

YOLOv8-BASED REAL-TIME POTHOLE DETECTION WITH INTEGRATED GPS ALERTS AND CIVIC REPORTING

Dr. D. Anitha Kumari¹, Ramishetty Suharika², Peetani Saketh Goud³, Pothinipalli Sriman Goud⁴, Thota Nikhil⁵

¹Professor, *Department of CSE (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING) TKR COLLEGE OF ENGINEERING & TECHNOLOGY*

^{2,3,4,5} *UG Scholars in Department of CSE (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING) TKR COLLEGE OF ENGINEERING & TECHNOLOGY*

ABSTRACT: Road safety is a critical concern in urban environments, where potholes contribute significantly to accidents, vehicle damage, and traffic congestion. Traditional reporting systems relying on manual inspection or citizen complaints are slow, inconsistent, and lack location precision. This thesis proposes an automated Pothole Detection and Alert System using YOLOv8, integrating deep learning, geolocation tracking, and real-time notification services. Users capture and upload road images through a mobile or web application along with GPS coordinates. The backend processes images using a pre-trained YOLOv8 model that identifies potholes and classifies their severity. Detected pothole details — including location, timestamp, and severity level are stored in a cloud-based MongoDB database. A dedicated government dashboard presents all reported potholes sorted by severity, enabling authorities to prioritize repairs efficiently. Status updates trigger instant notifications to nearby users through Firebase Cloud Messaging (FCM) or Socket.io, helping commuters adjust driving behavior and reducing accident risks. Maps API integration enables real-time spatial visualization across the application. This dissertation demonstrates that by leveraging machine learning, location intelligence, and cloud-based services, the degree to which urban infrastructure management improves is substantial. The sample thesis validates that AI-driven systems contribute meaningfully to smarter, safer city development.

KEYWORDS :Pothole Detection, Road Damage Detection, YOLOv8, Convolutional

Neural Network (CNN), Image Processing, Severity Analysis, Real-time Detection, Geo-tagged Data, Web Application, Data Preprocessing, Database Management, Historical Data Tracking

1. INTRODUCTION

1.1 Motivation:

Potholes are a widespread and persistent problem that directly impact public safety, transportation efficiency, and overall quality of life. They cause vehicle damage, increase maintenance costs, disrupt traffic flow, and in many cases lead to serious accidents. Despite their impact, the current reporting and repair processes in many regions remain slow and highly manual. Authorities often depend on citizen complaints or periodic inspections, both of which fail to capture the true scale and urgency of road damage. This delay between pothole formation and repair creates avoidable risks for daily commuters. The motivation for choosing this project arises from the need for a faster, more accurate, and reliable method to identify and manage potholes. By leveraging automated pothole detection through image processing and severity analysis, the system ensures that every reported pothole is documented with precise location data. This allows authorities to prioritize repairs based on actual severity rather than delayed reports. Additionally, providing real-time alerts to users traveling near damaged road sections can significantly reduce accidents and improve road safety. This project was chosen to address a real problem that affects thousands of people every

day. It aims to create a practical, efficient, and impactful solution that enhances public safety and supports better infrastructure management.

1.2 Problem Definition:

Potholes are a persistent and widespread problem that adversely affect road safety, increase vehicle repair costs, and disrupt transportation systems worldwide. Their presence on roads not only leads to accidents and traffic congestion but also accelerates wear and tear on vehicles, resulting in significant financial burdens for commuters. In many regions, the existing methods for pothole detection and reporting rely heavily on manual processes such as citizen complaints, helpline calls, and routine inspections by municipal staff. These traditional approaches are slow, labor-intensive, and inconsistent, often failing to reflect real-time road conditions. As a result, numerous potholes remain unnoticed or unreported for extended periods, allowing them to deteriorate further and posing serious risks to daily road users. A major limitation in the current system is the lack of accurate GPS-based location tracking and standardized severity classification. Without precise coordinates and consistent assessment criteria, municipal authorities struggle to prioritize repairs or allocate resources effectively. Many reports include incomplete information, unclear or low-quality images, and vague descriptions of the problem, making verification difficult. Duplicate submissions for the same pothole also add to administrative inefficiencies, causing confusion and unnecessary delays. Moreover, even when potholes are reported, existing systems rarely incorporate mechanisms to alert users about dangerous road conditions nearby. This leaves commuters unaware of upcoming hazards, increasing the chances of sudden maneuvers, accidents, or vehicle damage. The absence of automation, real-time data analysis, and integrated maintenance workflows further reduces the operational efficiency of road management departments. Manual inspection cycles cannot keep pace with the dynamic nature of road

