CHAPTER 1

INTRODUCTION

Roads are an essential part of everyday life, connecting cities, supporting transportation, and enabling economic activities. However, one of the most common issues affecting road quality is the formation of potholes. Potholes usually occur due to heavy traffic, continuous wear and tear, poor drainage, and climatic conditions such as rainfall. When they are not identified and repaired on time, they lead to several problems including accidents, vehicle damage, slower traffic movement, and inconvenience to the public. Many cities struggle to maintain smooth roads because manual reporting methods are slow, depend on complaints, and often do not reach the concerned authorities in time.

To address this gap, the Web-Based Pothole Detection and Reporting System is designed as a smart, simple, and technology-driven solution. The main purpose of this system is to allow citizens to report potholes easily and help authorities monitor road issues in a structured way. Since it is a web application, users do not need to install any app; they can directly open the website on their mobile or laptop. The user-friendly interface allows them to report potholes by either capturing an image through the device camera or uploading a photo from the gallery. Along with the image, the system also collects the location (GPS coordinates) automatically or allows the user to mark it on a map. This ensures that the reported pothole can be found accurately by the authorities.

A separate Admin Dashboard is provided for municipal or road maintenance teams. Admins can log in securely and view all pothole reports submitted by users. Each report displays the image, location on the map, date, and user description. The admin can then verify the report and update its status as New, In-Progress, or Resolved. This status-tracking feature helps authorities organize and prioritize repair work. Over time, the system also becomes a valuable database that shows which areas have frequent pothole issues, helping in better planning and resource management.

Overall, the Web-Based Pothole Detection and Reporting System creates an effective connection between citizens and authorities. It encourages public involvement in maintaining road quality, increases transparency, and reduces the delay in identifying and fixing potholes. By digitizing the reporting process, the system supports the smart city initiative and contributes to safer and smoother road conditions. This solution demonstrates how simple web technologies can solve real-world civic problems and improve daily life for people.

1.1 AIM AND OBJECTIVE

The aim of this project is to create a simple and efficient web-based system that allows users to report potholes by capturing or uploading images, while providing administrators with the tools to review, track, and resolve these reports.The system is designed to improve road safety, reduce accidents, and help authorities identify and repair damaged road areas more quickly.

I. To provide a simple and user-friendly interface for users to report potholes by using their device camera or by uploading images.

II. To automatically capture the accurate geographical location of the pothole using GPS or map-based services.

III. To maintain a centralized database that stores all reports along with images, location, description, and current status.

IV. To offer an admin dashboard where authorities can view, and manage all reported potholes.

V. To support faster road maintenance by helping authorities prioritize and repair potholes based on real-time data.

1.2 PROBLEM STATEMENT

I. Potholes on roads lead to accidents, vehicle damage, and traffic disruptions, creating serious safety and transportation challenges.

II**.** Existing detection methods rely on manual inspections and citizen complaints, which are slow, inconsistent, and often inaccurate.

III. Many potholes remain unnoticed due to the lack of real-time monitoring and absence of a centralized system for mapping and tracking road conditions.

IV. There is a need for an automated, intelligent system that can detect potholes accurately, capture their GPS location, and report them instantly to authorities for timely repair and improved road management.

CHAPTER 2

LITERATURE SURVEY

I.“ Pothole Detection Application RoadEye”

Year: 2025

Author: M.Reddy, A.Guptha, K.Ramesh

Contribution: This paper presents a user-friendly pothole reporting application called “RoadEye.” The system allows users to capture photos of potholes, upload them through a web/mobile interface, and automatically attach GPS location. The admin dashboard enables verification, status updating, and tracking of repairs. The authors emphasize user participation, making the system suitable for smart-city complaint management. The paper highlights improved communication between citizens and authorities through a structured reporting workflow**.**

II. “Artific Advancements in pothole Detection Techniques: A Comprehensive Review”

Year: *2025*

Authors: H.Singh, P.R. patil

Contribution: This review summarizes recent developments in pothole detection from 2019–2025, focusing on deep learning, image processing, laser scanning, and sensor-based methods. The authors compare different detection models such as YOLO, Faster R-CNN, and segmentation-based networks. The paper concludes that web-based reporting platforms combined with real-time detection can significantly improve road maintenance processes. It also identifies challenges related to dataset quality, lighting conditions, and deployment in real environments.

III. “PDS-UAV: A Deep Learning-Based Pothole Detection System ”

Year:2024

Authors: Othman Alzamzami, Ahmed Al-Delail, Nasser Al-Harthi

Contribution: This research uses drone (UAV) imagery combined with YOLOv8 to detect potholes over large road networks. The system analyzes aerial images, identifies damaged road sections, and sends the results to a web-based interface for authorities. The paper demonstrates that UAV-assisted detection provides higher coverage and reduces time-consuming manual surveys. It also includes a reporting module where detected potholes are stored with coordinates, helping municipal bodies plan repair activities more effectively.

IV. " Pothole Detection using Image Processing and Machine Learning Techniques"

Year:2021.

