REAL-TIME ROAD ANOMALY DETECTION FROM WEBCAM FOOTAGE USING RASPBERRY PI 5

Project Report Submitted for

Bharat AI-SoC Student Challenge

Submitted by

Adhithya P M
Anaswara Jathish
Jishnu Raveendran

Guided by

Dr. Sajith K.

February 2026

ACKNOWLEDGMENTS

We would like to thank the Arm Bharat AI-SoC Student Challenge Organizing Team for selecting us to work on Problem Statement 03, “Real-Time Road Anomaly Detection from Dashcam Footage on Raspberry Pi.” This project has given us the opportunity to learn more about real-world applications of AI-SoC design and edge computing.

We would like to thank Dr. Sajith K, Department of Electronics and Communication Engineering, Government College of Engineering Kannur, for his support and guidance in the development of this project. His knowledge and expertise in embedded systems and AI optimization have been instrumental in the successful implementation of our project on the ARM platform.

We would also like to thank the Department of Electronics and Communication Engineering, Government College of Engineering Kannur, for their support and facilities in the completion of this project.

Finally, we would like to thank all the members of our team for their hard work and contributions. The successful completion of this project would not have been possible without the help of all of them.

ABSTRACT

Road anomalies can be referred to as irregularities or defects on the surface of the road that may have an impact on the safety of vehicles and the comfort of drivers. Some of the road anomalies include potholes, cracks, uneven road surfaces, speed breakers, manholes, bumps, and road wear. It is important to ensure that the roads are in good condition and that there are no anomalies on the surface of the road in order to avoid impending accidents, hence saving lives.

This project details the design and implementation of a real-time road anomaly detection system using Raspberri Pi and webcam. The system captures live video input and processes each frame with computer vision techniques to identify and detect road anomalies effectively. The Raspberry Pi serves as a compact and cost effective processing unit, allowing analysis without the need for complex hardware

The implementation uses the Python programming language and the OpenCV library. The system continuously analyses the captured video frames, highlighting detected objects with bounding boxes and labels. It is designed to work with minimal delay, ensuring smooth real-time performance. The proposed system shows reliable detection accuracy under normal lighting and can be used in various practical situations like monitoring systems, automated security, and intelligent surveillance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Damage to road surfaces is a significant infrastructure issue, especially in developing nations like India, where rapid urbanisation, high traffic volumes, and scarce maintenance budgets cause roads to deteriorate faster. Potholes, cracks, and rough road surfaces not only affect the driving experience but also cause vehicle damage, traffic congestion, and accidents, which in turn affect economic efficiency and sustainability. Conventional road damage detection involves manual road surveys or the use of special vehicles with costly sensors. These processes consume labour and time, and often incapable of providing continuous, real-time monitoring.

The development of embedded systems, computer vision, and edge computing has enabled the creation of intelligent, low-cost solutions for autonomous road damage detection. Edge computing components like the Raspberry Pi have enough processing power to support real-time image processing, which is cost-effective and energy-efficient. By combining a webcam with a Raspberry Pi, images of road surfaces can be taken continuously as a vehicle is in motion.

The proposed system utilizes computer vision algorithms and lightweight deep learning models to analyse video frames in real time. Techniques such as image pre-processing, edge detection, feature extraction, and convolutional neural networks can be employed to identify potholes, manholes, speed breakers and other surface irregularities. Running the model locally on the Raspberry Pi reduces dependency on cloud connectivity, lowers latency, and enhances privacy.

The purpose of this project is to design and develop a low-cost, portable, and scalable road anomaly detection system using embedded hardware and computer vision. The proposed system aims to address the shortcomings of conventional inspection techniques by offering continuous monitoring, rapid detection, and enhanced road safety. Additionally, the system helps to make infrastructure management and smart city development by turning conventional vehicles into smart monitoring units.

CHAPTER 2

LITERATURE SURVEY

Road anomaly detection has evolved significantly with the adoption of machine learning and deep learning techniques. Early research focused on image segmentation and feature-based approaches for pothole detection. Aparna et al. [1] proposed machine learning and deep learning based segmentation methods for detecting potholes in urban roads, highlighting the importance of robust feature extraction under varying lighting conditions.

Lakshmi and Rajagopal [2] further explored Machine Learning and Deep Learning based image segmentation approaches for urban pothole detection, demonstrating CNNs.

