

YOLOv11 기반 자동 포트홀 신고 및 시각화 시스템 구현

유동주*¹, 이석훈*², 정동원**

Implementation of YOLOv11-based Automated Pothole Reporting and Visualization System

Dongju Yu*¹, Sukhoon Lee*², and Dongwon Jeong**

요 약

현대 사회에서 도로 상태는 교통 안전과 경제적 효율성에 중요한 영향을 미친다. 그러나 차량 증가와 기후 변화로 인해 발생하는 도로 손상, 특히 포트홀 문제는 운전자와 사회 전반에 심각한 피해를 초래한다. 본 논문에서는 이러한 문제를 해결하기 위해 최신 검출 기술인 YOLOv11 객체 탐지 모델을 활용해 포트홀을 자동으로 감지하고, 실시간으로 신고 및 시각화하는 시스템을 제안한다. 제안된 시스템은 스마트폰 애플리케이션과 웹 페이지로 구성되며, 차량 주행 영상을 활용해 포트홀 정보를 효율적으로 수집하고 이를 웹에서 시각화하여 도로 관리의 효율성을 향상시킨다. 이를 통해 도로 유지보수 비용 절감과 교통 안전 개선에 기여할 수 있을 것으로 기대된다.

Abstract

Road conditions are vital for ensuring traffic safety and maintaining economic efficiency in modern society. However, increasing vehicle numbers and the impacts of climate change have caused significant road damage, with potholes being a particularly critical issue that poses serious risks to drivers and society alike. To resolve the problem, this paper proposes a system that utilizes advanced technology(YOLOv11 object detection model) to automate pothole detection and enable real-time reporting and visualization. The proposed system in this paper combines a smartphone application with a web platform, leveraging footage captured during vehicle operation to efficiently collect and display pothole data online. This approach greatly enhances road management efficiency and is expected to reduce maintenance costs while improving traffic safety.

Keywords

object detection, pothole detection, smartphone application, road maintenance

* 국립군산대학교 소프트웨어학과
- ORCID¹: <https://orcid.org/0009-0001-6333-1136>
- ORCID²: <https://orcid.org/0000-0002-3390-5602>
** 국립군산대학교 소프트웨어학과 교수(교신저자)
- ORCID: <https://orcid.org/0000-0001-9881-5336>

· Received: Feb. 04, 2025, Revised: Feb. 20, 2025, Accepted: Feb. 23, 2025
· Corresponding Author: Dongwon Jeong
School of Software, Kunsan National University, 558, Daehak-ro, Gunsan, Jeonbuk-do, Republic of Korea
Tel.: +82-63-469-8911, Email: djeong@kunsan.ac.kr

I. 서 론

현대 사회에서 도로는 경제와 일상생활에서 핵심 인프라로 기능한다. 그러나 차량 수 증가와 폭설, 폭우와 같은 극단적 기후 현상 등 복합적 요인으로 도로 상태가 악화되고 있다. 이러한 요인들이 결합하여 발생하는 도로 파손의 대표적 사례는 바로 포트홀(Pothole)이다[1].

포트홀은 도로 표면의 아스팔트가 손상되어 생기는 항아리 모양의 크고 작은 함몰을 의미하며, 이는 도로의 미관을 해칠 뿐만 아니라 도로 이용자에게도 직접적인 피해를 준다. 포트홀로 인해 운전자의 주행 안정성이 저해되며, 심각한 경우 타이어 파손이나 조향 장치 고장 등으로 교통사고의 주요 원인이 된다. 실제로 포트홀로 인한 사고 및 차량 손상 건수는 매년 증가 추세를 보이며, 이는 개인의 안전 문제를 넘어 사회적, 경제적 손실로 이어진다[2].

이에 따른 포트홀 문제에 대처하기 위한 도로 유지보수 비용 또한 지속적으로 상승하고 있어, 국가 및 지방 정부의 재정적 부담이 커지고 있다. 따라서 포트홀의 조기 발견과 신속한 보수는 교통 안전을 높이고 경제적 손실을 줄이기 위한 중요한 과제로 대두된다.

기존의 포트홀 신고 방식은 운전자나 도로 이용자가 직접 신고하는 수동적인 방법에 의존하고 있어, 신고가 이루어지지 않거나 지연될 가능성이 크다. 이로 인해 실시간 도로 상태를 정확히 파악하거나 유지보수 계획을 수립하는 데 한계가 있다. 이러한 문제를 해결하기 위해 본 논문에서는 인공지능 기술을 활용하여 포트홀을 자동으로 검출하고 포트홀 지점을 신고할 수 있는 시스템을 제안한다. 여기서 신고란 포트홀을 검출하고 데이터베이스에 저장하는 연산을 의미한다. 제안 시스템을 통해 도로 관리자에게 실시간 도로 상태 정보를 제공하여 효율적인 유지보수 계획 수립에 기여하고자 한다.

