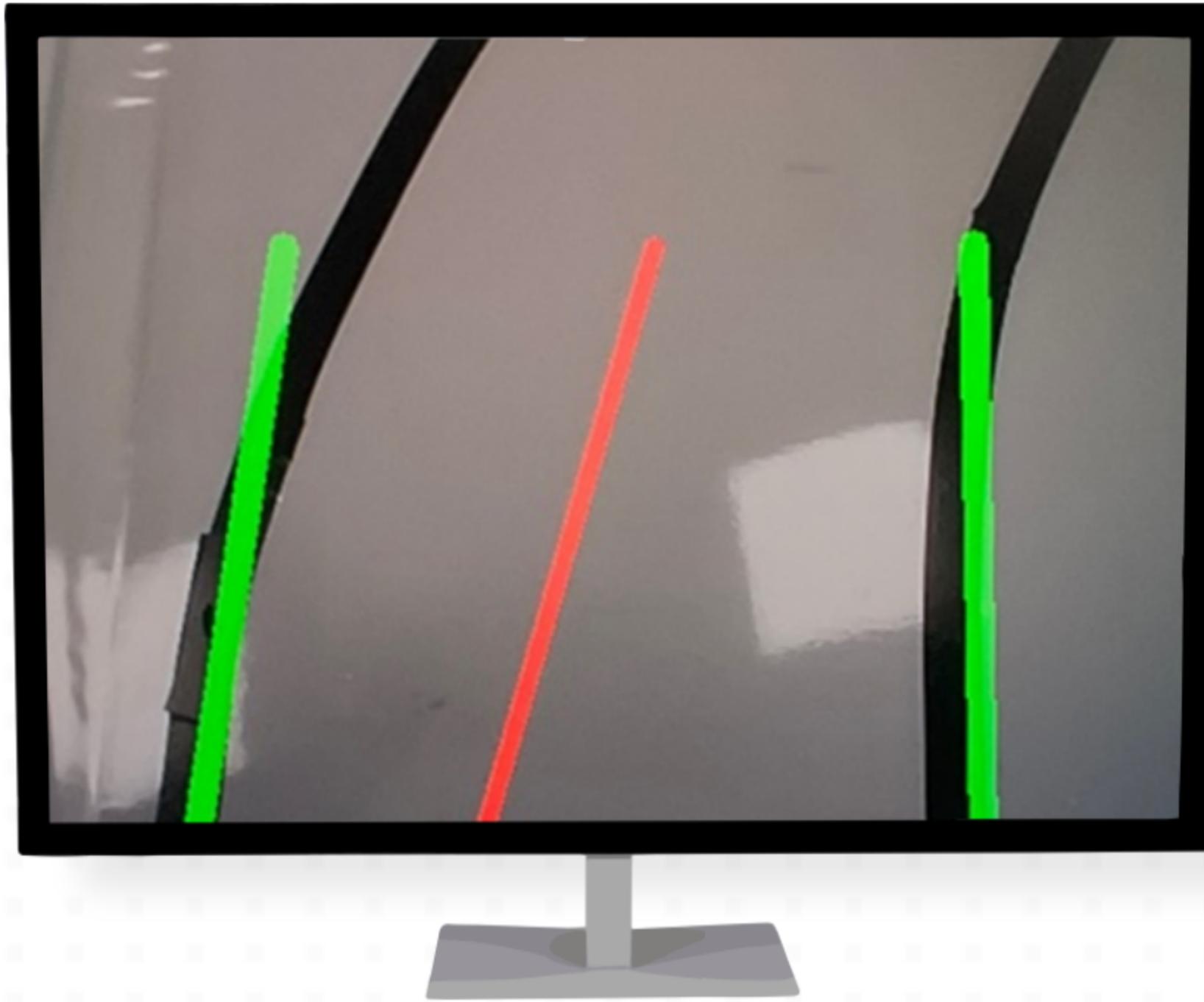


임베디드

- RC CAR 조립 완성
- 라인 디택션을 이용한 자율주행 구현 중

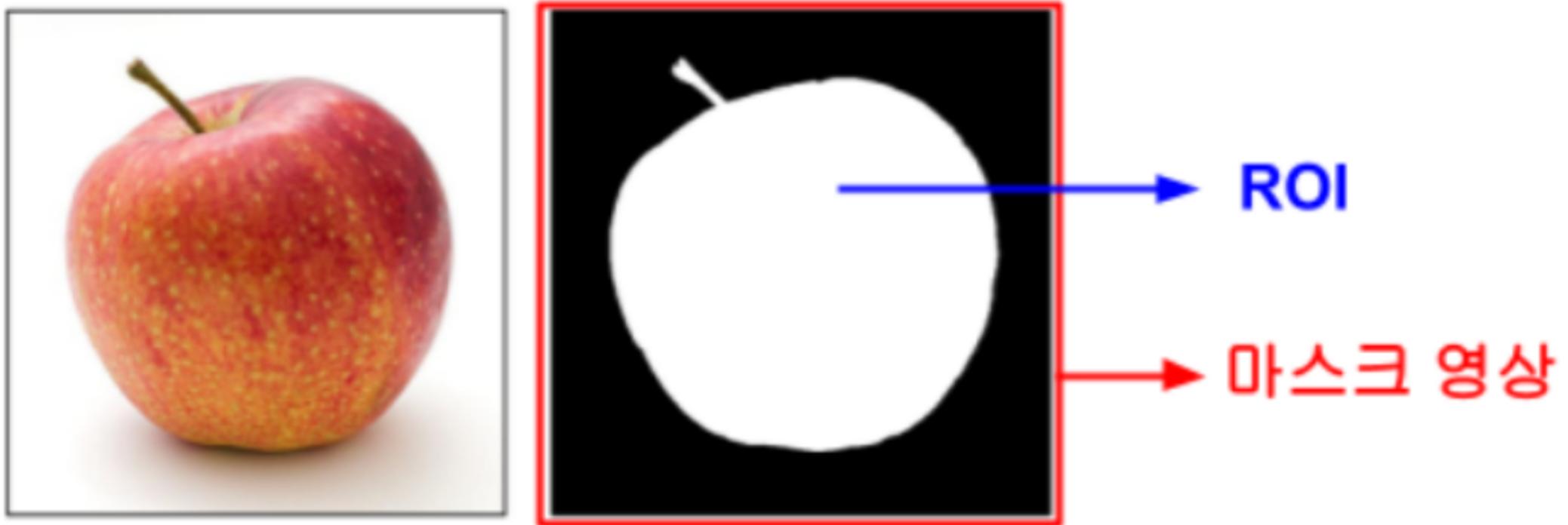


05 진행 상황



05 진행 상황

임베디드 HW =>

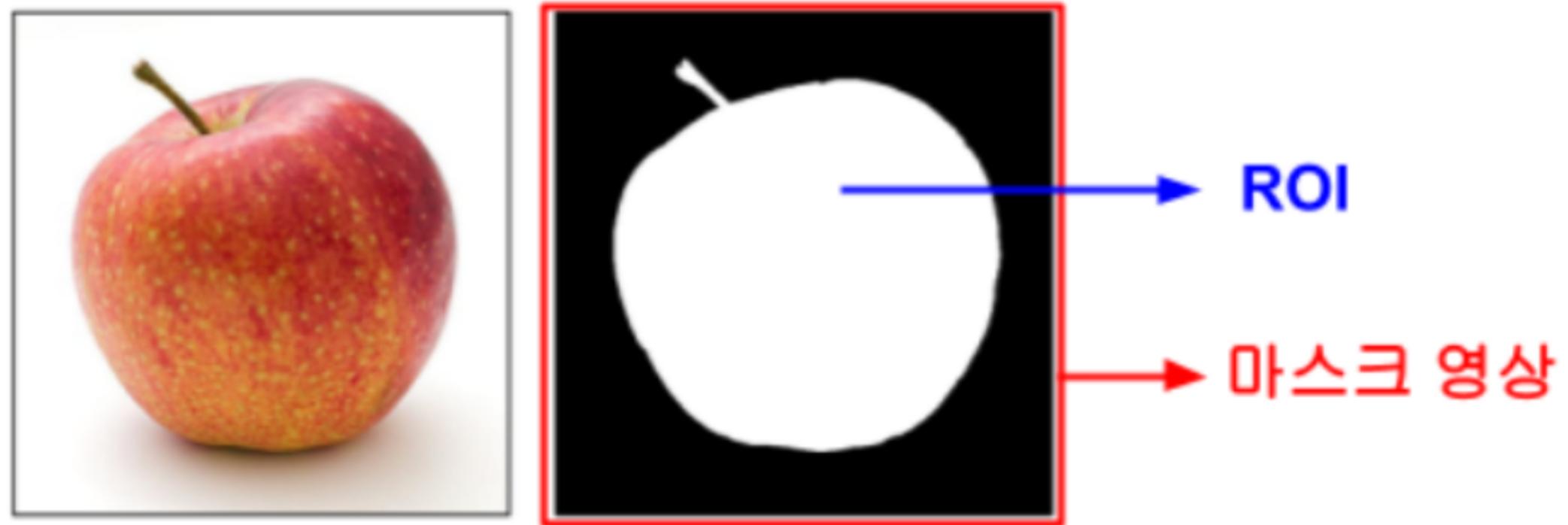


=> ROI (Region Of Interest) 기법 이용

=> 이미지를 관심 영역만 잘라
차선만 감지할 수 있도록 구현

05 진행 상황

임베디드 SW => 차선 감지 기능 오류