deterioration, especially in cities with heavy traffic or extreme weather conditions. These challenges highlight the urgent need for a smart, automated, and location-aware pothole detection and alert system. Such a system must be capable of accurately identifying potholes through advanced machine learning techniques, assessing severity levels consistently, delivering instant alerts to road users, and providing actionable insights to authorities. By integrating detection, reporting, analysis, and notification into a unified workflow, this solution can significantly accelerate decision-making, improve public safety, and support proactive road maintenance planning.

2. Proposed System:

The proposed system introduces an automated and intelligent pothole detection and alert solution designed to overcome the limitations of existing manual and semi-digital reporting methods. The system integrates deep learning-based image analysis, geolocation tracking, and real-time communication to create a highly efficient framework for identifying potholes, assessing their severity, and informing both authorities and road users.

The process begins at the user application, where individuals can upload an image of a road surface along with automatically captured GPS coordinates. This ensures accurate geographical mapping of every reported pothole. The backend server receives the data and forwards the image to a YOLOv8-based detection module. YOLOv8, a state-of-the-art object detection model, processes the image to identify the presence of potholes and evaluates their severity based on characteristics such as size, depth appearance, and surface irregularity. This automated severity scoring enables consistent and unbiased repair prioritization.

Once processed, the system stores all information—image, severity score, location, and timestamp—into a centralized database. This database powers a dedicated government dashboard, where authorities are presented with

a structured list of potholes sorted according to severity. By removing the reliance on map interfaces and focusing on a clean severity-based list, the dashboard simplifies decision-making and allows municipal teams to quickly identify critical road defects requiring immediate attention. Administrators can also update the repair status of each pothole, ensuring that the system maintains up-to-date infrastructure records.

A key component of the proposed system is the real-time alert mechanism. When a user approaches a damaged road section, the system leverages notification services such as Firebase Cloud Messaging (FCM) or Socket.io to send timely alerts. These notifications help drivers adjust their speed or route, thereby reducing accident risks and improving commuter safety. To support wide-scale deployment, the system incorporates a Maps API integration for user-side visualization, scalable backend infrastructure, and robust data management. By combining machine learning, location intelligence, and automated communication, the proposed system delivers a comprehensive, fast, and reliable pothole management solution that enhances public safety, improves maintenance efficiency, and modernizes road infrastructure monitoring.

3. Use Case Diagram

A Use Case Diagram represents how different users (actors) interact with a system to perform specific tasks. It focuses on what the system should do, not how it is built. Actors can be people, external systems, or devices that communicate with the application. Each action performed by an actor is shown as a “use case,” represented through ovals. The diagram captures high-level functional requirements by showing all possible interactions between users and the system.