본 논문에서는 최신 객체 검출 알고리즘인 YOLOv11[3]을 활용하여 포트홀을 검출하여 신고하는 애플리케이션과 검출된 포트홀 정보를 웹에서 실시간으로 지도상에 시각화하는 포트홀 관리 시스템을 제안한다.

본 논문은 구성은 다음과 같다. 먼저 제2장 관련

연구에서는 포트홀 검출을 위한 기존의 다양한 방법들을 분석하고, 각 방법의 특징과 한계를 평가한다. 제3장에서는 제안하는 포트홀 신고 시스템의 구체적인 구현 과정을 상세히 설명하며, 데이터 전처리, 딥러닝 모델 학습, 스마트폰 애플리케이션 개발, 포트홀 데이터 시각화 웹 페이지 구축 등을 논의한다. 마지막으로, 제4장에서는 연구의 주요 성과를 요약하고, 시스템의 성능 향상을 위한 향후 연구 방향과 개선 과제를 제안한다. 이러한 구성을 통해 포트홀 검출 시스템의 개발 및 구현 과정을 체계적으로 전달하고, 도로 관리의 효율성을 높이기 위한 실질적인 해결책을 제시한다.

II. 관련 연구

본 시스템의 핵심은 포트홀의 정확한 검출이다. 이를 위해 다양한 포트홀 검출 기법의 특징과 한계를 분석한다.

2.1 가속도 센서 기반 방법

가속도 센서를 활용한 포트홀 검출은 차량의 진동 데이터를 분석하여 포트홀의 위치와 심각도를 추정하는 방식이다[4]. 이 방법은 구현 비용이 낮고 설치가 간편하며, 실시간 데이터 수집이 가능하다는 장점을 지닌다. 예를 들어, 차량에 부착된 가속도 센서는 주행 중에 발생하는 진동 데이터를 분석하여 특정 위치에서의 진동 변화를 통해 포트홀을 감지할 수 있다.

그러나 가속도 센서는 도로 평탄도, 차량 속도, 서스펜션 특성 등 다양한 외부 요인에 영향을 받는다. 동일한 포트홀이라도 노면 상태와 차량 종류에 따라 진동 신호가 달라질 수 있어, 검출 정확도가 낮아지는 경우가 발생한다. 이러한 한계로, 가속도 센서 기반 방법은 높은 정밀도가 요구되는 상황에서는 제한적으로 활용된다.

2.2 라이다(LiDAR) 센서 기반 방법

라이다(LiDAR) 센서는 빛의 반사를 이용하여 물체의 거리와 형태를 측정하는 기술로, 이를 활용해

포트홀을 감지할 수 있다. 라이다 센서는 2D 및 3D 버전으로 분류되며, 특히 2D 라이다를 활용한 포트홀 검출 연구가 활발하게 진행되고 있다.

2D 라이다는 단일 평면에서 빛을 스캔하여 포트홀이 있는 구간에서 반사 신호 변화를 감지함으로써 포트홀을 식별한다. 예를 들어, 차량 전방에 2D 라이다를 설치하여 도로 표면의 패인 정도를 측정하고 측정된 데이터를 기반으로 포트홀을 검출하는 방식이다[5]. 라이다 센서는 높은 정밀도를 제공하는 반면, 하드웨어 비용이 높아 대규모로 적용하기 어렵다는 한계를 지닌다.

2.3 카메라 센서 기반 방법

카메라 센서를 이용한 포트홀 검출 방법은 도로 표면의 이미지를 획득하고, 이를 영상 처리 기술과 딥러닝 알고리즘으로 분석하는 방식이다[6-8]. 이 방법은 라이다 센서에 비해 상대적으로 저렴한 하드웨어로 구현 가능하며, 고해상도의 시각 정보를 통해 포트홀 유무를 파악할 수 있다는 장점을 지닌다. 최근에는 딥러닝 기술의 도입으로 검출 성능이 크게 향상되었다. 그러나 카메라 방식은 조명 조건과 날씨 변화에 민감해 야간이나 악천후 상황에서 성능이 저하될 수 있다. 또한, 이미지의 복잡성과 노이즈로 인해 실시간 처리에 높은 연산 자원이 요구될 수 있다.

2.4 검출 방법 비교 분석

기존 연구의 검출 방법에 대한 비교 분석한 결과를 토대로, 본 논문에서는 카메라 센서 기반 방법을 선택하였다. 카메라 센서는 가속도 센서보다 포트홀의 형태와 크기에 대한 상세한 정보를 제공하며, 라이다에 비해 경제적이면서도 충분한 검출 성능을 제공한다. 또한 딥러닝을 통해 지속적인 성능 향상이 가능하다는 점도 중요한 선택 이유이다.