With the advancement of real-time object detection models, YOLO-based architectures became prominent. Chaithra et al. [3] proposed YOLOv5s-M for pavement damage detection from street-view imagery, achieving high detection accuracy with real-time performance. Similarly, Bhargava et al. [4] implemented automated pothole detection using YOLOv5 and demonstrated improved detection speed and precision suitable for intelligent transportation systems.

Deep learning classification methods were also explored by Mehta and Singh [5], who applied CNN-based classification for pothole detection, achieving robust performance across different road textures.

Lee et al. [6] extended pothole detection by incorporating dimension estimation using in-vehicle technologies, emphasizing practical deployment in real-world driving scenarios.

Advanced learning strategies such as super-resolution and semi-supervised learning were introduced by Chen and Li [7], who utilized GAN-based models for improved road damage detection. Similarly, Zhang and Wei [8] proposed a lightweight YOLO-GAN-based approach for road surface disease recognition, highlighting computational efficiency.

Thermal imaging-based detection was investigated by Kaur et al. [9], where CNN models were used to improve detection accuracy under challenging environmental conditions. Multi-modal fusion techniques combining visible and infrared images were proposed by Patel and Desai [10], enhancing robustness in diverse lighting environments.

Participatory sensing approaches such as PotSpot were introduced by Sharma and Kapoor [11], integrating deep learning with crowd-sourced monitoring systems for scalable pothole detection.

Despite significant advancements, most studies focus primarily on potholes. Limited work addresses multi-class anomaly detection including manholes and speed breakers under real-time constraints on edge devices. Therefore, this project emphasizes lightweight, real-time multiclass detection on Raspberry Pi using optimized deep learning models.

CHAPTER 3

PROBLEM STATEMENT

In recent years, road safety has become a serious concern, especially due to poor road conditions. Common issues such as potholes, damaged manholes, and unmarked speed breakers often go unnoticed until they cause accidents or vehicle damage. These problems are particularly common during rainy seasons and in high-traffic areas.

Currently, road inspection is mostly done manually. Officials physically inspect roads and record damages, which is time-consuming and not always efficient. In many cases, repairs are delayed because damage is not detected early. Although some automated systems exist, they usually depend on expensive equipment or cloud based processing, making them difficult for large scale implementation.

With the advancement of embedded systems and edge AI, it is now possible to develop compact and affordable solutions that can perform intelligent processing directly on hardware devices. Raspberry Pi, which is based on ARM architecture, provides a low-cost platform capable of handling lightweight deep learning models.

The core objective of this project is to develop a real-time system that can detect road anomalies — specifically potholes, manholes, and speed breakers — using dashcam footage and process the data directly on a Raspberry Pi. The system should work continuously, analyze live video frames, and identify anomalies with reasonable accuracy while operating within limited hardware resources.

Several challenges need to be addressed while solving this problem. Raspberry Pi has limited processing power compared to high-end GPUs, so the deep learning model must be optimized carefully. The system also needs to perform well under different lighting conditions, shadows, and varying road textures. Since this is a real-time application, maintaining a good balance between speed and accuracy is very important.

Through this project, we aim to design a practical, cost-effective, and efficient solution that demonstrates how AI can be deployed on an ARM-based System-on-Chip platform for real-world applications. The final outcome should be a working prototype capable of detecting road anomalies in real time and generating alerts without relying on cloud processing.

CHAPTER 4

PROPOSED SYSTEM

4.1 Methodology

The proposed system is a real-time road anomaly detection framework developed to identify potholes, manholes, and speed breakers using video input captured through a USB webcam connected to a Raspberry Pi. Instead of relying on a commercial dashcam, a standard USB webcam is mounted on a vehicle to capture continuous road footage. The system leverages the YOLOv8 object detection model for accurate and efficient multiclass detection.

4.2 System Architecture

The overall architecture of the proposed system is divided into five primary modules:

1. Video Acquisition Module
2. Frame Preprocessing Module
3. YOLOv8 Detection Module
4. Inference and Visualization Module

4.2.1 Video Acquisition Module

A USB webcam is mounted in a forward-facing position to capture continuous road footage. The webcam is connected directly to the Raspberry Pi via USB interface. Video frames are captured using the OpenCV library in real time.

The captured video stream is processed frame-by-frame. The resolution is configured to balance detection accuracy and computational efficiency.

