Automated Road Damage Detection using YOLOv8

1 Introduction

Safe and well-maintained road infrastructure is essential for efficient transportation and public safety. Conventional road inspection methods rely heavily on manual surveys, which are time-consuming, expensive, and prone to human subjectivity. As road networks continue to expand, there is a growing need for automated and scalable solutions capable of detecting road damage accurately and efficiently.

This project presents an automated road damage detection system using deep learning-based object detection. The solution is developed using the Road Damage Detection 2022 (RDD2022) dataset and employs YOLOv8, a state-of-the-art real-time object detection framework. The objective is to detect, localize, and classify multiple types of road damage from images while producing predictions compatible with standard mean Average Precision (mAP) evaluation metrics.

2 Dataset Description

The Road Damage Detection 2022 (RDD2022) dataset is a large-scale, multi-national collection of road images captured under diverse environmental conditions such as varying illumination, weather, and road quality.

2.1 Dataset Organization

The dataset is divided into three subsets:

- **Training set**: Images with annotated ground truth labels used for model training.
- **Validation set**: Labeled images used for performance evaluation and model selection.
- **Test set**: Unlabeled images used for final evaluation.

Each image may contain zero or more instances of road damage.

2.2 Damage Categories

The model is trained to detect five classes of road damage:

1. Longitudinal Crack
2. Transverse Crack

3. Alligator Crack
4. Other Corruption
5. Pothole

Annotations are provided in YOLO format, where each object is represented by a class identifier and normalized bounding box coordinates.

3 Model Architecture

3.1 YOLOv8 Overview

YOLOv8 is a single-stage object detection architecture optimized for both accuracy and inference speed. The model consists of three primary components:

- **Backbone**: Extracts hierarchical feature representations from the input image.
- **Neck**: Aggregates multi-scale features to enhance detection across varying object sizes.
- **Detection Head**: Predicts bounding boxes, object confidence scores, and class probabilities.

3.2 Model Selection

The YOLOv8s (small) variant is selected for this project to balance detection performance and computational efficiency. This choice is particularly suitable for detecting thin and small objects such as cracks while remaining feasible on limited GPU hardware. Pretrained weights trained on the COCO dataset are used to initialize the model through transfer learning.

4 Data Augmentation and Preprocessing

YOLOv8 applies built-in data augmentation techniques during training to improve model robustness and generalization. These include random horizontal flipping, color space augmentation, scaling, translation, and mosaic augmentation. Images are resized to a fixed resolution prior to training, and bounding box coordinates are automatically adjusted to reflect these transformations.

5 Training Strategy and Hyperparameters

5.1 Training Setup

The model is trained using the Ultralytics YOLOv8 framework with the default optimizer configuration. Training is performed on an NVIDIA GPU with mixed precision enabled to improve computational efficiency.

Parameter	**Value**
Image size	768 × 768
Batch size	16
Epochs	30
Optimizer	AdamW (default)
Mixed Precision (AMP)	Enabled
Device	NVIDIA GPU

Table 1: Key training hyperparameters

5.2 Hyperparameters

6 Model Evaluation

6.1 Evaluation Metrics

Model performance is evaluated using Mean Average Precision (mAP), which is the standard metric for object detection tasks:

- **mAP@0.5**: Measures detection accuracy at an Intersection over Union (IoU) threshold of 0.5.
- **mAP@0.5:0.95**: Computes the average mAP across multiple IoU thresholds, providing a stricter evaluation.

6.2 Validation Results

After 30 epochs of training, the model achieved the following validation performance:

- **mAP@0.5**: approximately 0.60
- **mAP@0.5:0.95**: approximately 0.32

The model demonstrates strong performance on crack-related classes and the "Other Corruption" category. Pothole detection remains comparatively more challenging due to smaller object size and fewer training instances.

7 Performance Improvements over Baseline

Several design choices contributed to improved performance compared to a baseline configuration:

- **Transfer Learning**: Initialization with pretrained weights accelerated convergence and improved generalization.
- **Higher Image Resolution**: Using a 768×768 input resolution enhanced detection of fine-grained damage patterns.
- **Adequate Training Duration**: Training for 30 epochs allowed the model to converge without overfitting.

- **Mixed Precision Training**: Enabled efficient GPU utilization while maintaining numerical stability.

8 Test Inference and Submission Format

For the test dataset, the trained model generates predictions in YOLO-compatible format:

```
<class_id> <x_center> <y_center> <width> <height> <confidence>
```

A prediction file is generated for every test image, including empty files for images with no detected objects, ensuring full compliance with the evaluation requirements. All prediction files are packaged into a single submission archive.

9 Conclusion

This work presents a robust and scalable solution for automated road damage detection using YOLOv8. By leveraging transfer learning, optimized image resolution, and efficient training strategies, the system achieves competitive performance on a challenging real-world dataset. The approach is reproducible, computationally efficient, and suitable for deployment in intelligent transportation and infrastructure monitoring applications.