카메라 센서 기반 방법은 다른 센서들과 비교해 접근성과 확장성이 뛰어나다는 장점을 지닌다. 특히 스마트폰에 내장된 카메라를 활용할 수 있어 시스템 구축에 용이하다. 또한, 인터넷 상에 풍부하게

존재하는 이미지 데이터를 활용할 수 있어 모델 학습에 유리하다.

따라서 본 논문에서는 이러한 장단점을 종합적으로 고려하여 스마트폰 카메라와 딥러닝 기술을 활용한 자동 포트홀 신고 시스템을 제안한다.

III. 포트홀 자동 신고 시스템 구현

이 장에서는 포트홀 신고 시스템의 구현 과정을 살펴본다. 구체적으로 딥러닝 모델 학습, 스마트폰 애플리케이션 개발, 그리고 포트홀 신고 정보를 시각적으로 제공하기 위한 웹 구현 방법에 대해 순차적으로 기술한다.

그림 1은 제안된 포트홀 신고 시스템의 전체 구조를 보여주며, 제안된 시스템은 다음과 같은 절차로 동작한다. 먼저, 포트홀 신고 애플리케이션이 설치된 스마트폰을 차량 블랙박스처럼 도로를 촬영할 수 있는 위치에 설치한다. 운전 시작 전, 사용자는 애플리케이션을 실행한다. 애플리케이션은 일정 간격으로 도로 이미지를 촬영하고, 이를 서버로 전송한다. 서버에서는 딥러닝 모델을 활용해 전송된 이미지에서 포트홀 존재 여부를 판별하고, 포트홀이 검출되면 해당 정보를 데이터베이스에 저장한다. 사용자는 웹 페이지에서 데이터베이스에 기록된 포트홀 신고 정보를 효과적으로 확인할 수 있다. 그림 2는 제안한 시스템의 전체적인 처리 절차를 보여준다.

3.1 데이터셋 전처리 및 포트홀 검출기 학습

본 논문에서는 포트홀 검출을 위해 YOLO(You Only Look Once)[9] 계열 모델 중 최신 버전인 YOLOv11의 Nano 모델을 사용하였다. 이 모델은 파라미터 수가 적어 실시간 처리에 유리하며, YOLO 시리즈 내의 다른 Nano 모델(YOLOv5n, YOLOv8n, YOLOv10n)과 비교했을 때도 높은 성능을 보인다[10]. 따라서 본 논문에서는 효율 및 성능을 만족시키는 YOLOv11 Nano 모델을 최종적으로 선택하였다. YOLO는 딥러닝 기반 실시간 객체 탐지 알고리즘으로, 이미지를 단 한 번 살펴보면 객체의 종류와 위치를 동시에 예측할 수 있다.