=> ROI (Region Of Interest) 기법 이용

=> 이미지를 관심 영역만 잘라 차선만 감지할 수 있도록 구현

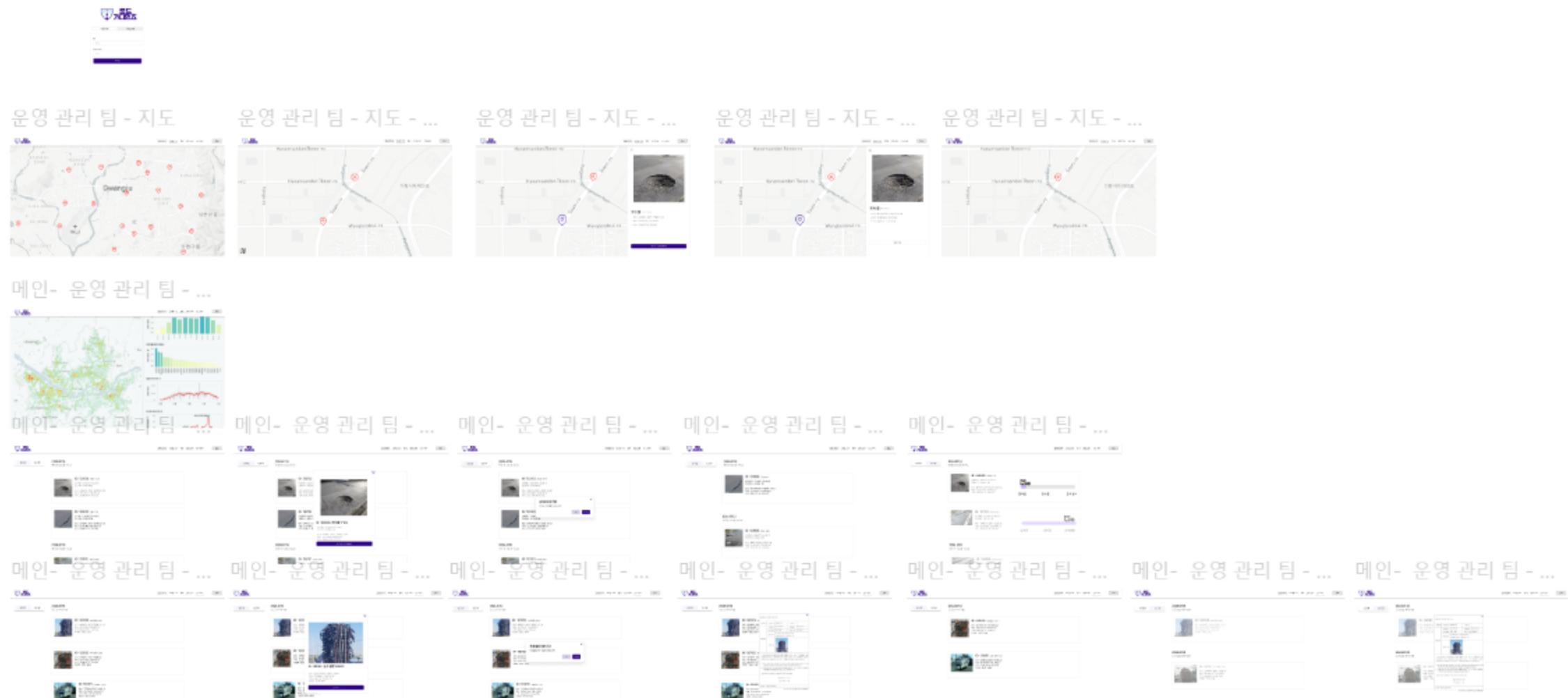
05 진행 상황



WEB

- Figma 도안 최종 완성
 - 기능별 필요한 스켈레톤
코드 구축 진행

운영관리과 로그인

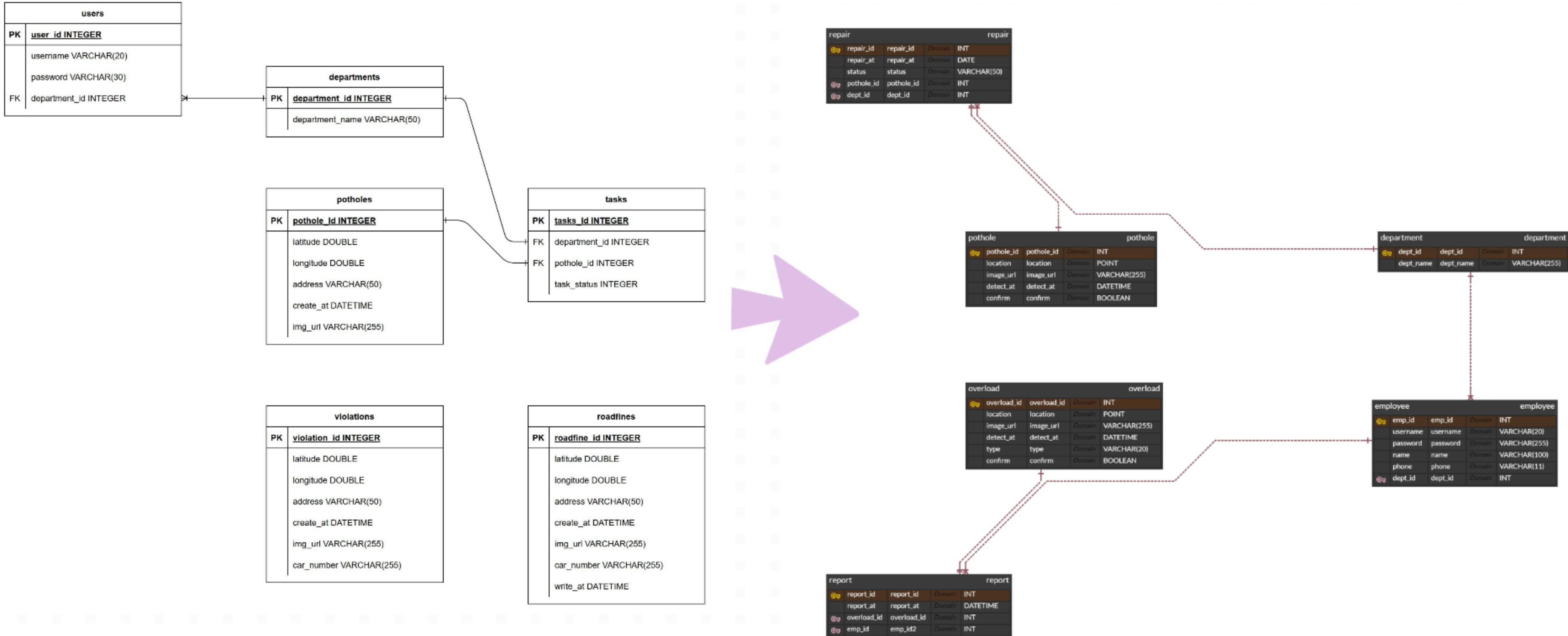


유지보수과 로그인



05 진행 상황

WEB => ERD 수정



05 진행 상황

하드웨어 --- WEB

