Real-Time Road Anomaly Detection Using Raspberry Pi 4

Name: Banaj gupta, khushi aftab

Platform: Raspberry Pi 4 (4GB RAM)

1. Introduction

Road conditions play an important role in vehicle safety. Problems such as unmarked speed bumps and damaged road surfaces can cause accidents, especially when drivers are not warned in advance.

This project aims to develop a real-time road anomaly detection system that can run on a low-cost edge device. Instead of using cloud processing, the entire system runs locally on a Raspberry Pi 4.

2. Objective

The main objectives of this project are:

- To train an object detection model for road anomaly detection
- To optimize the model for edge deployment
- To achieve real-time inference on Raspberry Pi 4 using CPU only

3. System Overview

The system is divided into three stages:

1. **Model Training** on a computer using a labeled dataset
2. **Model Optimizatio**n using ONNX conversion and INT8 quantization
3. **Deployment** on Raspberry Pi 4 for real-time inference

4. Dataset and Model

The dataset contains images of different road conditions categorized into three classes: unpaved roads, unmarked speed bumps, and road irregularities. The images were annotated in YOLO format.

YOLOv8 Nano was selected because it is lightweight and suitable for devices with limited computational resources.

5. Training Process

The model was trained using the Ultralytics YOLO framework. Early stopping was used to avoid overfitting. The trained model was tested visually to ensure that bounding boxes were correctly predicted.

6. Model Optimization

To improve performance on Raspberry Pi:

- The trained model was exported to ONNX format
- INT8 quantization was applied to reduce computation cost

- ONNX Runtime was used for inference

These optimizations significantly improved inference speed while maintaining acceptable accuracy.

7. Hardware Setup

- Raspberry Pi 4 Model B
- 4GB RAM
- CPU-only inference
- USB camera for image input
- HEATSINK
- SD card

8. Results

The optimized model achieved an average speed of **4-5** FPS on Raspberry Pi 4, which meets the requirements for real-time road monitoring.

9. Challenges

Some challenges faced during the project include:

- Limited processing power of Raspberry Pi
- ONNX Runtime compatibility issues on ARM
- Performance tuning without GPU acceleration
- INT8 Quantization Compatibility Issues

While attempting to optimize the model using INT8 quantization, it was found that the quantized model did not run correctly on Raspberry Pi in certain cases. Due to ONNX Runtime limitations on ARM architecture, inference failed for the quantized version, and finally only the standard **best.onnx** model worked reliably. This required multiple trials and debugging to identify a stable deployment approach.

- Dataset Collection Difficulties

Dataset collection was also a major challenge, as road anomaly data is not easily available in public datasets. Images had to be collected from multiple sources and manually reviewed. Maintaining proper annotations, class balance, and handling variations in lighting and road conditions required considerable effort.

These challenges were resolved through optimization and careful model selection.

10. Conclusion

This project shows that real-time computer vision applications can be successfully implemented on low-cost edge devices using proper optimization techniques. The system can be further improved by using better hardware or additional post-processing.

11.Future Scope

- Integration with vehicle warning systems
- Deployment on more powerful edge devices
- Further reduction in model size