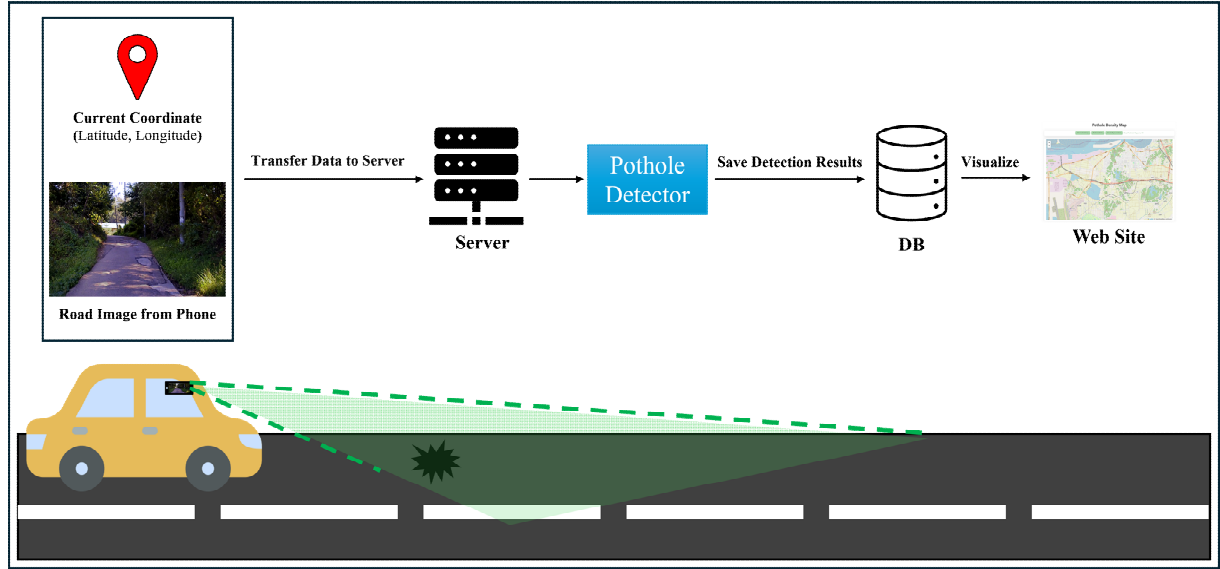


그림 1. 시스템 구조도

Fig. 1. Architecture of system

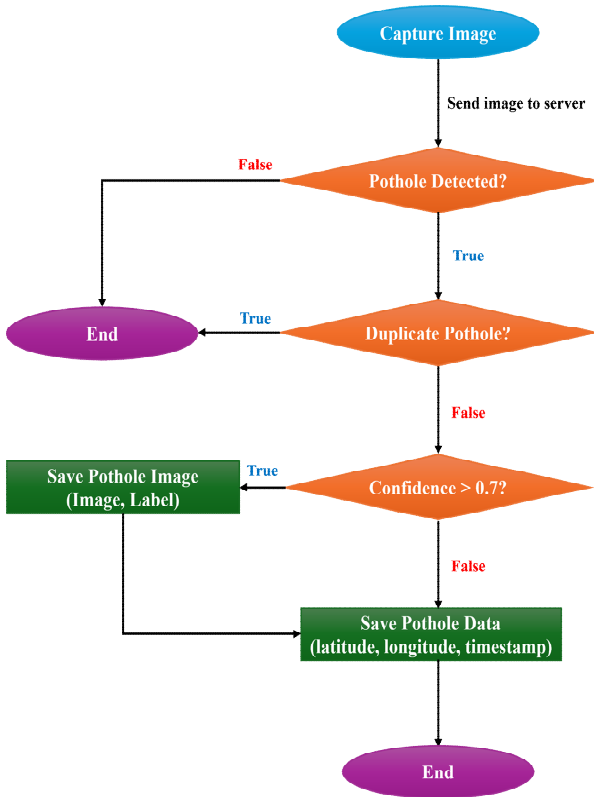


그림 2. 전체적인 처리 절차

Fig. 2. Overall processing procedure

이는 R-CNN[11] 계열 모델이 영역 제안과 분류를 순차적으로 수행하는 것과 달리, 객체 검출과 분류를 동시에 처리함으로써 높은 처리 속도를 제공한다.

본 논문에서는 Ultralytics의 오픈소스 프레임워크를 활용하여 COCO 데이터셋[12]으로 사전 학습된 모델을 포트홀 데이터셋에 적합하도록 파인튜닝(Fine-tuning)을 수행하였다.

학습 데이터는 AIHUB[13]에서 제공하는 '도로 장애물 및 표면 인지 영상(수도권)' 데이터셋으로, 동적 객체(고라니, 보행자), 정적 객체(낙하물, 라바콘, 표지판, 쓰레기), 노면(포트홀, 보수된 포트홀, 맨홀)을 바운딩 박스와 세그멘테이션(Segmentation) 방식으로 라벨링한 이미지와 어노테이션 파일로 구성된다.

전체 데이터셋은 647,830개의 이미지로 구성되었으며, 학습에는 포트홀이 포함된 이미지 중 바운딩 박스 넓이가 2,000 이상인 데이터를 선별하여 사용하였다. 이는 지나치게 작거나 부정확한 라벨 데이터를 제거해 모델의 정확도를 높이기 위함이다.

최종적으로 26,291장의 이미지를 선정해 8:2 비율로 학습 및 검증 데이터로 분할하였다. 데이터셋에 대한 정보는 표 1과 같다.

표 1. 데이터셋 정보

Table 1. Details of dataset

Dataset	Number of samples
Train	21,032
Test	5,259
Total	26,291

학습 과정에서 사용된 하이퍼파라미터는 배치 크기(Batch size)는 128, 에폭(Epoch)은 100, 학습률(Learning rate)은 0.01로 설정하였다. 기타 하이퍼파라미터는 Ultralytics의 기본값을 따랐으며, 이는 COCO 데이터셋을 비롯한 다양한 벤치마크에서 검증된 설정으로, 학습 안정성과 일반화 성능을 보장하도록 설계되었기 때문이다. 하이퍼파라미터에 대한 상세 정보는 표 2와 같다.

표 2. 하이퍼파라미터
Table 2. Hyperparameters

Hyperparameter	Value
Batch size	128
Epochs	100
Learning rate	0.01
Optimizer	SGD
Momentum	0.937
Image size	640

본 논문에서 사용된 실험 환경의 운영체제로는 Ubuntu 22.04 LTS를 사용했으며, CPU는 Intel i9-13900K(5.5GHz), RAM은 32GB, GPU는 NVIDIA RTX A6000을 사용해 학습을 진행했다. 실험 환경에 대한 정보는 표 3과 같다.