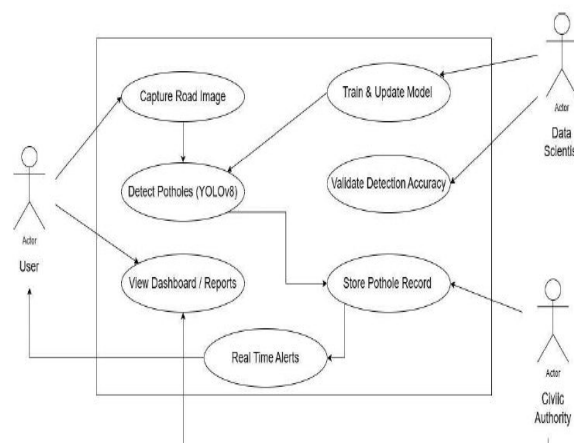


Fig.1: Use Case Diagram

In this pothole detection project, the Use Case Diagram includes actors like Driver, Admin, System, and Database. They interact with use cases such as capture image, detect pothole, geotag location, store data, send alert, and view reports. This diagram helps clearly identify what services the system must provide and how each user benefits from it.

Use Case Diagrams are useful for requirement analysis, stakeholder discussions, and feature planning. They offer a simplified view of system functionality, ensuring every key interaction is included before development begins. Ultimately, they help teams avoid misunderstandings, define user expectations, and establish a solid foundation for the system’s design and development.

A. Sequence Diagram

A Sequence Diagram shows how different system components interact step by step over time. It uses lifelines and arrows to represent objects and the messages they exchange. In the pothole detection system, the Camera captures a live frame and sends it to the YOLOv8 Detector, which identifies potholes and returns bounding boxes. The GeoTagger attaches GPS coordinates, and the enriched data is then stored in the Database. Finally, when a user approaches the pothole location, the Alert Service retrieves the data and sends a notification.

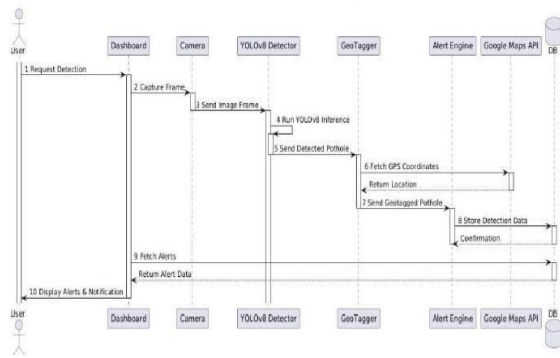


Fig. 2: Sequence Diagram

4.1 Algorithms

4.1.1 Computer Vision Object Detection

Computer Vision–based object detection is the core technology used to automatically identify potholes from road images or video frames using deep learning. When a user captures or uploads an image, it first undergoes preprocessing steps such as resizing to a fixed resolution and normalizing pixel values to ensure consistent and efficient model performance. During training, data augmentation techniques like brightness adjustment, rotation, and flipping may be applied to improve the model’s ability to handle different lighting, weather, and road conditions.

The preprocessed image is then passed through a Convolutional Neural Network (CNN)-based object detection model. The network has been trained on labeled road images to learn patterns associated with potholes, such as irregular textures, cracks, depressions, and surface distortions. During detection, the model predicts bounding boxes around suspected pothole regions and assigns confidence scores that indicate the probability of accurate detection.

To refine the results, overlapping or duplicate detections are removed using techniques such as Non-Maximum Suppression (NMS), ensuring only the most accurate bounding box is retained. The final output includes pothole location coordinates, confidence score, and estimated size, which can be used to calculate

severity. These results are then sent to the backend, stored in the database with GPS and timestamp data, and displayed on dashboards or maps for monitoring and decision-making.

4.1.2 REST API Routing & Data Handling

REST API routing and data handling form the communication bridge between the frontend interface, backend server, AI detection module, and database. The system follows a RESTful architecture where different HTTP methods such as POST and GET are used to perform specific operations. When a user uploads a road image through the web interface, the frontend sends a POST request to a defined API endpoint. This request typically contains the road image (either as a file or base64-encoded string) along with metadata such as latitude, longitude, timestamp, and user ID formatted in JSON. The POST method is used because new data (image and location information) is being submitted to the server for processing and storage.