12.References

[1] Ultralytics YOLOv8 Documentation

https://docs.ultralytics.com

[2] ONNX Runtime Documentation

https://onnxruntime.ai/docs/

[3] Raspberry Pi 4 Model B – Official Documentation

https://www.raspberrypi.com/products/raspberry-pi-4-model-b/

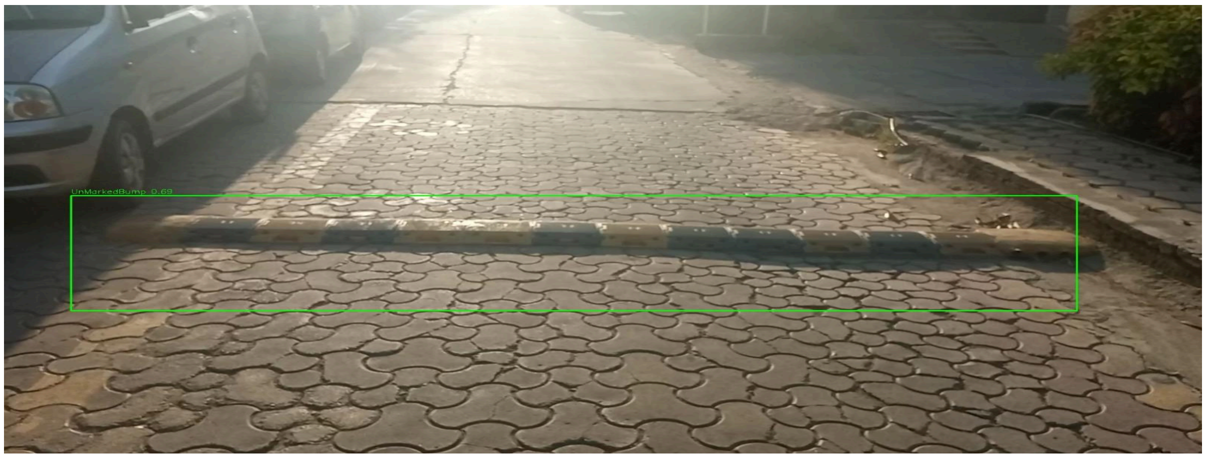
[4] Redmon, J., et al. “You Only Look Once: Unified, Real-Time Object Detection.”

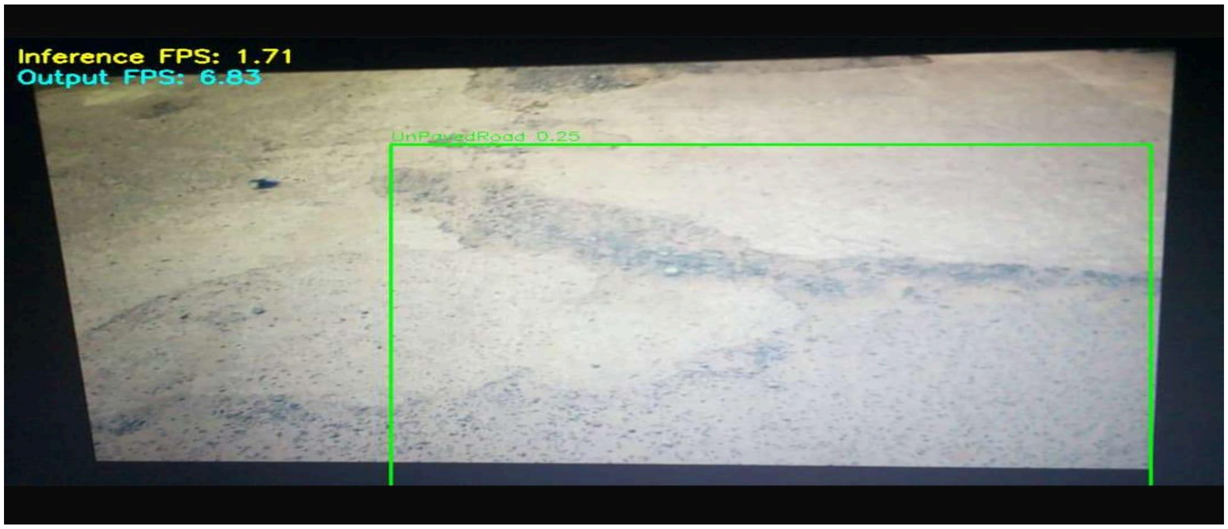
https://arxiv.org/abs/1506.02640

[5] ARM Developer – Edge AI Optimization

https://developer.arm.com/solutions/machine-learning-on-arm

13.Sample Detection Results




Inference FPS: 1.71
Output FPS: 6.83
UnPavedRoad 0.25