표 3. 실험 환경
Table 3. Experimental environment

Classification	Specification
OS	Ubuntu 22.04 LTS
CPU	Intel i9-13900K 5.5GHz
RAM	32GB
GPU	NVIDIA RTX A6000

본 논문에서는 검출기의 성능을 평가하기 위해 대표적인 지표인 mAP(mean Average Precision)를 사용하였다. mAP는 객체 검출 분야에서 널리 활용되는 지표로, 정밀도(Precision)와 재현율(Recall)을 종합적으로 측정하여 검출 성능을 정량화한다. 정밀도와 재현율은 각각 식 (1)과 (2)와 같이 정의된다.

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

여기서 TP(True Positive)는 모델이 객체를 올바르게 검출한 경우, FP(False Positive)는 객체가 아닌 것을 잘못 검출한 경우, FN(False Negative)는 실제 객체가 있음에도 검출하지 못한 경우를 각각 의미한다.

정밀도는 잘못된 검출(FP)을 줄이는 데 중점을 두며, 재현율은 놓친 검출(FN)을 줄이는 데 중점을 둔다. 포트홀 검출 문제에서는 도로 상태, 조명 조건, 포트홀의 크기 및 형태가 다양하게 변화하기 때문에 FP와 FN을 모두 줄이는 것이 중요하다. 이를 위해 두 지표(정밀도, 재현율)를 개별적으로 분석하면 특정 오류 유형의 발생 원인을 파악할 수 있다.

그러나 정밀도와 재현율만으로 모델의 전반적인 성능을 충분히 평가하기는 어렵다. 따라서 본 논문에서는 정밀도-재현율(PR) 곡선 아래 면적인 AP(Average Precision)를 구한 뒤, 모든 클래스에 대해 이를 평균 내는 mAP를 지표로 사용하였다. mAP는 식 (3)과 같이 정의된다.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (3)$$

여기서 N 은 클래스의 수를 의미한다, 본 논문에서는 검출 대상 클래스가 포트홀 하나이므로 $N=1$ 이 된다. 즉, 단일 클래스의 AP 값이 곧 mAP가 된다.

본 논문에서는 검출기의 성능을 보다 세밀하게 평가하기 위해 두 가지 형태의 mAP를 사용하였다. mAP50은 IoU(Intersection over Union) 임계값(Threshold)을 0.5로 고정하여 계산한 mAP이며, mAP50-95는 IoU 임계값을 0.5에서 0.95까지 0.05 간격으로 변화시키며 평균을 계산한 값이다.

mAP50은 낮은 IoU 임계값(0.5)에서의 검출 성능을 평가할 때 유용하며, mAP50-95는 다양한 임계값 구간에서 모델이 얼마나 일관되게 포트홀을 검출하는지를 종합적으로 분석하는 데 적합하다.

모델 학습 및 실험 결과, 학습한 검출기의 mAP50과 mAP50-95 값은 각각 0.54736과 0.30882로 나타났으며, 상세한 성능 결과는 표 4와 같다.

표 4. 검출기 성능평가 결과

Table 4. Detection model performance evaluation results

Precision	Recall	mAP50	mAP50-95
0.63978	0.49262	0.54736	0.30882



그림 3. 포트홀 검출기 결과
Fig. 3. Pothole detection results

그림 3은 테스트 이미지를 학습된 검출기의 입력으로 하여 얻은 결과이다. 그림 3을 분석한 결과, 동그란 형태의 포트홀은 높은 신뢰도(Confidence)로 검출되었으나, 세로로 긴 포트홀의 경우에는 낮은 신뢰도를 보이는 것을 확인할 수 있었다. 따라서 세로로 길거나 불규칙한 형태의 포트홀에 대한 검출 성능 개선이 필요하다고 판단된다.

3.2 실시간 포트홀 신고 애플리케이션

본 논문에서는 스마트폰 카메라를 활용하여 포트홀을 자동으로 신고할 수 있는 애플리케이션을 개발하였다. 애플리케이션은 크게 클라이언트와 서버 부분으로 나뉘며, 각 부분의 주요 기능과 구현 방식은 다음과 같다.

클라이언트는 iOS와 안드로이드(Android) 플랫폼을 모두 지원하기 위해 플러터(Flutter) 프레임워크로 구현되었다.

본 애플리케이션은 사용자 계정 관리를 위한 로그인 및 회원가입 화면, 그리고 포트홀 검출을 위한 카메라 화면으로 구성된다. 카메라 화면에서는 촬영한 이미지를 서버로 전송하는 기능을 제공한다. 사용자가 시작 버튼을 누르면, 현재 카메라에서 촬영된 이미지와 스마트폰의 위치 정보가 2초마다 서버로 전달된다. 전송 주기를 2초로 설정한 이유는 서버 과부하 방지와 이미지 전송 속도를 고려한 결과이다. 그림 4는 구현된 애플리케이션의 카메라 화면

의 실행 결과를 보여준다.