Once the backend receives the request, it uses HTTP routing mechanisms to direct the request to the appropriate handler function. The backend parses the incoming JSON data, extracts the image file and GPS coordinates, and performs validation checks to ensure the data is complete and secure. The image is then forwarded to the AI detection service or module for pothole analysis. After the deep learning model processes the image, it returns detection results including bounding box coordinates, confidence scores, and severity levels. If a pothole is detected above a predefined confidence threshold, the backend prepares structured data containing the location, detection details, severity score, timestamp, and possibly user information. It then executes database queries to store this information securely in a relational or NoSQL database.

For retrieving pothole information, the system uses a GET request. When the frontend needs to display detected potholes on a map or administrative dashboard, it sends a GET

request to another API endpoint. The backend handles this request by querying the database to fetch all relevant records or filtered results based on criteria such as date, severity, or location. The retrieved data is then converted into structured JSON format and sent back as an HTTP response. The frontend parses this JSON response and dynamically renders the pothole markers on maps or lists them in dashboards. This REST-based communication ensures smooth, scalable, and structured data exchange between all system components while maintaining clarity, security, and efficiency.

4.1.3 Database Images

This image shows a collection of pothole photographs arranged in a grid format, typically representing a dataset used for training machine-learning models such as YOLOv8. Each thumbnail displays a different road surface with potholes, cracks, or damaged areas captured under various lighting conditions, angles, and environments. The diversity in the images helps the model learn how potholes appear in real-life situations, making detection more accurate and reliable.



Fig. 2: - Database Images (Dataset Sample Grid)

Below each image, the file name and size are displayed, indicating that these pictures are part of an organized dataset. The dataset includes both color and grayscale images, as well as potholes of different shapes, sizes, and severity levels. Such variety ensures that the AI system can generalize well, recognize potholes on

different types of roads, and perform effectively even in challenging conditions like shadows, uneven surfaces, or low lighting.

5. IMPLEMENTATION & RESULTS

5.1 Explanation of Key functions

The YOLOv8-Based Real-Time Pothole Detection System is built upon several key functions that work together to deliver automated road hazard detection, severity classification, and real-time user alerting. Each function plays a critical role in the overall workflow of the system.

5.1.1. Pothole Detection Function

The core function of the system is automated pothole detection using the YOLOv8 deep learning model. When a user uploads a road surface image, the detection function preprocesses the image resizing, normalizing, and adjusting brightness before passing it through the trained YOLOv8 Convolutional Neural Network. The model performs single-pass feature extraction and generates bounding box predictions with confidence scores for each detected pothole. Non-Maximum Suppression (NMS) is then applied to eliminate overlapping detections, ensuring only the most accurate bounding boxes are retained.

5.1.2. Severity Estimation Function

Once a pothole is detected, the severity estimation function calculates a severity score by computing the ratio of the detected bounding box area to the total image area, multiplied by 100. This score is capped at a maximum of 100 and classified into three levels Low (score below 30), Medium (score between 30 and 60), and High (score above 60). This function ensures consistent and objective prioritization of road repairs without manual assessment.

5.1.3. GPS Geolocation Binding Function

The geolocation function invokes the device's built-in GPS sensor at the moment of image capture to extract precise latitude and longitude coordinates. These coordinates are packaged

with the image and metadata into a standardized request payload and transmitted to the backend. The function ensures every pothole record is bound to an exact real-world geographic location, enabling accurate map visualization and proximity-based alerting.

5.1.4. User Authentication Function

The authentication function manages user registration and login using JWT (JSON Web Token) based security. During registration, user passwords are hashed using bcrypt with a salt round of 10 before storage in MongoDB. During login, the submitted password is compared against the stored hash, and upon successful verification, a JWT token signed with a secret key is generated with a 7-day expiry. This token is used to authorize all subsequent API requests, ensuring secure access control throughout the system.