4.2.2 Frame Preprocessing Module

Before passing each frame to the detection model, preprocessing operations are performed to ensure compatibility with the trained YOLOv8 model. The preprocessing steps include:

- Resizing frames to the required input dimension (e.g., 640×640 or lower for faster inference)
- Conversion of color space (BGR to RGB if required)
- Normalization of pixel intensity values

These operations help reduce computational load while maintaining detection quality.

4.2.3 YOLOv8-Based Detection Module

The core of the proposed system is the YOLOv8 object detection model. YOLOv8 is an advanced single-stage object detection framework known for:

- High detection accuracy
- Anchor-free detection mechanism
- Improved feature extraction backbone
- Efficient real-time performance

The model was trained on a custom dataset containing annotated images of potholes, manholes, and speed breakers. Each object in the dataset was labeled using bounding boxes and class identifiers.

To ensure compatibility with the Raspberry Pi environment, a lightweight variant of YOLOv8 was selected. The model input resolution and batch size were carefully chosen to balance inference speed and detection performance.

4.2.4 Model Deployment and Optimization

Since Raspberry Pi is a resource-constrained embedded device, model optimization was necessary to achieve stable real-time performance. The following strategies were applied:

- Use of a lightweight YOLOv8 variant
- Reduction of input image resolution
- CPU-based inference optimization
- Efficient memory management

The trained model was exported in an optimized format suitable for embedded inference. The inference pipeline was configured to minimize latency while maintaining acceptable detection accuracy.

4.2.5 Inference and Output Module

During execution, each preprocessed frame is passed to the YOLOv8 model for inference. The model outputs:

- Bounding box coordinates
- Class labels (Pothole, Manhole, Speed Breaker)
- Confidence scores

The detected anomalies are visualized on the output frame using bounding boxes and text labels. The annotated frame is displayed in real time on the Raspberry Pi.

4.2.6 Working Procedure

The overall working procedure of the proposed system is summarized as follows:

1. Capture live video using USB webcam.
2. Extract video frames sequentially.
3. Preprocess each frame.
4. Perform YOLOv8 inference.
5. Classify detected objects.
6. Display annotated output.
7. Generate alert if required.
8. Repeat the process continuously.

CHAPTER 5

DATASET DETAILS

5.1 Overview

The performance of any deep learning-based object detection system largely depends on the quality and diversity of the training dataset. For this project, a custom dataset was prepared to train the YOLOv8 model for detecting road anomalies, specifically potholes, manholes, and speed breakers.

The dataset was designed to represent real-world road conditions with variations in lighting, texture, and background complexity to improve the robustness of the model.

5.2 Data Collection

The dataset was generated using a USB webcam mounted on a Raspberry Pi 5, which was placed in a stable position to record road footage in the forward-facing direction. The video recordings were obtained in different environmental settings to ensure that the model generalizes well in real-world applications. The data generation process involved obtaining video recordings in both daytime and evening lighting conditions, as well as along urban and semi-urban roads. It is important to note that the data generation process involved roads with different textures and materials, including asphalt and concrete roads, to expose the model to different visual patterns. Additionally, the camera was placed at different distances and angles to address the effects of perspective that may be encountered in real-world applications involving actual vehicle movement. The recorded videos were then processed to generate individual image frames, after which a rigorous filtering process was conducted to eliminate blurred, low quality, and redundant images.

5.3 Dataset Composition

The final dataset is a collection of images with annotations of the various classes of road anomalies that have been identified in this project. Each image may contain one or more road anomalies depending on the image being captured. For each anomaly in an image, bounding boxes were placed to precisely locate the object and label it accordingly. This was done to ensure accuracy in the labeling process and to enhance the reliability of the training dataset.

To ensure that the training process is effective, the dataset was split into three portions: training, validation, and testing. Approximately 70% of the images were used for training, which was necessary to enable the model to learn features and patterns associated with potholes, manholes, and speed breakers. About 20% of the images were used for validation, which was necessary to ensure that the model is not overfitting as it is being trained. The remaining 10% of the images were used for testing, which was necessary to ensure that the performance of the model is accurate as it is being used to predict the performance of the model.

5.4 Data Annotation

All images were annotated using a standard object detection labeling tool. Each anomaly instance was labeled with class ID.