Authors: R. Kumar, S.Naveen, Priya Sharma

Contribution: This paper discusses various image-processing techniques and machine-learning models used to detect potholes accurately from road images. The authors highlight how camera-based detection can replace manual inspection and allow continuous monitoring. The study compares different algorithms such as edge detection, segmentation, and CNN-based classifiers, showing that a trained model can identify potholes under different lighting and road conditions. The research also emphasizes the need for real-time reporting systems to help road authorities respond faster.

V."*Crowdsourced Road Issue Reporting System Using Web and GIS technologies"*

Year: May 2021, 17 April 2020

Authors: S. Chakravarth T.Mishra

Contribution: This study describes a web-based crowdsourcing system where citizens report road issues—including potholes—using images, descriptions, and location data. The backend integrates GIS mapping to visualize all reports on an interactive dashboard. Admin users can review complaints, update repair status, and prioritize tasks based on severity and frequency. The authors conclude that crowdsourced reporting improves transparency, makes authorities more responsive, and helps build smarter road-maintenance strategies.

CHAPTER 3

PROPOSED SOLUTION

3.1 WEB APP OVERVIEW

- The Pothole Detection and Reporting System is a web-based platform designed to simplify the process of identifying and reporting road potholes by enabling users to submit images along with location details directly through a mobile-friendly website.
- It offers a structured reporting environment that allows administrators to view submitted reports, verify pothole locations, and update the repair status through an organized and interactive dashboard.

3.2 USER-SPECIFIC FEATURES

- **Image Capture and Upload**
  Users can capture pothole images using the device camera or upload existing photos through the web interface for easy reporting.
- **Automatic Location Detection**
  The platform fetches the user's GPS location automatically
- **Simple Report Submission**
  .A user-friendly form lets users add a short description and submit the report

3.3 ADMIN-SPECIFIC FEATURES

- **Structured Report Management:**
  The admin dashboard displays all submitted pothole reports with images, locations, timestamps, and descriptions for quick verification.
- **Status Update Controls**
  Admins can update each report’s status as New, In-Progress, or Resolved
- **Map-Based Visualization:**
  The system provides a map view showing exact pothole locations, helping admins prioritize repairs based on severity or frequency

3.4 DATA PROCESSING AND MANAGEMENT

- All submitted images, locations, and descriptions are stored in a structured database for efficient record-keeping and analysis.
- Helps authorities access complete information instantly.

3.5 REPORTING AND USER INTERACTION

- Provides action Allows users to capture or upload pothole images in web app
- Automatically fetches GPS coordinates

3.6 IMPACT ON ROAD MAINTENANCE

- Enables authorities to detect pothole-prone areas quickly
- Minimizing delays caused by traditional manual reporting systems.
- Reduces the risk of accidents and vehicle damage by allowing timely repairs

3.7 OUTCOMES

- Encourages citizens to contribute actively to road maintenance.
- Administrators can track, verify, and update repair statuses.
- Provides a scalable, technology-driven solution that aligns with modern urban infrastructure goals.

