Milestone 1 Submission: AI-Powered Road Damage Detection System

1. Problem Statement

The goal of this project is to detect potholes and major cracks, both longitudinal and transverse, in real-time, using images captured by smartphones or dashcams mounted inside moving vehicles on Indian roads.

In Scope

Aspect	Specification
Damage Types	Potholes, Longitudinal + transverse cracks
Dataset	RDD2022 Indian subset (~7,000 images)
Input	Single RGB images (640×480, smartphone quality)
Deployment	Offline evaluation + simulated edge (laptop CPU, <100ms/frame)
Metrics	mAP@0.5, Precision/Recall/F1, FPS, false positives per minute

Explicitly Out of Scope

- Minor cracks, alligator cracks, and construction zone damage
- Live vehicle deployment or any hardware integration
- Multi-modal sensors such as LiDAR, thermal, or stereo cameras
- Training data from non-Indian roads
- Full video processing pipelines

Future Dataset Expansion

The project is currently focused on the RDD2022 Indian subset. However, the system's design allows for the potential expansion of the dataset to include road data from other countries in future iterations, if time and project scope permit.

2. Why This Problem Matters

Pothole-related accidents in India rose by 53% between 2020 and 2024, now accounting for over 3,713 cases every year. India's road network spans 5.5 million kilometres, and keeping it safe through manual inspection alone costs the government around ₹500 crore annually. This is a figure that's simply not sustainable.

Our system is designed for smartphones and dashcams used in personal vehicles or civic fleet cars, making crowd-sourced road monitoring a realistic option. This is especially relevant in monsoon-heavy India, where poor road conditions are linked to roughly 70% of road accidents.

3. Key Stakeholders

The AI-Powered Road Damage Detection System involves two primary groups of stakeholders:

1. **Municipal Corporations and Road Maintenance Authorities:** These entities are responsible for road upkeep. The system provides them with timely, accurate, and location-specific data on road damage, enabling proactive maintenance and efficient resource allocation.
2. **Common Citizens and Road Users:** This group benefits from safer roads and reduced accident rates. They also act as potential data contributors (crowd-sourced monitoring via personal devices), allowing them to quickly report road conditions to maintenance authorities.

4. Current State-of-the-Art Solutions

We reviewed over 25 papers published between 2018 and 2026. YOLO-based architectures dominate real-time detection tasks, and the RDD2022 dataset has become the standard benchmark for road damage research.

Key Approaches & Benchmarks

Category	Representative Works	Datasets	Performance	Limitations
YOLO Real-Time	YOLOv5/v8, SEA-YOLOv8	RDD2022	mAP@0.5: 0.55-0.63 (India); F1: 0.70-0.77; FPS: 20-50 GPU	Domain shift to India (−15% mAP), slow on CPU
Advanced Detection	YOLO-RD, RT-DETR	RDD2022 India/Japan	mAP: 0.65-0.75; Precision: >0.75	High compute, poor on small potholes
Segmentation	U-Net, DeepLab	Crack500, custom	IoU: 0.70-0.75	<10 FPS, misses irregular shapes
Lightweight	MobileNet+YOLO	Custom India	FPS: 15-25 CPU; Recall: 0.72	Lower accuracy (mAP: 0.58)

The official RDD2022 baseline uses YOLOv5s, achieving mAP@0.5 = 0.62 overall, but this drops to 0.55 on the Indian subset, largely due to differences in lighting conditions and road textures.

Standard Metrics & Success Criteria

- Primary: mAP@0.5 (target > 0.70)
- Supporting: Precision = TP/(TP+FP), Recall = TP/(TP+FN), F1 = 2PR/(P+R)
- Practical: FPS > 10 on CPU, fewer than 5 false positives per minute, minimum detection distance

5. Shortcomings & Research Gaps

Critical Limitations

- Domain Bias: Models trained on Japanese or US roads perform poorly on Indian roads, with a 15-25% drop in mAP. Monsoon-related wetness, shadow patterns, and asphalt textures are largely unrepresented in existing training data.
- Edge Deployment: Most published solutions rely on GPU hardware
- Small Object Detection: Potholes smaller than 100px are frequently missed, particularly when there's motion blur or partial occlusion.
- India-Specific Gap: No existing work specifically fine-tunes on the RDD2022 Indian subset, which contains over 7,000 images.

The Opportunity

By combining India-specific fine-tuning with targeted augmentation (simulating blur, shadows, and rain) and model distillation, we expect a 10-20% mAP improvement over baseline while maintaining 15+ FPS on a standard laptop CPU. This directly addresses the lack of affordable, practical tools for civic road monitoring.