ngrok

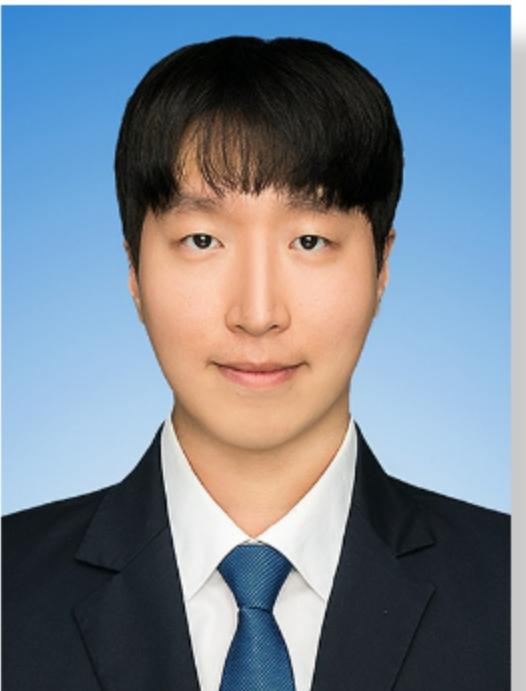
```
C:\Users\SSAFY\Desktop\ngrok\ngrok.exe - ngrok http 8080
ngrok
(Ctrl+C to quit)

Try our new Traffic Inspector: https://ngrok.com/r/ti

Session Status          online
Account                  souffle1903@gmail.com (Plan: Free)
Version                 3.13.0