The annotations were saved in YOLO format, which includes normalized bounding box coordinates relative to image dimensions.

5.5 Data Preprocessing

Before training the YOLOv8 model, some preprocessing techniques were employed to ensure that the dataset was standardized and ready for learning. The first step involved resizing all images to conform to the required dimensions for model training. This was important for ensuring that the dataset was standardized and that the model was computationally efficient during training. The second step involved the removal of blurred images to ensure that the model did not learn any erroneous features. Finally, the annotations were checked to ensure that the bounding boxes were correctly positioned.

5.6 Data Augmentation

In order to improve the generalization capability of the model and avoid overfitting, data augmentation methods were employed during the training process. These methods artificially expanded the diversity of the dataset without requiring any new data to be collected. Horizontal flipping was employed to simulate the effect of viewpoint variation, while random changes in brightness and contrast were used to train the model to handle varying lighting conditions. Small angle rotations were also introduced to simulate the effect of slight camera tilt or uneven road perspectives, and scaling and cropping operations were conducted to improve the model's capability to detect anomalies from different distances. By adopting these data augmentation methods, the dataset was made more representative of real-world driving scenarios, including the effects of shadows, lighting changes, and small perspective changes.

5.7 Challenges in Dataset Preparation

During the process of preparing the dataset, a number of real-world issues were encountered. One of the most important issues was the similarity in appearance between manholes and potholes, as both tend to be circular or dark patches on the road surface. This made it more likely that the two would be confused with each other during the process of annotation and training of the model. Another issue was that the speed breakers tended to be very varied in terms of shape, size, and markings, especially in semi-urban regions where uniform markings were not always used. Another issue was related to the effects of shadows cast by trees, vehicles, and other structures in the vicinity, which sometimes produced the illusion of edges or boundaries that resembled road defects. The effect of motion blur introduced during the movement of vehicles also impacted the clarity of images in some frames.

5.8 Summary

The custom dataset created for this project has been designed to offer a wide range of representative examples of road anomalies. Through the addition of lighting, road texture, and viewing conditions, the dataset is able to train the YOLOv8 model to detect road anomalies in real-time using distinctive features. The organized dataset split and augmentation techniques help to ensure that the model generalizes well when deployed on Raspberry Pi 5.

CHAPTER 6

MODEL DETAILS

6.1 Model Selection

For the proposed road anomaly detection system, the YOLOv8 (You Only Look Once version 8) object detection model was chosen because of its excellent balance between accuracy and real-time processing capabilities. Since the proposed system is implemented on a Raspberry Pi 5, computational efficiency was an important factor in this choice. YOLOv8 is a single-stage object detection model that is capable of localizing and classifying objects in a single pass of the network. This makes it much faster compared to two-stage object detection models.

6.2 Architecture Overview

The YOLOv8 architecture is composed of three main components: the backbone, the neck, and the detection head.

The backbone is responsible for feature extraction. It processes the input image using convolutional layers to capture low-level and high-level features such as edges, textures, and shapes. These features help the model differentiate between potholes, manholes, and speed breakers under varying lighting and road conditions.

The neck performs multi-scale feature fusion. It combines feature maps from different layers of the backbone to enable detection of objects at multiple scales. This is particularly important in road scenarios where anomalies may appear small at a distance and larger as the vehicle moves closer.

The detection head generates the final predictions. It outputs bounding box coordinates, object confidence scores, and class probabilities for each detected anomaly. This allows the system to both locate and classify road defects accurately.

6.3 Working Mechanism

YOLOv8 processes the entire image at once and predicts multiple bounding boxes along with their corresponding class labels. During training, the model learns to minimize a combined loss function that accounts for bounding box regression, object confidence, and classification accuracy.

During inference, Non-Maximum Suppression (NMS) is applied to remove duplicate or overlapping bounding boxes. Only the predictions with the highest confidence scores are retained, ensuring clean and reliable detection outputs.

6.4 Model Variant and Optimization

To ensure smooth deployment on the Raspberry Pi 5, a lightweight variant of YOLOv8 was selected. This reduced model size helps in lowering memory usage and improving inference speed while maintaining acceptable detection performance.

Additionally, the input resolution during inference was optimized to balance accuracy and processing speed. This adjustment enables the system to achieve near real-time detection performance without requiring dedicated GPU hardware.