CHAPTER 4

ARCHITECTURE AND FLOWCHART

4.1 ARCHITECTURE

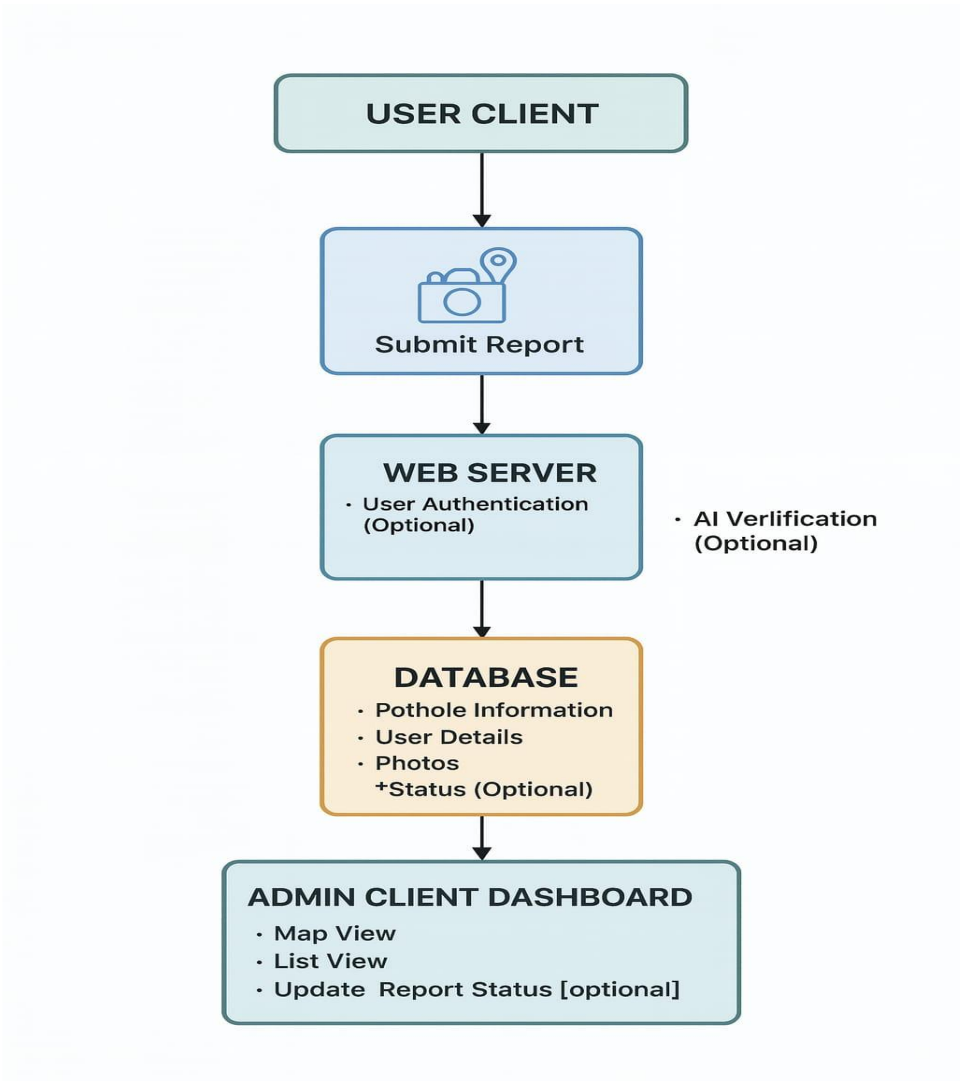


Fig:4.1: System Architecture

The system is built using multiple interconnected components that work together to enable pothole reporting, detection, and admin management. The architecture includes the **User Interface Layer**, **Application Layer**, **Machine Learning Layer**, **Storage Layer**, and **Admin Management Layer**.

- **User Interface Layer (Client-Side)**

  This layer allows users to upload or capture pothole images, view detection results, and Submit reports through a simple web interface. It also handles user interactions, form inputs, and communication with the server via API calls.

- **Application Layer (Flask Server)**

  The application layer manages all system operations, including receiving images, validating requests, triggering detection, sending data to the model, returning results to users, and managing report flow. It acts as the central controller for all communication.

- **Machine Learning Layer (YOLO Model Engine)**

  This layer processes the uploaded images using the trained YOLO model. It performs pothole detection, generates bounding boxes, calculates confidence scores, and creates output images.The detection logic is handled by the `detect.py` script.

- **Storage Layer (File Storage + Database)**

  The system stores raw images in an uploads folder and detection outputs in a results folder. A database (MySQL / SQLite / MongoDB) stores report details such as user info, image paths, detection results, location, and status. This ensures all reports are saved securely and can be retrieved anytime.

- **Admin Management Layer**

  This layer provides administrators with a dashboard to view all reported potholes, check detection images, verify locations, and update the status of reports. It allows admins to manage the entire lifecycle of each complaint from submission to resolution.

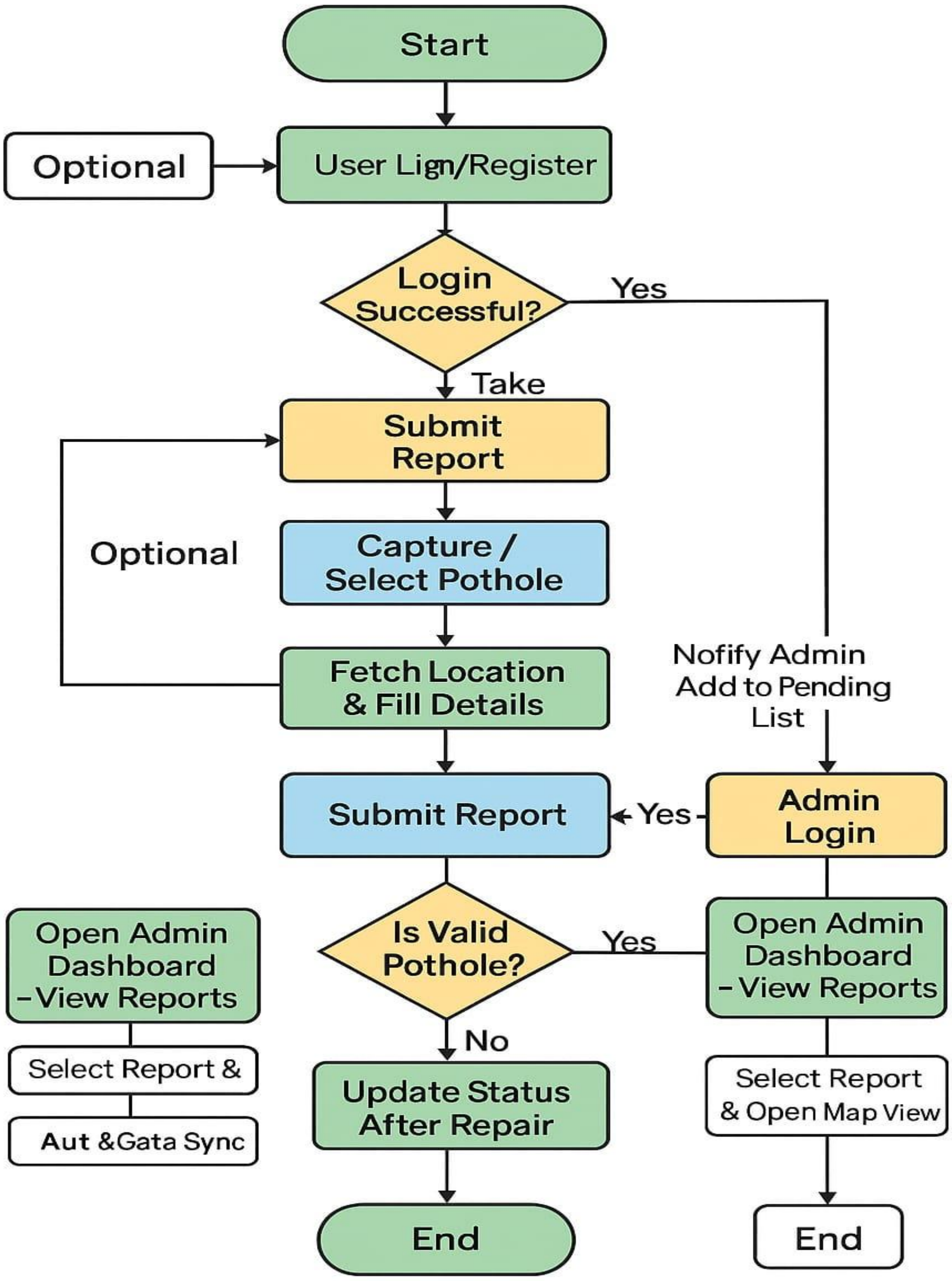
- **Communication Layer (APIs and Routing)**
  APIs enable seamless interaction between frontend, backend, and model. Requests pass through defined routes, ensuring smooth data exchange, processing, and response.