Region                  Japan (jp)
Latency                44ms
Web Interface           http://127.0.0.1:4040
Forwarding              https://82c2-61-80-142-239.ngrok-free.app -> http://localhost:8080

Connections             ttl     opn      rt1      rt5      p50      p90
                        0       0       0.00    0.00    0.00    0.00
```

04 역할 분배



박민철

FRONT-END
AI



정우영

임베디드 (SW)
AI, BACK-END 보조



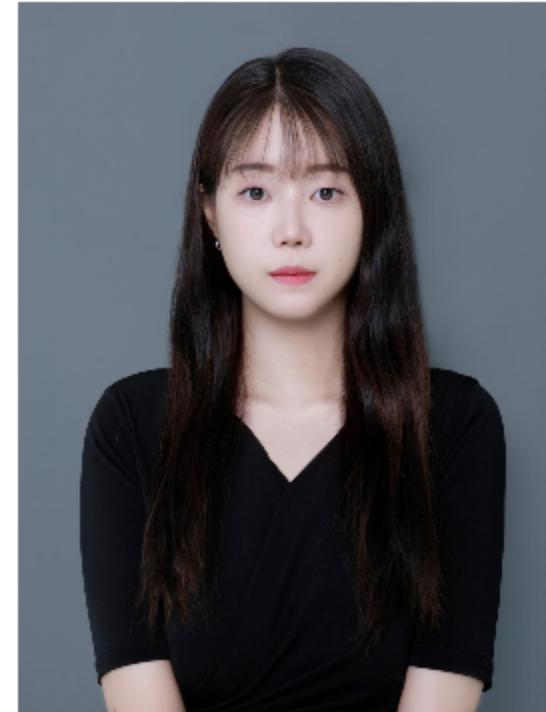
박건국(팀장)

임베디드 (HW)
서버, FRONT-END 보조



정하림

BACK-END
서버



지민경

FRONT-END
데이터분석, AI 보조

Q & A



Thank You

C104 - 박건국, 박민철, 정우영, 정하림, 지민경



로드 가 디 언 즈