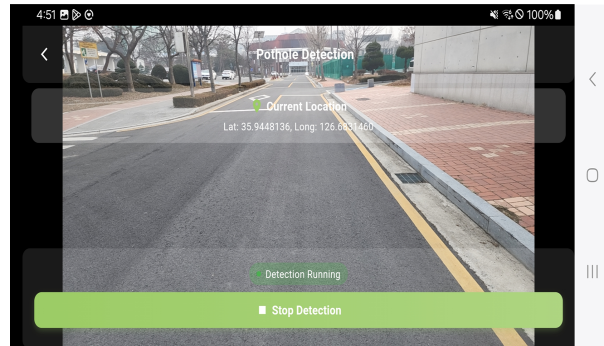


그림 4. 애플리케이션의 화면
Fig. 4. Application screen

서버는 파이썬(Python) 기반의 경량 웹 프레임워크인 FastAPI를 사용하여 구현하였다. FastAPI는 간결하고 효율적인 구현이 가능하며, 특히 PyTorch 기반 딥러닝 모델과의 호환성이 뛰어나다는 장점을 지닌다.

서버는 사용자 인증(회원가입 및 로그인) 기능을 제공하며, 사용자가 업로드한 이미지를 입력으로 받아 포트홀을 검출하는 역할을 수행한다. 포트홀 검출은 PyTorch로 학습된 딥러닝 모델을 통해 이루어지며, 검출된 포트홀 정보는 PostgreSQL 데이터베이스에 저장된다. 데이터베이스는 User 테이블과 Pothole 테이블로 구성되는데, User 테이블에는 사용자 계정 관리를 위한 정보가 저장되고, Pothole 테이블에는 검출된 포트홀의 정보가 기록된다. 각 테이블의 구조는 표 5와 같다.

표 5. 데이터베이스 테이블 구조

Table 5. Database table structure

Table	Field	Data type
User	user_id	PK, Integer
	name	VARCHAR(50)
	email	VARCHAR(100)
	password	VARCHAR(50)
	phone	VARCHAR(20)
Pothole	pothole_id	PK, Integer
	latitude	DOUBLE PRECISION
	longitude	DOUBLE PRECISION
	discovered	TIMESTAMP
	user_id	FK, Integer

PostgreSQL을 선택한 이유는 PostGIS 확장을 통해 지리 데이터를 효과적으로 처리할 수 있기 때문이다. PostGIS는 위도(Latitude)와 경도(Longitude)를 기반으로 한 지리적 거리 계산을 효율적으로 수행할 수 있게 해주어, 사용자 위치를 기준으로 반경 내의 포트홀을 신속하게 검색하고 분석할 수 있다.

포트홀 검출 과정에서 신뢰도 임계값은 0.5로 설정되었으며, 모델이 검출한 포트홀의 신뢰도가 이 값을 초과할 경우에만 해당 객체를 포트홀로 인정한다. 신뢰도는 모델이 특정 객체가 실제로 포트홀일 확률과 해당 객체의 존재 확률을 결합해 계산된다. 신뢰도는 식 (4)와 같이 정의된다.

$$Confidence = P(object) \cdot P(class|object) \quad (4)$$

신뢰도가 0.5를 초과한 포트홀 검출 결과는 검출 시각, 위도, 경도 정보와 함께 데이터베이스에 저장된다. 신뢰도가 0.7 이상인 경우, 검출된 포트홀 이미지와 바운딩박스 정보는 학습 데이터로 활용하기 위해 서버에 저장된다. 신뢰성 있는 데이터를 확보하기 위해, 동일한 사용자가 5m 이내에서 이미 감지한 포트홀은 중복 저장되지 않도록 설정하였다.

3.3 포트홀 데이터 시각화 웹 시스템

포트홀 신고 데이터를 효과적으로 시각화하기 위해 HTML(Hyper Text Markup Language), CSS(Cascading Style Sheets), JavaScript를 활용한 웹 시

스템을 개발하였다. 지도 기반 시각화를 위해 JavaScript의 경량 오픈소스 라이브러리인 Leaflet을 사용하였으며, 이를 통해 포트홀의 위치와 분포를 직관적으로 확인할 수 있도록 구현하였다. 또한, 히트맵을 적용하여 포트홀이 집중적으로 발생하는 지역을 시각적으로 강조하였다. 구현된 웹 페이지의 그림 5와 같다.

IV. 결론 및 향후 과제

본 논문에서는 차량에 부착된 스마트폰 애플리케이션을 활용해 포트홀을 자동으로 검출하고 신고할 수 있는 시스템을 제안하고 구현하였다. 본 시스템은 사용자가 직접 포트홀을 신고하지 않아도 차량의 주행 영상을 통해 포트홀을 자동으로 감지하고 신고할 수 있도록 설계되었다. 이를 통해 기존의 수동적인 포트홀 신고 방식의 효율성을 향상시키는 장점을 제공한다.