- **Output and Reporting Layer**
  This layer presents detection results, confirmation messages, and status updates to users and admins. It ensures clear communication of detection accuracy and report progress.

The architecture of the Pothole Detection and Reporting System is designed as a simple ,Scalable workflow connecting users, backend services, and municipal administrators.

- The process begins on the user’s smartphone, where the web application allows citizens to initiate a report by capturing or uploading a pothole image along with its GPS location.
- This collected data is then sent to the cloud backend, built using Flask or any similar web API framework, which processes incoming information, validates the report, and forwards it for storage.
- The backend uploads the photo, location, and additional metadata into the database, where pothole reports are permanently stored for future retrieval. Once a new report is saved, the backend notifies the admin dashboard, a web interface used by municipal authorities.
- Admins can view potholes on an interactive map, check the severity from the submitted images, and take required actions.
- Administrators can then update the status of each pothole

4.2 FLOW CHART



Start
Optional
User Lign/Register
Login Successful?
Yes
Take
Submit Report
Optional
Capture / Select Pothole
Fetch Location & Fill Details
Nofify Admin Add to Pending List
Submit Report
Yes
Admin Login
Open Admin Dashboard - View Reports
Is Valid Pothole?
Yes
Open Admin Dashboard - View Reports
Select Report &
Aut &Gata Sync
No
Update Status After Repair
Select Report & Open Map View
End
End

Fig 4.2: Flow Chart

The figure 4.2 illustrates the workflow of the web app

- The flowchart explains how the entire pothole reporting system works from both the user and admin sides.
- It starts with the user logging into the web application to access the reporting features.
- Once logged in, the user can either use the live camera to capture a pothole or upload an existing image from their device.
- After the image is selected, the system automatically collects the user's GPS location, or the user can manually enter it if needed.
- The user then adds a brief description and submits the report through the web app.
- This report is sent to the backend server, where all the information—such as the image, location, and user details—is stored safely in the database.
- At the same time, the admin dashboard receives a notification about the new report.
- On the admin side, the admin logs into the system to review the submitted reports.
- The admin opens each report, checks the uploaded image, and views the pothole location on the map to verify it.
- If the pothole report is valid, the admin updates its status to New, In-Progress, or Fixed based on the road repair progress.
- If the report is not valid, the admin can reject it.
- Once the admin completes the verification and updates the status, the process comes to an end.

The complete working process of the pothole reporting system, showing how both the user and admin interact with the application. It begins with the user logging into the web app to access the reporting features, after which the user can either capture a pothole through the live camera or upload an existing image. Once an image is selected, the system automatically collects the user's GPS location, or the user can manually enter it if needed. The user adds a short description and submits the report, which is then sent to the backend server where the image, location, and user details are securely stored in the database. Simultaneously, the admin dashboard receives a notification about the new report. The admin logs into the system, reviews the submitted report, checks the uploaded image, and verifies the pothole location on the map. Based on the accuracy and validity of the report, the admin marks its status as New, In-Progress, or Fixed, or rejects it if the report is invalid. This single, streamlined flow ensures faster reporting, clearer communication, and more efficient pothole management from submission to verification.