6.5 Suitability for the Proposed System

YOLOv8 was chosen because it provides a practical trade-off between accuracy and computational efficiency. Its ability to detect objects at multiple scales, combined with its single-stage architecture, makes it highly suitable for real-time road anomaly detection using a webcam-based setup on embedded hardware.

CHAPTER 7

TRAINING AND TESTING RESULTS

7.1 Training Performance

The YOLOv8 model was trained on the custom dataset prepared with annotated images of potholes, manholes, and speed breakers. During the training phase, the model learned to localize and classify road anomalies by minimizing the combined loss function, which includes localization loss and classification loss. Validation was carried out at regular stages to check the performance of the model and avoid overfitting. The validation accuracy showed a steady improvement, which ensured that the model generalized well to the unseen data. The use of data augmentation methods played a major role in improving the robustness of the model to varying lighting conditions.

The testing was carried out in actual road environments including urban streets, semi-urban roads, and uneven rural roads. The system processed video frames continuously and generated bounding boxes with class labels and confidence scores in real time. The system was tested on roads containing multiple potholes of varying sizes and depths. The model successfully detected several potholes simultaneously within a single frame, even when they appeared at different distances and angles. Confidence scores ranged from moderate to high (approximately 0.50 to 0.91), depending on lighting conditions and object clarity. Larger and clearly visible potholes produced higher confidence scores.

Fig. 7.1. Multiple Pothole Detection on Uneven Rural Road Surface

One of the key evaluation scenarios involved distinguishing between visually similar objects such as manholes and potholes. The system demonstrated effective classification performance by correctly labeling both objects in the same frame.

The model maintained good confidence levels (above 0.75 in most cases), indicating strong feature extraction and classification capability.

The system was further evaluated for detecting speed breakers under natural daylight conditions. Even with painted road markings and surrounding background variations, the model accurately localized the speed breaker and assigned a high confidence score (around 0.83).

This confirms the model's ability to identify structured road obstacles in addition to surface damage.

The system was also tested in scenarios where road anomalies appeared alongside other road elements such as vehicles, road markings, and shadows. The model successfully detected manholes even when partially surrounded by damaged pavement or irregular textures.

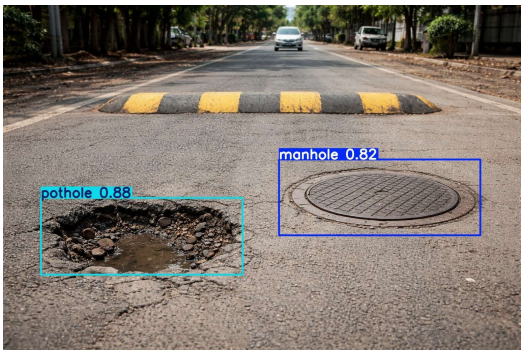

Fig. 7.2. Simultaneous Detection of Pothole and Manhole on Urban Road

Fig. 7.3. Real time Detection of Speed Breaker under Daylight Conditions

Although minor confidence variations were observed due to lighting and perspective distortion, the system consistently localized the anomaly with acceptable accuracy.

THe road anomaly detection system proposed in the paper showed stable and reliable results in various road environments. The model was tested in natural daylight conditions, with moderate shadows from trees, vehicles, and other road infrastructure. In most scenarios, the YOLOv8 model was able to detect potholes, manholes, and speed breakers with high accuracy. The bounding boxes were properly localized, and the confidence levels were within an acceptable range, signifying proper feature extraction and classification capabilities.

It was noticed that larger and more prominent anomalies had higher confidence levels than smaller and partially damaged ones. Objects with clear edges, high contrast, and less obstruction were more confidently detected. This is a clear indication of the model's capability to properly learn spatial and texture-based features during training. However, smaller potholes or anomalies at a farther distance from the camera occasionally showed slightly lower confidence levels.

Fig. 7.4. Detection of Manhole with Surrounding Surface Damage

A slight drop in detection confidence was noticed in more challenging scenarios such as motion blur, uneven illumination, and extreme shadow areas. Fast camera motion and varying illumination intensity slightly impacted feature clarity, thereby affecting prediction confidence. However, the model was still able to detect most anomalies correctly, signifying robustness in dynamic real-world environments.