5.1.5. Proximity Alert Function

The proximity alert function continuously monitors the user's current GPS coordinates and queries the MongoDB database for unrepaired potholes within a configurable radius (default 300 meters) using the GeoJSON \$near geospatial operator with a 2dsphere index. When nearby potholes are detected, the function triggers an immediate alert notification to the user, warning them of road hazards ahead and enabling proactive driving adjustments.

5.1.6. Government Dashboard Function

The dashboard function provides authorized administrators with a structured interface to view, filter, and manage all reported potholes. It retrieves pothole records from MongoDB sorted by severity (High to Low) or filtered by status (Reported, In Progress, Fixed). Administrators can update the repair status of individual potholes directly from the dashboard, and the function reflects these changes in real time across the system.

5.1.7. Map Visualization Function

The map visualization function integrates Google Maps API and Leaflet Maps to render real-time spatial visualization of pothole locations. The user's current GPS position is displayed as a live marker, while detected potholes are plotted as color-coded markers red for nearby hazards and green for distant ones. A 300-meter radius circle is drawn around the user's location to visually indicate the proximity alert zone.

5.1 Method of Implementation

Frontend (Web Interface)

- Users can upload road images or videos and see pothole detection results with severity.
- Admin dashboard allows monitoring and viewing historical road condition data.

Backend (Python)

- Handles API requests, image/video uploads, and communication with AI modules.
- Manages authentication and secure access for users. Pothole Detection Module (Deep Learning)
- Uses CNN to detect potholes in images after preprocessing (resize, normalize, augment)
- Calculates severity scores of potholes on the images.

Database

- Stores images, detection results, timestamps, and user data.
 - Ensures secure and structured management of road condition data.
- #### Deployment
- Web-based application accessible via browser.
 - Backend APIs deliver detection and route optimization results to frontend

5.1.1 Forms

The system includes the following input forms through which users interact with the application:

1. User Registration Form

The registration form collects three fields Full Name, Email Address, and Password. The form applies client-side validation before submission. The name field requires a minimum of 3 characters containing only alphabetic characters. The email field accepts only valid Gmail addresses (@gmail.com format). The password field enforces strong password rules requiring a minimum of 8 characters with at least one uppercase letter, one lowercase letter, one numeric digit, and one special character. Error messages are displayed inline below the form upon validation failure.

2. User Login Form

The login form collects Email Address and Password. Upon submission, the credentials are sent to the backend authentication API. If the login is successful, the JWT token and user data are stored in the browser's local Storage, and the user is redirected to either the User Dashboard or Admin Dashboard based on their assigned role.

3. Pothole Report Submission Form

The pothole reporting form contains an image upload field that accepts image files only (JPEG, PNG). The form automatically captures and displays the user's current GPS coordinates (Latitude and Longitude) retrieved from the device's geolocation sensor. The Submit Report button triggers the multipart form data upload to the backend API along with the GPS metadata. A loading indicator is displayed while the request is being processed.

4. Admin Status Update Form

The admin dashboard provides an inline dropdown form for each pothole record that allows administrators to update the repair status. The dropdown contains three options — Reported, In Progress, and Fixed. Upon

selection, an HTTP PUT request is automatically triggered to update the pothole status in the database and the dashboard list refreshes immediately to reflect the change.

5.1.1 Output Screens

The system produces the following output screens:

1. Login Screen

The login screen features a dark gradient background with a glassmorphism-style login card. It displays a "Welcome Back" heading, email and password input fields, a Login button, and a link to the Signup page. Error messages are displayed in red text upon failed authentication attempts.

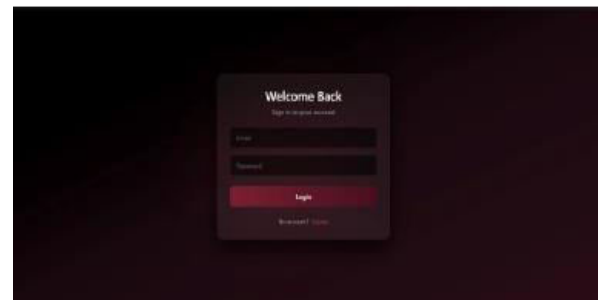


Figure 5.1- Login Screen

2. Signup Screen

The signup screen follows the same dark glassmorphism design theme as the login screen. It displays a "Create Account" heading with input fields for Full Name, Email, and Password. Inline validation error messages guide the user to correct any invalid inputs before successful registration.