CHAPTER 5

IMPLEMENTATION

5.1 FUNDAMENTAL FEATURE

I. Pothole Reporting

- Users can quickly capture photos using the device’s live camera or upload images from the gallery.
- The system ensures high-quality image submission to help admins verify potholes accurately.
- A short description or additional notes can be added to provide context about the pothole’s severity or location.

II. Automatic Location Detection

- The application automatically fetches the user's GPS coordinates using browser geolocation APIs.
- Users can alternatively select or adjust the pin on an interactive map for precise location accuracy.
- The latitude and longitude data are stored along with the report for backend processing and map display.

III. User-Friendly Interface

- A clean and responsive UI ensures smooth usage on both mobile and desktop devices.
- Clear buttons, simple forms, and guided prompts make reporting quick and hassle-free.
- Accessibility features like readable fonts and intuitive layout enhance usability for all users.

IV. Admin Dashboard

- A centralized web dashboard displays all submitted pothole reports in a structured format.
- Admins can filter, sort, or search reports based on status, date, or location.
- Dashboard widgets or counters show real-time statistics such as total reports, resolved reports pending cases.

V. Status Update System

- Every report goes through predefined stages: **Pending → In-Progress → Resolved**.
- Admins can update the status with one click and optionally add remarks or repair details.
- Status changes trigger notifications or alerts to users (optional feature based on implementation).

VI. Image and Data Storage

- Uploaded photos are stored securely on the server or cloud storage with unique filenames.
- Each report record includes the image URL, location, timestamp, user ID, and description.
- Data integrity is maintained through proper validation and database indexing for fast retrieval.

VII. Map Integration

- Reported pothole locations are displayed on an interactive map using APIs like Google Maps or OpenStreetMap.
- Admins can zoom in/out, view exact coordinates, and analyze multiple reports in a specific region.
- Clustering features can group nearby pothole reports for better visualization in high-density areas.

VIII. Report Tracking

- Users can view the status of their previously submitted reports anytime.
- A timeline or status indicator shows the progress from submission to resolution.
- Admins can monitor unresolved reports and prioritize tasks based on severity or location.

5.2 APPLICATION OF TECHNOLOGY

Technologies Used:

Backend:

Layer	**Technology**	**Purpose**
Web Framework	Flask (Python)	API endpoints, file upload, running YOLO model
AI Model	YOLOv8/YOLOv5	Detect potholes in images or webcam feed
Database	MongoDB or MySQL	(Optional) Store detection logs, image paths, results
Server	Python + Flask	Executes detection, saves results

Frontend:

Component	**Technology**
UI	HTML5, CSS3
Design	Bootstrap
Interactivity	JavaScript
Result preview	Canvas / HTML image preview

Overview :

- The backend uses Python and Flask to manage the main logic of the system.
- It handles tasks like receiving user reports, saving data, and running the pothole detection model (YOLO).
- The database stores everything securely—photos, locations, timestamps, and admin updates.
- The frontend is built using standard web technologies like HTML, CSS, Bootstrap, and JavaScript, making the interface responsive and easy to use on both mobile and desktop.
- The system also includes an image preview feature so users can see the picture before submitting their report.

Folder Structure

```
SmartPotholeDetection
|── backend/
│   ├── app.py          (Flask backend)
│   ├── detect.py        (YOLO detection logic)
│   ├── requirements.txt
│   ├── static/
│   │   ├── results/      (YOLO output images)
│   └── uploads/         (Uploaded images)
│
|── frontend/
│   ├── index.html
│   ├── style.css
│   ├── script.js
│   ├── logo.png
│
|── models/
│   └── best.pt          (Trained YOLO model)
```

The project is organized into three main sections. The **backend** folder contains the Flask application (`app.py`), the YOLO detection script (`detect.py`), a `requirements.txt` file, and directories for storing uploaded images and detection results. The **frontend** folder includes the main interface files such as `index.html`, `style.css`, `script.js`, and supporting assets like `logo.png`. Additionally, a separate **models** folder is used to store the trained YOLO model file (`best.pt`) required for pothole detection.

