ARM BHARAT
AI-SoC Student Challenge

Real-Time Road Anomaly Detection from Dashcam Footage on Raspberry Pi

—

Team : Sharan Tej, Subham Chhotaray, Gourab Kumar Sial

1.Abstract :

Road infrastructure monitoring is a critical challenge for improving traffic safety and reducing vehicle damage. Traditional cloud-based vision systems suffer from high latency, connectivity dependency, and privacy concerns. This project presents a real-time edge AI system for detecting road anomalies such as potholes and unexpected obstacles using dashcam footage processed entirely on a Raspberry Pi. A lightweight object detection model is optimized using quantization techniques and deployed using an edge inference framework to run efficiently on CPU-only hardware. The system captures live video, performs real-time inference, and logs detected anomalies with timestamps. The project demonstrates a complete embedded vision pipeline and highlights the trade-offs between accuracy, speed, and compute constraints on Arm-based platforms.

2.Introduction :

Road anomaly detection traditionally relies on manual inspection or centralized cloud-based vision systems. While cloud-based processing enables high computational capability, it introduces significant latency, continuous connectivity requirements, bandwidth consumption, and privacy concerns. For real-time safety applications such as pothole detection from moving vehicles, delayed inference reduces system effectiveness. Therefore, performing inference directly on the edge device is a more practical and scalable solution.

Edge inference refers to executing neural network models directly on local hardware rather than transmitting data to remote servers. In this project, all video processing and anomaly detection are performed on a Raspberry Pi without reliance on cloud infrastructure. This approach reduces end-to-end latency, ensures operation in low-connectivity environments, and preserves data privacy by avoiding transmission of raw video streams. Additionally, edge processing significantly reduces operational costs associated with cloud compute and storage.

A critical constraint of this project is CPU-only execution without external accelerators such as GPUs, TPUs, or dedicated AI hats. Running the detection pipeline entirely on the Arm Cortex-based CPU demonstrates efficient utilization of limited computational resources. This constraint enforces careful model selection, quantization, and runtime optimization. From an engineering perspective, CPU-only deployment ensures lower power consumption, reduced hardware cost, and easier scalability for mass deployment scenarios such as fleet vehicles or municipal monitoring systems.

Real-world deployment demands reliability, efficiency, and affordability. A solution that requires expensive accelerators limits large-scale adoption. By achieving near real-time inference using only the Raspberry Pi CPU, this project demonstrates a practical and deployable architecture suitable for embedded automotive environments. The system highlights the trade-offs between model accuracy, computational load, memory footprint, and inference speed on Arm-based platforms, which are widely used in edge and IoT systems globally.

Overall, this project emphasizes efficient edge AI deployment, demonstrating how optimized neural networks can operate within strict hardware constraints while maintaining functional real-time performance for safety-critical road monitoring applications.

3. Problem Statement and Motivation

3.1 Problem Statement

The objective of this project is to design and implement a real-time edge AI application capable of detecting road anomalies such as potholes, cracks and unexpected obstacles from dashcam footage using a Raspberry Pi. The system must process video streams in near real-time, operate under constrained computational resources, and log detected anomalies reliably.

3.2 Motivation

Road anomalies contribute significantly to vehicle damage, traffic congestion, and accidents. Manual inspection of road conditions is slow and expensive, while cloud-based AI solutions introduce latency and require continuous internet connectivity. Edge AI enables on-device processing, reducing latency, improving privacy, and lowering deployment costs. Raspberry Pi serves as an ideal platform for prototyping such solutions due to its affordability, availability, and real-world deployment feasibility.

4. System Requirements:

4.1 Hardware Requirements:

Component	Description
Processing Unit	Raspberry Pi 5 (8 GB RAM)

Camera	Lenovo Essential FHD Webcam
Storage	16GB microSD card
Compute	CPU-only execution (no external accelerators)

4.2 Software Requirements:

Software	**Purpose**
Raspberry Pi OS	Operating system
Python 3.11.9	Application development
OpenCV2 4.13.0	Video capture and image processing
ONNX Runtime	Edge model inference, Quantised model
YOLO V8	Real-time anomaly detection

5. System Architecture :

Raspberry Pi 5 (8 GB RAM) was used, along with Lenovo FHD USB Cam.

OS used: Raspberry Pi OS (64 bit)

16 GB microSD was used.

A virtual environment was created inside the Raspberry Pi, and the required packages were installed. Models and codes required for setting up in Pi, were moved from the System to Pi using “scp”.

The annotated video and detections.log were saved in Pi through the code. Later, moved to the system using scp.

6.Methodology:

We used YoloV8 for Image processing, we fine tuned it with data set from Roboflow:

https://universe.roboflow.com/abdo-pfqkm/projet_route_dataset-dgri8/dataset/6/download/yolov8

The .ipynb notebook used for training the same on Google Colab has been attached in the Github repo. Three models, best_int8.onnx, best_fp16.onnx, and best_fp32.onnx were generated. The performance was tested on testing video, maintaining a decent 12-16 FPS on fp32, as tested through Raspberry Pi. Thus, we used best_fp32.onnx in our final testing video.

7.Hardware Utilization:

Efficient utilization of embedded hardware resources is essential for achieving stable real-time inference on edge devices. This project was deployed and tested on a Raspberry Pi 5 with 8GB RAM and a 16GB high-speed microSD card. The system operates entirely on the onboard CPU without any external accelerators or AI-specific hardware.

7.1 System specifications:

The deployment configuration is summarized below:

Raspberry Pi 5, 8GB RAM
16GB high-speed microSD storage
Raspberry Pi OS 64-bit
USB Webcam for video input
ONNX Runtime (CPU execution provider) for inference

The Raspberry Pi 5 provides improved CPU performance compared to earlier models, making it suitable for real-time video analytics workloads when combined with optimized models.

7.2 Real Time Inference performance (FPS):

The deployed YOLOv8 model was fine-tuned on a Roboflow dataset specifically curated for road anomaly detection. The trained model was exported to ONNX format and integrated into the OpenCV-based inference pipeline.

During continuous real-time testing:

Average frame rate achieved was approximately 12 to 13 frames per second.
The lowest observed frame rate was 12 FPS.
The highest observed frame rate reached 16 FPS.

The frame rate remained stable during sustained execution, indicating that the selected model architecture and optimization level were appropriate for CPU-only deployment on Raspberry Pi 5.

The achieved performance exceeds the minimum project requirement of five frames per second, demonstrating that real-time anomaly detection is feasible on embedded Arm-based platforms without dedicated AI accelerators.

REPOSITORY LINK :

https://github.com/SubhamC07/RoadAnomalyDetectionARM

YOUTUBE VIDEO LINK :

https://youtu.be/IShCUN7VvHY?si=QYojKj8qebK-2zkr

OUTPUT VIDEO LINK :

https://youtu.be/YCjfnL0lWjc