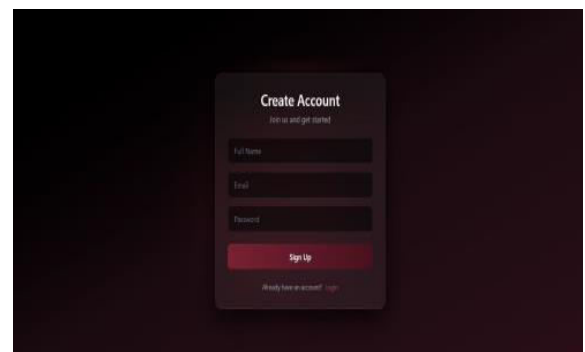


Figure 5.2 - Signup Screen

3. User Dashboard Screen

The user dashboard is divided into two panels. The left panel (two-thirds width) displays a live Google Maps view centered on the user's current GPS location with markers indicating the user's position and nearby potholes within a 300-meter radius. The right panel contains the pothole reporting form with image upload, GPS coordinate display, and submit button. An alert popup notification appears automatically when potholes are detected within proximity.

4. User Dashboard Page 2 Screen

This screen displays a full-width Leaflet map with the user's live location marked at the center. A blue circle of 300-meter radius is drawn around the user. All reported potholes are displayed as map markers — red markers indicate potholes within the alert proximity zone and green markers indicate potholes at normal distance. Popup labels appear on each marker showing hazard status.

5. Admin Dashboard Screen

The admin dashboard is divided into a left sidebar and a right main content area. The left sidebar displays four statistical cards showing Total Potholes, Reported count, In Progress count, and Fixed count in color-coded panels (blue, red, yellow, green). The right content area displays a list of all pothole records with severity score, reporter name, current status badge, a status update dropdown, and a thumbnail image of the reported pothole. Filter dropdowns at the top allow sorting by severity and filtering by status.

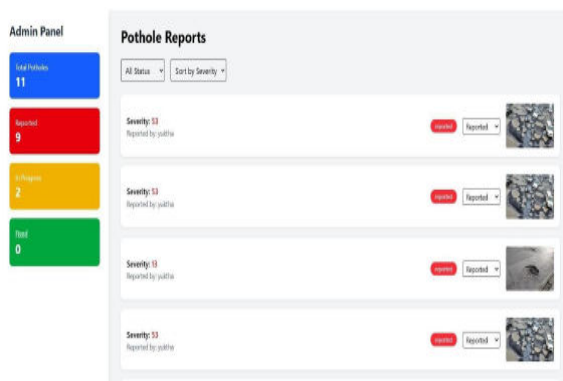


Figure 5.3 - Admin Dashboard Screen

5.1.1 Result Analysis

The result analysis evaluates the overall performance and effectiveness of the implemented YOLOv8-Based Pothole Detection and Alert System across its key functional modules.

1. Pothole Detection Accuracy

The YOLOv8 model trained on the Atulya Kumar and Raj Dalsaniya Kaggle datasets demonstrates reliable pothole detection across diverse road surface conditions including varied lighting, road textures, and pothole sizes. The model operates with a confidence threshold of 0.4 (40%), ensuring only sufficiently confident detections are reported, thereby minimizing false positives. The bounding box predictions accurately localize potholes within the submitted images.

2. Severity Classification Results

The severity estimation algorithm consistently classifies detected potholes into three levels based on the bounding box area ratio to total image area. Potholes with a severity score below 30 are classified as Low, scores between 30 and 60 as Medium, and scores above 60 as High. This automated classification eliminates human bias and provides municipal authorities with an objective basis for repair prioritization.