5.3 WORKFLOW

Here is an extended **workflow**

I. The user opens the web application and accesses the reporting interface.

II. The user captures or uploads an image of the pothole.

III. The image is sent to the backend and stored in the uploads folder.

IV. The backend triggers the YOLO model to analyze the uploaded image.

V. The model detects potholes and generates an output image with markings.

VI. The processed output is saved in the results folder.

VII. The backend sends the detection result and confidence data back to the frontend.

VIII. The user views the detection output and confirms the pothole report.

IX. The system stores the report details, including image, detection result, and metadata.

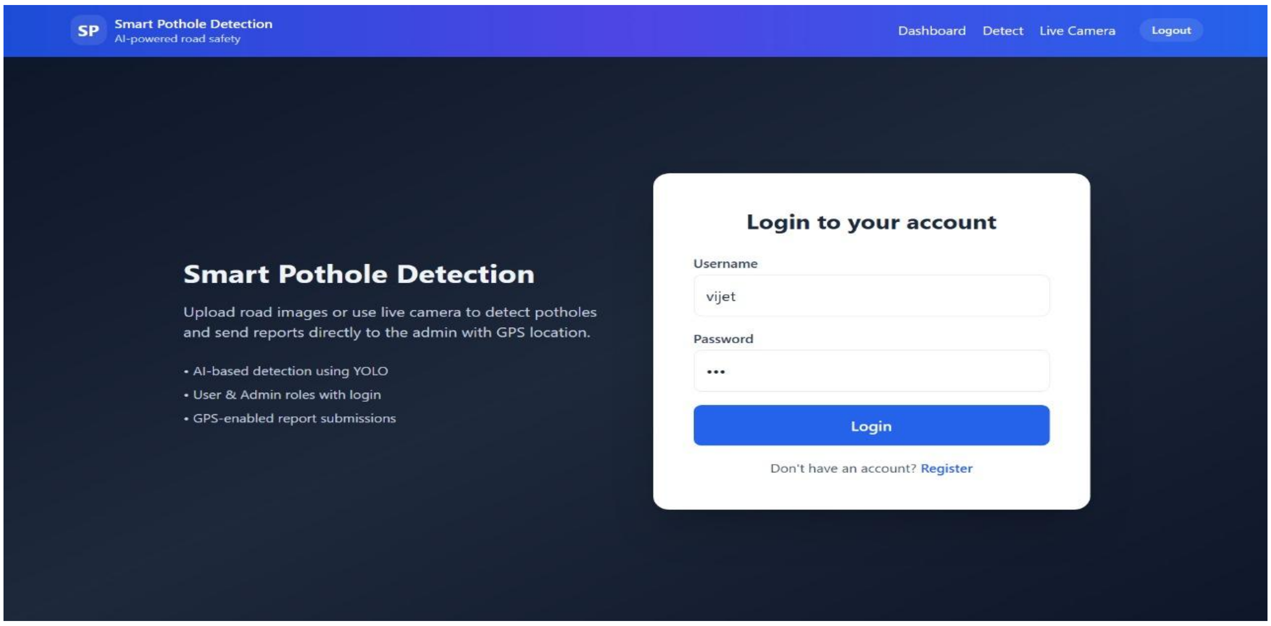
X. The admin accesses the dashboard to review submitted reports.

XI. The admin checks images, detection results, and location information.

XII. The admin updates the report status as Pending, In-Progress, or Resolved.

CHAPTER 6

SCREENSHOTS



Smart Pothole Detection

Upload road images or use live camera to detect potholes and send reports directly to the admin with GPS location.

- AI-based detection using YOLO
- User & Admin roles with login
- GPS-enabled report submissions

