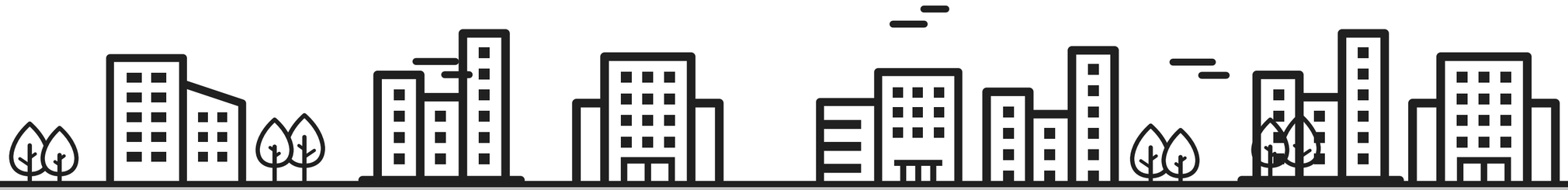
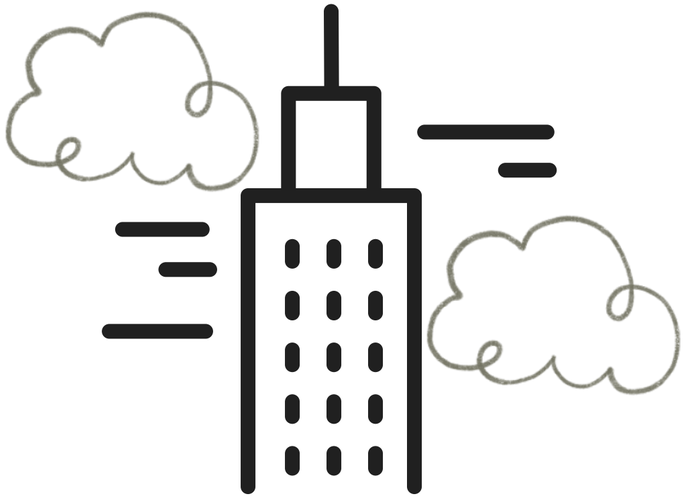
LEE HYEON JU

도로 안전과 유지보수 효율성 향상

YOLO 기반 Pothole 감지 시스템



목차



개발 목적:
포트홀 위험성

'도로 위 지뢰'로 불리는 포트홀의 위험 및 현황

운전자 안전 위협

- 차량 파손 및 사고 유발
- 급제동으로 인한 2차 사고 위험
- 야간 주행 시 식별 어려움

지역별 분포

- 서울·경기권 20%
- 수도권 집중
- 교통량 영향

노선별 현황

- 중앙고속도로 최다
- 전체의 14.7%
- 노선 특성 고려

현행 신고의 한계

- 수동 신고 방식의 비효율성
- 위치 파악의 부정확성
- 신속한 대응 어려움

발생 건수

- 22,000여 건
- 5년간 누적 (2020~2024)
- 연평균 4,400건

배상액 규모

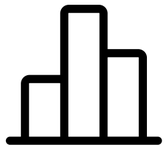
- 총 136억 원
- 연평균 27.2억 원
- 지속적 증가 추세

사용 모델
YOLO

YOLO 기반 포트홀 감지 모델 소개

YOLO 알고리즘

- 단일 신경망 처리
- 빠른 객체 탐지
- 높은 정확도
- 효율적 학습
- 다양한 버전



포트홀 감지 적용

- 이미지 분할
- 특징 추출
- 포트홀 식별

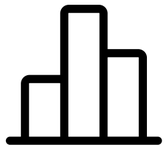
실시간 탐지 능력

- 고속 처리
- 다중 객체 인식
- 주행 중 감지

사용데이터

Kaggle

- 이미지 데이터셋 681장
normal 352 / potholes 329중
랜덤으로 244장의 이미지 사용



Roboflow

- anootate (주석)
- 데이터 증강

Yolo detection

- 실시간 탐지 기능
- yolov8 사용

출처 : https://www.kaggle.com/datasets/atulyakumar98/pothole-detection-dataset

성과 및 성능지표

Model	yolov8n		
epochs	10	50	90 (9h)
Precision(정밀도)	0.508	0.888	0.864
R(recall 재현율)	0.143	0.395	0.77
mAP50 (mean Average Precision @ IoU =0.5)	0.403	0.529	0.529
mAP50-95	0.191	0.248	0.055
result	정밀도와 mAP50이 낮아서 에포크를 높이기로 결정	정밀도는 크게 개선됐으나, 재현율이 낮아 탐지되지 않은 객체가 많음	정밀도와 mAP50이 높은 수준에 도달했으나, 재현율이 여전히 낮아 실환경 적용 시 탐지 성능 보완이 필요

구현 · 코드


```
#yolo8을 위한 라이브러리설치

!pip install ultralytics

# 데이터셋 존재여부확인
import os
!ls {dataset.location}/

#yolo 학습

from ultralytics import YOLO

#모델생성(사전학습)_nsml변경가능
model = YOLO('yolov8n.pt')

#학습시작
model.train(data='/content/pothole-2/data.yaml', epochs=70, imgsz=640)
```

YOLO N,S,M,L Model을 각각 사용해보았으나
컴퓨터 환경으로인해 n-model로만 진행

학습을 통해 모델링한 결과,
작은 pothole도 detection이 잘 되는 결과로
시각화 가능.

```
import cv2
from ultralytics import YOLO
from IPython.display import display, Image
import tempfile

# YOLO 모델+best.pt가져오기
model = YOLO('/content/best.pt')

# 비디오 파일 경로 (내가 테스트해보고 싶은 영상)
video_path = '/content/TEST3.mp4'

# 비디오 읽기
cap = cv2.VideoCapture(video_path)
fps = int(cap.get(cv2.CAP_PROP_FPS))
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))

# output video 파일명+경로설정
output_path = '/content/output_video.mp4'
fourcc = cv2.VideoWriter_fourcc(*'mp4v')
out = cv2.VideoWriter(output_path, fourcc, fps, (width, height))

while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break

    # YOLO로 프레임 처리
    results = model.predict(source=frame, save=False, show=False)
    annotated_frame = results[0].plot()  # 결과가 그려진 프레임

    # 결과를 출력 비디오에 쓰기
    out.write(annotated_frame)
```

기대효과

YOLO 기반 포트홀 감지 시스템의 장점

배상액 | 연간 배상액 감소 예상

유지비용 | 유지보수 비용 절감

투자효과 | 장기적 도로 관리 비용 절감

간접비용 | 교통 정체 및 사고 비용 감소

부가가치 | 스마트시티 연계 새로운 가치 창출

사고 예방

- 조기 포트홀 감지로 사고 위험 감소
- 실시간 모니터링으로 신속한 대응
- 도로 관리 신뢰도 제고

안전성 향상

- 운전자 안전 확보
- 차량 파손 방지
- 2차 사고 예방 효과

발전과제

YOLO 기반 포트홀 감지 시스템의 경제성

모델 성능 | 더다양한 환경에서의 데이터셋 확보

기능 확장 | 블랙박스, 실시간 도로상태
모니터링 시스템과 통합

시장성 | 정부/지자체와 협업
블랙박스 제조사와 파트너십을 맺어
자동신고 기능 상용화

기술적 통합 | 탐지된 포트홀 위치를 도로보수팀의
GPS 디바이스와 연동하여 자동작업배정