CHAPTER 8

HARDWARE IMPLEMENTATION

8.1 Overview

The hardware configuration of the proposed system is designed to support real-time road anomaly detection using an embedded edge computing platform. The system is built around the Raspberry Pi 5 and a USB webcam for video acquisition. The selected hardware ensures improved performance while maintaining portability and low power consumption.

8.2 Raspberry Pi 5

The Raspberry Pi 5 is the central processing unit of the system. It is an ARM-based single-board computer that provides a much-improved level of performance compared to its earlier versions, making it ideal for real-time deep learning inference tasks.

The key specifications of Raspberry Pi 5 include:

- Quad-core 64-bit ARM Cortex-A76 processor
- Clock speed up to 2.4 GHz
- 4GB/8GB LPDDR4X RAM (depending on configuration)
- Dual USB 3.0 ports and USB 2.0 ports
- Dual HDMI output
- PCIe interface support
- Improved power management and thermal design

The Raspberry Pi 5 was chosen because of its several key advantages that match the requirements for real-time object detection. It provides much better CPU processing power compared to the previous versions, making it capable of running YOLOv8 inference efficiently on an embedded platform. The enhanced memory bandwidth enables faster data transfer between the processor and memory, which is required for processing continuous video frames in real-time object detection. Moreover, the Raspberry Pi 5 offers better thermal management performance during continuous operation, ensuring smooth functioning during prolonged testing and deployment. The compatibility of the Raspberry Pi 5 with Linux-based operating systems enables smooth integration with popular AI and deep learning libraries. Additionally, the small form factor and power-efficient design of the Raspberry Pi 5 make it a suitable choice for edge computing applications, where portability and power efficiency are critical factors. The device runs a Linux-based operating system, enabling seamless integration with Python, OpenCV, and the YOLOv8 framework.

8.3 USB Webcam

A standard USB webcam is used for capturing live road footage. The webcam is mounted in a forward-facing position to monitor road conditions.

Features of the webcam include:

- Real-time video streaming capability
- Adjustable resolution (e.g., 640×480 or 1280×720)
- Plug-and-play USB connectivity

The webcam is interfaced with the Raspberry Pi 5 through a USB 3.0 port to ensure stable and high-speed data transfer. Video frames are captured using the OpenCV library.

8.4 Power Supply

The Raspberry Pi 5 is powered using a regulated 5V USB-C power supply. Since deep learning inference can generate sustained CPU load, a stable power source is essential for reliable operation.

For testing and portability, the system can also be powered using a high-capacity power bank that supports the required voltage and current specifications.

8.5 Hardware Integration

The hardware components of the system are interconnected in a simple and efficient way. The USB webcam is directly connected to the Raspberry Pi 5 using a USB 3.0 port, which allows for high-speed data transfer necessary for continuous video streaming. A USB-C power supply is used to ensure stable power supply to the Raspberry Pi 5. All computational tasks, such as image processing and YOLOv8 model inference, are carried out on the Raspberry Pi 5. The system is a completely self-contained edge device that does not depend on any external servers or cloud computing resources for computational support.

The chosen hardware configuration has several benefits that make it suitable for real-time road anomaly detection with YOLOv8. The improved CPU design of the Raspberry Pi 5 is one of the factors that enhance the processing speed of the CPU, making it efficient for video frame processing in object detection applications. The system has the capability to process continuous data streams without any delays. Moreover, the device maintains consistent performance even under continuous processing, which is a critical requirement for real-time systems that need uninterrupted processing.

The Raspberry Pi 5 has a compact design that ensures the entire system is portable and easy to implement in real world applications. The device is compact enough to be seamlessly integrated into vehicle based systems or other edge systems. Moreover, the hardware configuration is a cost-effective solution for implementation compared to traditional high performance computing systems, making it suitable for implementation in academic and small scale applications.

CHAPTER 9

OPTIMISATION TECHNIQUES

To ensure that the system operated smoothly and efficiently on the Raspberry Pi 5, several optimization methods were applied during the development process. The main objective was to achieve accurate and real time detection of road anomalies while maintaining steady system performance. The detection model was refined to make it lighter and more suitable for the processing capacity of the Raspberry Pi. The image size used for analysis was also adjusted so that the system could process each frame faster without losing important visual details.

The software was further streamlined to remove unnecessary operations and improve response time. The camera feed and detection process were coordinated carefully to provide consistent and uninterrupted live output. Proper cooling and stable power supply were maintained to ensure reliable performance during long periods of testing and operation.