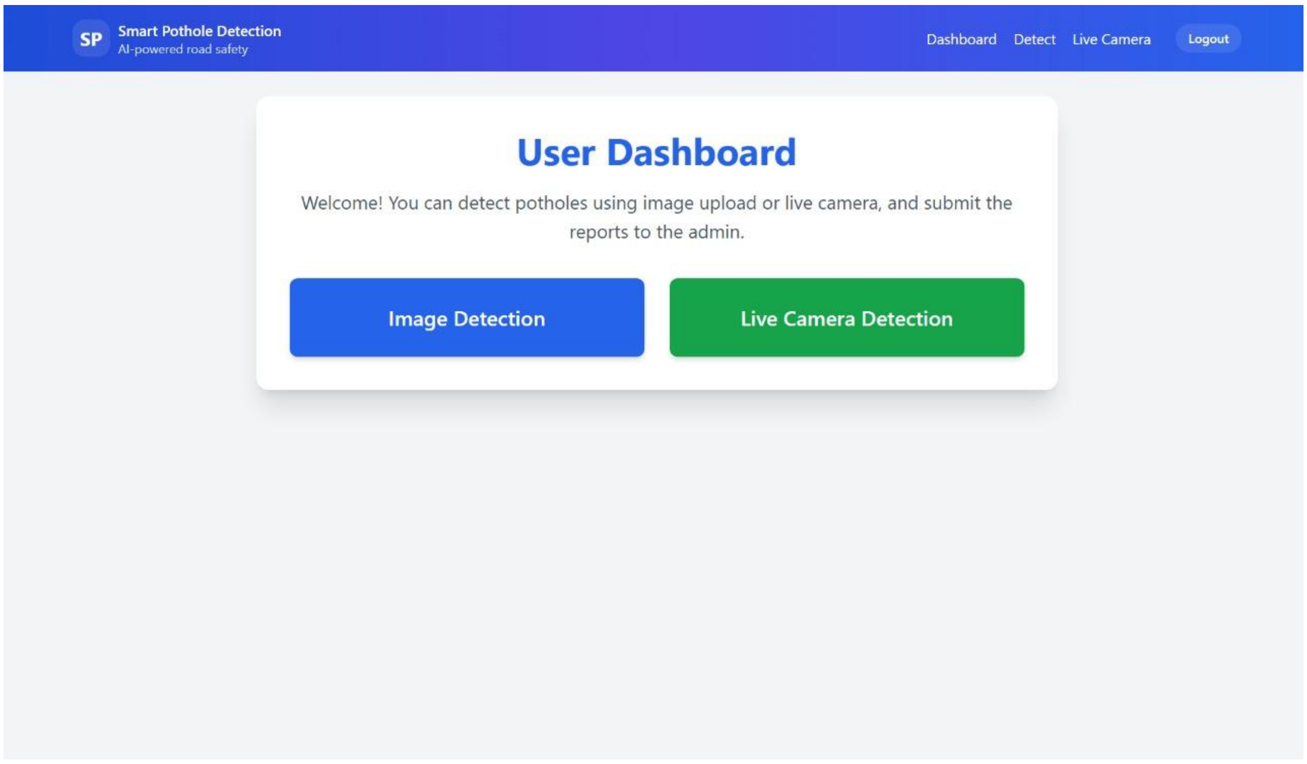
Login to your account

Username: vijet

Password: •••

Fig:6.1:User Login

The Figure 6.1, shows The user login system allows registered users to securely access the pothole reporting platform. Each user logs in with their email and password, which are verified through the backend before granting access. After successful login, the user can submit pothole reports, upload images

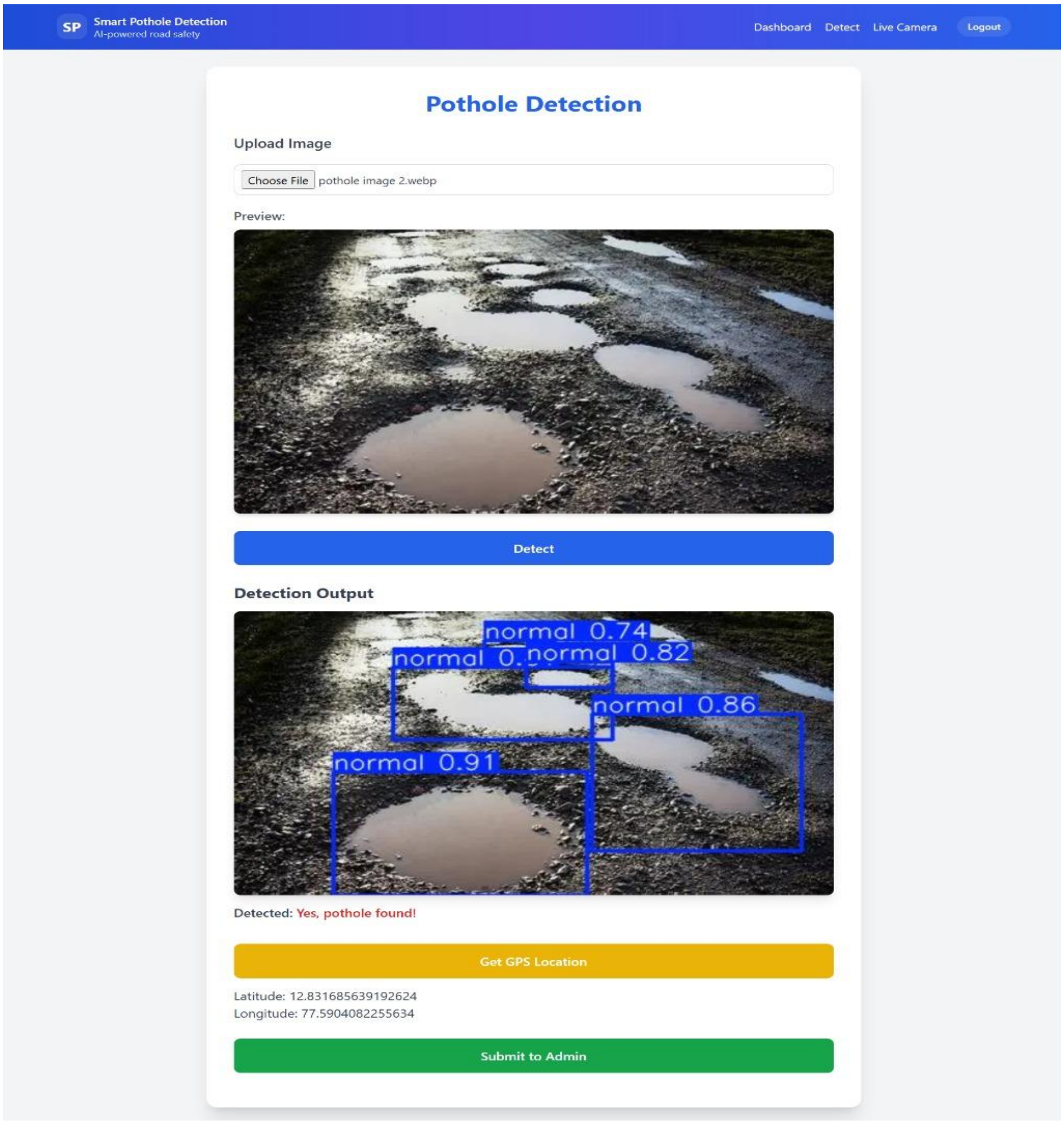


User Dashboard

Welcome! You can detect potholes using image upload or live camera, and submit the reports to the admin.