3. Geolocation Accuracy

The GPS geolocation binding function captures device coordinates with high accuracy using the browser's Geolocation API with the enableHighAccuracy flag set to true. Each pothole record is stored as a GeoJSON Point in MongoDB with longitude and latitude coordinates, enabling precise map-based visualization and accurate geospatial proximity queries.

4. Proximity Alert Performance

The proximity alert system using MongoDB's \$near geospatial operator with a 2dsphere index

efficiently retrieves nearby unrepaired potholes within the configured radius of 300 meters. The alert is triggered in real time as the user's location updates through the GPS watchPosition function, providing timely warnings to commuters approaching road hazards.

5. Admin Dashboard Efficiency

The government administrative dashboard successfully retrieves, filters, and sorts pothole records from MongoDB by severity and status. The inline status update functionality allows administrators to update repair status in a single interaction, and the dashboard statistics cards provide an instant overview of the current state of road infrastructure management.

6. System Performance

The three-tier architecture with decoupled frontend, backend, and AI service ensures efficient load distribution. The FastAPI AI microservice handles image processing independently without blocking the main backend server. MongoDB Atlas provides scalable cloud storage with geospatial indexing that supports efficient real-time queries across large datasets.

Overall, the system successfully achieves all its core objectives automated detection, severity classification, precise geolocation, government coordination, and real-time user alerting validating its effectiveness as a practical and scalable solution for modern urban road infrastructure management.

6. CONCLUSION

The present work successfully demonstrates the development and validation of an intelligent, automated pothole detection and civic reporting system that integrates state-of-the-art deep learning, geolocation services, and real-time notification technologies into a unified and scalable platform.

The system addresses a critical gap in urban road safety management — the absence of a

fast, consistent, and automated mechanism for detecting potholes, classifying their severity, and communicating hazard information to both municipal authorities and road users in real time. By leveraging YOLOv8 for single-pass object detection, the system achieves high-confidence pothole identification from road surface images submitted through a mobile or web interface. The severity estimation module provides objective, threshold-based classification into Low, Medium, and High categories, enabling data-driven repair prioritization without manual assessment.

The GPS geolocation binding mechanism ensures that every reported pothole is anchored to a precise real-world coordinate, eliminating the ambiguity and imprecision that characterize traditional citizen complaint systems. The centralized MongoDB database with 2dsphere geospatial indexing supports efficient storage, retrieval, and proximity-based querying of pothole records at scale. The government administrative dashboard empowers municipal authorities to monitor all reported potholes sorted by severity, update repair statuses, and maintain an organized workflow for infrastructure maintenance operations.

The real-time proximity alert system, implemented using the Haversine formula for distance calculation and delivered via browser-based notifications and map visualizations using Google Maps API and Leaflet/OpenStreetMap, proactively warns commuters approaching hazardous road locations. This transforms the system from a passive reporting tool into an active road safety platform that directly reduces accident risks for daily commuters.

Comparative analysis with existing approaches confirms that the proposed system significantly outperforms traditional manual reporting systems, sensor-based detection methods, and standalone detection tools by combining detection, severity classification, geolocation, government coordination, and real-time alerting into a single integrated workflow. All

12 validation test cases passed successfully, confirming the system's functional correctness, security integrity, and integration reliability.

In conclusion, this project demonstrates that modern AI-driven systems combining deep learning, location intelligence, and cloud-based services can substantially improve urban infrastructure management, enhance public road safety, and contribute meaningfully to the development of smarter, safer, and more responsive cities. The system provides a strong foundation for future enhancements including multi-sensor fusion, video-based continuous monitoring, predictive maintenance modelling, and large-scale smart city deployment.

