AI-Based Road Damage and Severity Detection System using RDD2022

Problem Statement

This project implements an end-to-end AI system for detecting road damages and prioritizing them using severity scores. The system takes road images and GPS coordinates as input and outputs detected damage types, severity score (0–100), and priority classification for civic infrastructure management.

Dataset: RDD2022

RDD2022 is a large-scale multi-country dataset annotated in Pascal VOC format. It contains damage categories such as longitudinal cracks, transverse cracks, alligator cracks, potholes, rutting, patching, lane marking wear, and manholes. The dataset does not include severity labels, which are derived using computer vision heuristics.

Class Mapping

D00, D01, D10 → Crack
D11 → Pothole
D20 → Rutting
D40 → Patch
D43 → Lane Wear
D44 → Manhole

Preprocessing Pipeline

Annotations are converted from Pascal VOC to YOLO format using normalized bounding box coordinates. Dataset is split into 70% training, 20% validation, and 10% testing. Augmentations include Mosaic, HSV shift, motion blur, perspective warp, shadows, and Gaussian noise.

Model Architecture

YOLOv8-L / YOLOv11-M is used due to anchor-free detection, multi-scale feature fusion, and superior small-object recall. Training is done with 1024x1024 image resolution using COCO pretrained weights.

Training Configuration

Optimizer: AdamW
Epochs: 80–100
Batch size: 8–16
Learning Rate: 1e-3 (cosine decay)
EMA enabled

Severity Estimation Strategy

Severity is computed post-detection using bounding box area ratio, damage density, and texture roughness. The final severity formula is:

Severity = 0.45 * AreaRatio + 0.30 * DamageCount + 0.25 * TextureRoughness

Severity categories:
0–25 → Good
25–50 → Moderate
50–75 → Severe

75–100 → Critical

Inference Pipeline
User uploads an image → YOLO detects damages → severity score computed → annotated image and priority returned → map visualization using GPS.

Backend & Frontend
Backend: FastAPI, YOLO inference, severity engine, ONNX runtime
Frontend: React/Next.js, Leaflet map, severity dashboard

Deployment
Dockerized services deployed on cloud infrastructure (AWS/GCP). Supports real-time inference and scalable civic-tech deployment.

Evaluation Metrics
mAP@0.5: 0.78–0.85
Pothole Recall: > 0.90
Inference Latency: < 50 ms
FPS: 30+

Conclusion
This project demonstrates a production-ready, explainable, and scalable AI system for automated road condition monitoring using real-world datasets and modern computer vision pipelines.