Fig 6.2: User Dashboard

The Figure 6.2, shows the user dashboard provides an easy-to-use interface where users can report potholes either by using the **live camera** to capture real-time images or by **uploading existing photos** from their device. It allows users to view detection results instantly and submit reports with location details. The dashboard also shows a list of all previously submitted reports along with their current status, helping users track progress and stay updated on the issues they have reported.



Detected: Yes, pothole found!

Latitude: 12.831685639192624
Longitude: 77.5904082255634

Fig 6.3: Pothole Detection

The user dashboard allows users to capture images through the live camera or upload photos,view the pothole detection results, check the location on the map, and submit the final report. It provides a simple and clear interface for completing the detection and reporting process.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 CONCLUSION

The Pothole Detection and Reporting System provides a modern and practical way to deal with potholes that often stay unnoticed or unrepaired for long periods. By allowing users to capture photos through a live camera, upload images, and automatically detect potholes with accurate location mapping, the system makes reporting much easier and more reliable. It removes the complications of traditional reporting methods and gives people a quick way to share problems they see on the road.

All reports include images, detection results, and exact locations, which helps authorities understand the issue clearly. The user dashboard allows individuals to review the detection output, check the map location, and submit the report with confidence. At the same time, the dashboard helps officials access reports, verify the details, and update the status as the repair progresses.

This creates a smooth and organized process for both users and administrators.Overall, the project helps bridge the gap between citizens and road authorities. It encourages faster action, better communication, and safer roads. By using simple technology in a meaningful way,the system shows how digital tools can make everyday problems easier to report and solve, ultimately helping create better road conditions for everyone.

7.2 FUTURE ENHANCEMENTS

I. **Real-Time Mobile App Integration**
Develop a dedicated Android/iOS app for faster and on-the-go pothole reporting.

II. **Automatic GPS Tracking While Driving**
Enable background detection using the phone camera and sensors to auto-detect potholes during travel.

III. **Improved Detection Using Video Streams**
Allow continuous pothole detection from dashcams or CCTV footage instead of single images.

IV. **Severity Level Classification**

Enhance the model to identify pothole severity (minor, moderate, severe) to help authorities prioritize repairs.

V. **Automated Notifications**

Send SMS or email alerts to users when their reported pothole is reviewed or resolved by admin.

VI. **Integration with Government Systems**

Connect the platform with municipal or smart city portals for automated task assignment to repair teams.

VII. **Heatmap Visualization**

Display a heatmap of frequently damaged areas to help authorities identify high-risk zones.

VIII. **Repair Progress Tracking with Photos**

Allow admins to upload repair progress images so users can see updates visually.

IX. **Multi-Language Support**

Add language options to make the system accessible to more users.

X. **Analytics Dashboard**

Provide detailed analytics for authorities, such as number of reports, repair time, and location statistics.

In the future, the Pothole Detection and Reporting System can be enhanced with several advanced features to make it more efficient and user-friendly. A mobile application could be introduced to allow users to report potholes instantly while on the move, and real-time GPS tracking could help automatically detect potholes as vehicles travel. The detection model can be improved to classify pothole severity, enabling authorities to prioritize repairs more effectively. Additional features like automated notifications, repair progress updates, heatmaps showing high-risk areas, and integration with municipal systems would further improve communication and streamline the repair process. Multi-language support and a detailed analytics dashboard can also be added to make the platform more accessible and useful for both citizens and authorities. These enhancements would strengthen the system's capability to ensure safer roads and faster issue resolution

REFERENCE

[1] Flask Documentation. (2024). *Flask Web Framework*. Retrieved from https://flask.palletsprojects.com

[2] OpenCV Documentation. (2024). *OpenCV Library for Image Processing*. Retrieved from https://docs.opencv.org

[3] Leaflet Maps Documentation. (2024). *Leaflet JavaScript Library for Interactive Maps*. Retrieved from https://leafletjs.com

[4] Ultralytics YOLO Documentation. (2024). *YOLO Model Training and Inference Guide*. Retrieved from https://docs.ultralytics.com

[5] Google Maps Platform Documentation. (2024). *Location and Geocoding APIs*. Retrieved from https://developers.google.com/maps

[6] Ultralytics. (2023). YOLOv8: State-of-the-Art Real-Time Object Detection. Available at**: https://github.com/ultralytics/ultralytics**

[7] Glenn Jocher, Ultralytics. (2023). YOLOv8 Documentation. Available at**: https://docs.ultralytics.com**

[8] Jocher, G., Chaurasia, A., & Qiu, J. (2023). YOLOv8 – Ultralytics YOLO Series. Ultralytics Technical Report.

[9] Ultralytics. (2023). YOLOv8 Release Notes & Model Details. Available at: https://docs.ultralytics.com/models/yolov8

[10] Waze(Community Reporting Research). (2019). *Crowdsourced Road Hazard Reporting System*.