수집된 데이터는 웹 페이지에서 시각화되어 특정 지역의 포트홀 발생 빈도를 확인할 수 있다. 이러한 시각화는 도로 관리자들이 도로 상태를 파악하고, 유지보수 작업을 수행하는 데 도움을 준다. 특히, 포트홀 발생 빈도가 높은 지역을 사전에 예측하거나 우선적으로 관리할 수 있는 기반을 제공함으로써 도로 관리의 효율성을 높이는 데 기여할 수 있을 것으로 판단된다.

하지만 본 시스템의 신뢰성을 더욱 향상시키기 위해 해결해야 할 몇 가지 과제가 존재한다. 먼저, 서버에서 작동하는 포트홀 검출 모델의 성능을 개선하는 것이 필요하다. 이를 위해서는 모델 변경, 데이터 수집, 데이터 증강과 같은 다양한 방법이 적용될 수 있는데, 특히 생성형 AI를 활용한 데이터 증강 기술은 부족한 데이터를 보완하며, 다양한 환경에서 모델 성능을 강화하는 데 중요한 역할을 할 것으로 기대된다[14].

또한 현재 시스템은 단일 혹은 소수의 사용자를 대상으로 설계되었기 때문에, 향후 사용자 수가 증가함에 따라 서버 과부하 문제가 발생할 가능성이 있다. 이러한 문제를 해결하기 위해 서버 성능을 최적화하고 분산 처리 기술을 적용하는 방안이 필요하다.

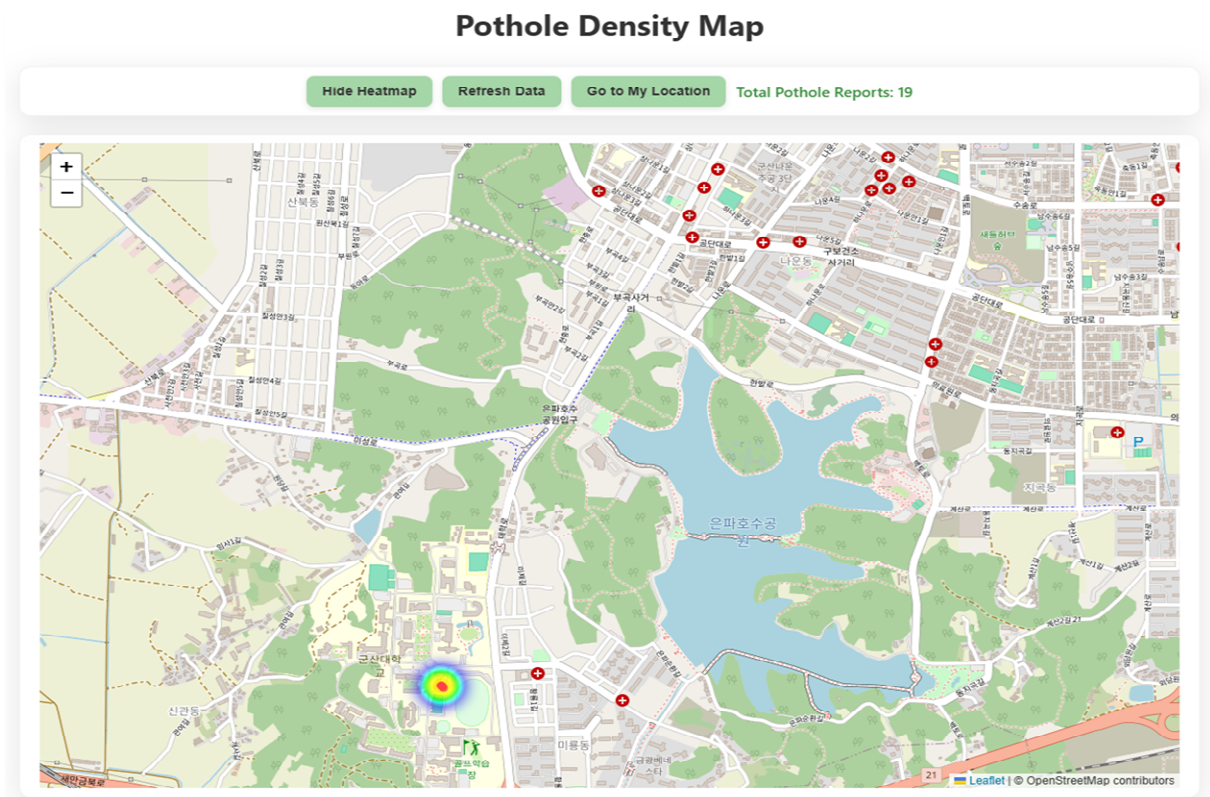


그림 5. 포트홀 신고 시각화 웹
Fig. 5. Pothole reporting and visualization website

이와 같은 개선 과제를 해결함으로써 시스템의 성능과 안정성을 강화하고, 궁극적으로 더 나은 도로 안전 관리와 효율적인 유지보수 체계를 실현할 수 있을 것으로 기대된다. 이러한 연구는 도로 안전 향상뿐만 아니라 스마트 도로 관리 기술 발전에도 기여할 것으로 판단된다.