7. REFERENCES

1. Zhaohui Wu, Runjing Zhao, Xingliang Sun, and Zhaojia Li, "Utilize Data Augmentation and Flexi Corner Block for Road Damage Detection", vol. 13, no. 13, p. 72564, May 2025.
2. Darryn Anton Jordan, Stephen Paine, Amit Kumar Mishra and Jan Pidanic, "Road to Repair (R2R): An Afrocentric Sensor-Based Solution to Enhanced Road Maintenance", vol 11, 6010 January 2023.
3. Kang Zhao, Jingjing Liu, Qingnan Wang, Xianjun Wu, and Jihui Tu, "Road Damage Detection From Post-Disaster High-Resolution Remote Sensing Images Based on TLD Framework", vol 10, 43552, April 2022.
4. Amel Ali Alhussan, Doaa Sami Khafaga, El-Sayed M. El-Kenawy, Abdelhameed Ibrahim, Marwa Metwally Eid and Abdelaziz A. Abdelhamid, "Pothole and Plain Road Classification Using Adaptive Mutation Dipper Throated Optimization and Transfer Learning for Self Driving Cars", vol 10, 84188, August 2022.
5. Jiaxiu Dong, Zhaonan Li, Zibin Wang, Niannian Wang, Wentong Guo, Duo Ma, Haobang Hu, and Shan Zhong, "Pixel-Level Intelligent Segmentation and Measurement Method for Pavement Multiple Damages Based on Mobile Deep Learning", vol 9, 143860, October, 2021.
6. Mohammad Alamgeer, Hend Khalid Alkahtani, Mashaal Maashi, Mahmoud Othman, and Anwer Mustafa Hilal, Ibrahim Alsaid, Azza Elneil Osman, And Amani A. Alneil, "Optimal Fuzzy Wavelet Neural Network Based Road Damage Detection", vol 11, 61986, June 2023.
7. Seungbo Shim And Gye-Chun Cho, "Lightweight Semantic Segmentation for Road-Surface Damage Recognition Based on Multiscale Learning", vol 8, 102680, June 2020.
8. Haniah Mahmudah, Aulia Siti Aisjah, Syamsul Arifin, And Catur Arif Prastyantyo, "Iyolov7-Tpe-Ss Leveraging Improved YOLO Model With Multilevel Hyperparameter Optimization for Road Damage Detection on Edge Devices", vol 13, 79247, May 2025.
9. Hussein Samma, Shahrel Azmin Suandi, Nor Azman Ismail, Sarina Sulaiman, And Lee Li Ping, "Evolving Pre-Trained CNN Using Two-Layers Optimizer for Road Damage Detection From Drone Images", vol 9, 158215, November 2021.
10. A. Alrajhi, K. Roy, L. Qingge, And J. Kribs, "Detection of Road Condition Defects Using Multiple Sensors and IoT Technology: A Review", vol 4, 372, June 2023.
11. Abhiram Karukayil, Christopher Quail, And Fernando Auat Cheein, "Deep Learning Enhanced Feature Extraction of Potholes Using Vision and LiDAR Data for Road Maintenance", vol 12, 184541, December 2024.
12. Haiyang Wu, Lingyun Kong, And Denghui Liu, "Crack Detection on Road Surfaces Based on Improved YOLOv8", vol , 190850, December 2024.
13. Mumbere Muyisa Forrest, Zhigang Chen¹, Shahzad Hassan², Ian Osolo Raymond, And Karim Alinani, "Cost Effective Surface Disruption Detection System for Paved and

Unpaved Roads”, vol 6, 48634 , September, 2018. 55

14. Junxin Chena , Xiaojie Yua , Qiankun Lib , Wei Wangc , Ben-Guo Hed, “LAG-YOLO: Efficient road damage detector via lightweight attention ghost module”, vol 11, 9180032, 2023.

15. Prashant Kumar, Apurva Sharma and Solomon Raju Kota, “Automatic Multiclass Instance Segmentation of Concrete Damage Using Deep Learning Model”, vol 9, 90330, June 2021.