These optimizations helped the system perform real time detection with reduced delay and improved stability. As a result, the project successfully demonstrated that efficient design and careful refinement can make advanced detection systems work effectively even on compact, low cost hardware platforms.

CHAPTER 10

CHALLENGES FACED

While developing the real time road anomaly detection system, we came across several challenges that tested both the technical and practical aspects of the project. One of the major difficulties we faced was running the YOLOv8 model smoothly on the Raspberry Pi 5. Since the Pi has limited processing power compared to a standard computer, achieving real-time detection without reducing accuracy required a lot of fine-tuning. We had to resize input frames, adjust detection thresholds, and optimize the model to make it run efficiently on the embedded hardware.

Another major challenge was maintaining accuracy under different lighting and road conditions. The same road looked completely different in bright sunlight, shadows, or during wet weather, which sometimes caused false detections. One particular issue was the visual similarity between potholes and manholes both appear as dark circular regions on the road, and the model occasionally confused one for the other. To reduce these misclassifications, we had to improve the dataset with more varied images and adjust the training parameters carefully.

We also faced some hardware related challenges, especially while setting up the Raspberry Pi environment. Installing Python libraries, configuring dependencies, and setting up the camera took a lot of trial and error. Camera focusing and placement were critical too, since even a slight tilt or glare from headlights or sunlight could affect the clarity of captured frames. Another issue we encountered was power stability, particularly when using portable power banks, as the Pi would occasionally restart during long runs.

Preparing a custom dataset was another time consuming part of the project. Capturing and labeling images of potholes, manholes, and speed breakers under various conditions required patience and accuracy. Storage limitations on the Pi and the delay during real-time display also posed minor difficulties during testing.

Despite all these challenges, through repeated testing, debugging, and adjustments, we managed to make the system work effectively. Overcoming these obstacles helped us understand not only the technical aspects of embedded AI and computer vision but also the importance of teamwork, patience, and persistence in completing a real world engineering project.

CHAPTER 11

APPLICATIONS

The developed real time road anomaly detection system has a wide range of applications across transportation, infrastructure, and smart city domains. It can be effectively used for smart city infrastructure monitoring, where continuous detection of potholes, manholes, and speed breakers helps authorities maintain better road conditions. When mounted on vehicles, such as public buses or municipal inspection vehicles, the system can perform automated road inspections during regular operations, minimizing the need for manual surveys. The technology also contributes to road safety management by providing early detection of road damages that could lead to accidents or vehicle breakdowns. Government agencies and municipal departments can utilize this system for efficient maintenance planning, identifying and prioritizing road sections based on the severity of detected anomalies. With integration into navigation platforms, the system can aid in route optimization by alerting drivers about poor road conditions ahead. It can further serve as a valuable tool in autonomous vehicle systems, assisting in terrain adaptation and safe navigation. The project also opens opportunities for crowdsourced road condition mapping, where data from multiple units can be combined to create real time maps of road quality. In the logistics and transport industry, it can assist in fleet management by monitoring road conditions to reduce vehicle wear and operational costs. Additionally, the system provides a practical platform for research and education in embedded AI and computer vision applications. Beyond everyday use, it can also support disaster and emergency response efforts, helping identify damaged routes after floods or heavy rains to guide repair and rescue operations efficiently.

CHAPTER 12

FUTURE SCOPE

The system can be further enhanced by integrating additional sensors and connectivity features to improve its accuracy and usability. Incorporating a LiDAR sensor will allow the measurement of the depth and shape of road anomalies, enabling more precise analysis of pothole dimensions and severity. The addition of a GPS module will make it possible to geo-tag detected anomalies, storing their exact latitude and longitude for location-based monitoring.

Integration with OpenStreetMap can provide an interactive visualization of detected potholes, manholes, and speed breakers on a digital map, making it easier for authorities and users to identify damaged road sections quickly. Furthermore, cloud connectivity using platforms such as ThingSpeak or Firebase can be employed for real-time data upload and centralized monitoring through a web dashboard.

Deploying the system on moving vehicles would enable continuous road inspection and data collection, leading to the creation of a large-scale smart road monitoring network. These advancements would transform the current prototype into a comprehensive AI-driven road management system, supporting safer transportation and efficient infrastructure maintenance within smart city environments.