References

- [1] Wikipedia, <https://en.wikipedia.org/wiki/Pothole>. [accessed: Jan. 21, 2025].
- [2] Newsis, "Highway 'potholes' exceeded 22,752 cases in the past five years... Compensation amount reached 4.438 billion KRW", https://www.newsis.com/view/NISX20240730_0002830621. [accessed: Jan. 20, 2025]
- [3] R. Khanam and M. Hussain, "YOLOv11: An Overview of the Key Architectural Enhancements", arXiv preprint arXiv:2410.17725, Oct. 2024. <https://doi.org/10.48550/arXiv.2410.17725>.
- [4] M. Mednis, et al., "Real time pothole detection using Android smartphones with accelerometers", International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS), Barcelona, Spain, pp. 1-6, Jun. 2011. <https://doi.org/10.1109/DCOSS.2011.5982206>.
- [5] J. J. Kim, B. H. Kang, and S. I. Choi, "2D LiDAR based 3D Pothole Detection System", Journal of Digital Contents Society, Vol. 18, No. 5, pp. 989-994, Aug. 2017. <https://doi.org/10.9728/dcs.2017.18.5.989>.
- [6] S. H. Cho, et al., "A study on pothole detection using a pre-trained deep learning model based on UAV images", Korean Society of Civil Engineers Conference, Jeonnam, Korea, pp. 49-50, Oct. 2023.
- [7] A. Dhiman and R. Klette, "Pothole detection using computer vision and learning", IEEE Transactions on Intelligent Transportation Systems, Vol. 21, No. 8, pp. 3536-3550, Aug. 2020. <https://doi.org/10.1109/TITS.2019.2931297>.

- [8] J. M. Lee and S. J Kim, "Bicycle Road Analysis using Clustering based on Deep Learning", The Journal of Korean Institute of Information Technology, Vol. 21, No. 1, pp. 131-138, Jan. 2023. <http://dx.doi.org/10.14801/jkiit.2023.21.1.131>.
- [9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 779-788, Jun. 2016.
- [10] D. J. Yu, et al., "Comparative Evaluation of Performance of YOLO Models for Pothole Detection", Proceedings of KIIT Conference, Jeju, pp. 846-849, Nov. 2024.
- [11] R. Girshick, et al., "Rich feature hierarchies for accurate object detection and semantic segmentation", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, pp. 580-587, Jun. 2014.
- [12] T. Y. Lin, et al., "Microsoft COCO: Common Objects in Context", European Conference on Computer Vision (ECCV), Zurich, Switzerland, pp. 740-755, Sep. 2014. https://doi.org/10.1007/978-3-319-10602-1_48.
- [13] AIHub, <https://www.aihub.or.kr> [accessed: Jan. 21, 2025]
- [14] M. J. Cho, et al., "Dataset Construction Using Generative AI for Pothole Detection and Performance Evaluation of CNN Models", Journal of Digital Contents Society, Vol. 25, No. 10, pp. 3061-3069, Oct. 2024. <https://doi.org/10.9728/dcs.2024.25.10.3061>.

저자소개

유 동 주 (Dongju Yu)



2019년 3월 ~ 현재 :

국립군산대학교 소프트웨어학과
학사과정

관심분야 : 인공지능, 데이터분석

이 석 훈 (Sukhoon Lee)



2009년 2월 : 고려대학교

전자및정보공학부(학사)

2011년 2월 : 고려대학교

컴퓨터·전파통신공학과(공학석사)

2016년 2월 : 고려대학교

컴퓨터·전파통신공학과(공학박사)

2016년 3월 ~ 2017년 3월 :

아주대학교 의료정보학과 연구강사

2017년 4월 ~ 현재 : 국립군산대학교 소프트웨어학과
부교수

관심분야 : 사물인터넷, 메타데이터 레지스트리, 데이터
품질, 연합 학습

정 동 원 (Dongwon Jeong)



1997년 2월 : 군산대학교

컴퓨터과학과(학사)

1999년 2월 : 충북대학교

전자계산학과(석사)

2004년 2월 : 고려대학교

컴퓨터학과(박사)

2005년 4월 ~ 현재 :

국립군산대학교 소프트웨어학과 교수

관심분야 : 데이터베이스, 시맨틱 서비스, 빅데이터,
사물인터넷, 엣지컴퓨팅, 지능형 융합 서비스