6. Our Contribution Dimensions

Contribution Area	**How We Address It**	**Expected Impact**
Performance	India-specific fine-tuning	+15% mAP vs. baseline
Efficiency	Model distillation	15+ FPS on laptop CPU
New Context	Smartphone deployment	Practical civic use case
Robustness	Cross-region generalisation	Works on unseen Indian roads

To our knowledge, this is the first undergraduate project to combine India-specific fine-tuning, edge optimisation, and cross-region testing on the RDD2022 dataset.

7. Detailed Implementation Strategy

This section outlines the specific technical methodologies employed to address the critical research gaps and achieve the project's performance and efficiency targets.

Data Engineering & Domain Adaptation

Since standard models exhibit a 15-25% drop in accuracy on Indian roads, our strategy is designed to bridge this *domain gap*.

- **India-Specific Fine-Tuning:** The RDD2022 Indian subset (~7,000 images) is utilised as the primary training corpus rather than generic datasets, focusing the model's learning on Indian road characteristics.
- **Weather & Lighting Augmentation:** Synthetic data generation is implemented to simulate challenging conditions, such as monsoon wetness, high-contrast shadow patterns, and motion blur. This ensures the final model maintains high-performance under poor visibility.
- **Tiling for Small Object Detection:** To resolve the "Small Object Detection" gap, we use image tiling (e.g., Slicing Aided Hyper Inference - SAHI). By dividing the 640×480 input into smaller patches during training, the model is trained to better learn the features of potholes smaller than 100 pixels.

Architectural Optimization for Edge CPU

To achieve the crucial <100ms/frame inference goal on a standard laptop CPU (simulated edge deployment), we have optimised the detection architecture beyond standard YOLOv8.

- **Model Distillation:** A "Teacher-Student" framework is used, where a heavy, high-accuracy model (e.g., YOLOv8x) transfers complex feature knowledge to a lightweight student model (e.g., YOLOv8n or YOLOv10n). This maintains accuracy while significantly boosting speed.
- **Quantisation (INT8/OpenVINO):** The final trained PyTorch/Ultralytics model is converted to a highly efficient format (like OpenVINO or ONNX). Reducing precision from FP32 to INT8 will significantly boost inference speed on general-purpose CPUs with minimal loss in mAP.

- **Pruning:** We will remove redundant neural network connections that do not contribute meaningfully to pothole detection, further reducing the computational load for the simulated edge deployment.

Systematic Validation & MLOps

To ensure *cross-region generalisation* and long-term system reliability, we will implement systematic validation and MLOps practices.

- **Stratified K-Fold Validation:** We utilize an 80/20 train/test split, ensuring that images from different Indian cities within the RDD2022 set are proportionally represented in both segments to prevent overfitting to a single location.
- **Ablation Studies:** We will systematically disable one key technical improvement at a time (e.g., training without monsoon augmentation) to quantify and report exactly which technique contributes most to the expected 15% mAP boost.
- **Automated CI/CD Pipeline:** The MLOps framework will be used to automatically re-test the model against the defined Success Criteria (mAP, Precision, Recall, FPS) every time a new optimization or code change is pushed.

8. Evidence Plan - Proving Success

Quantitative Validation

- A mAP/F1 comparison table contrasting the baseline YOLOv8 against our fine-tuned model (80/20 train/test split)
- An ablation study examining the effect of augmentation and the trade-off between model size and inference speed

- **A Benchmarking Report:** This report will compare the inference speed (in **FPS** and **milliseconds/frame**) of the baseline model, the distilled/pruned model, and the final quantized model. All tests will be executed on the specified **standard laptop CPU** to definitively demonstrate the achievement of the **15+ FPS** target.

Qualitative Validation

- Detection videos with bounding box overlays
- Error analysis covering false positive and false negative case studies
- Confusion matrix visualisation

9. Team Contributions and Roles

This section outlines the specific roles and responsibilities of each team member, framed around the necessary project milestones to deliver the AI-Powered Road Damage Detection System.

Yash Kumar – Project Design & Documentation Lead

- **Milestone Focus:** Project conceptualisation, scope definition, and comprehensive documentation of all project phases.
- **Responsibilities:** Defining and structuring the problem statement, establishing project objectives and scope, preparing all project documentation, and ensuring proper reporting of methodologies and results.

Nishant Kumar – Technical Implementation: Data & Initial Experimentation

- **Milestone Focus:** Data readiness and foundational technical setup for model development.
- **Responsibilities:** Data engineering and preprocessing (especially for the RDD2022 Indian subset), supporting initial model experimentation, and assisting in technical system integration.

Rahul Yadav – Technical Implementation: Model Training & Optimization

- **Milestone Focus:** Core machine learning development, including training and optimisation of the final model architecture.
- **Responsibilities:** Machine learning model training and hyperparameter optimisation, implementing key algorithms, and conducting performance evaluation and testing (mAP, F1, precision/recall).

Neeraj - Technical Implementation: Deployment & Backend Integration

- **Milestone Focus:** Planning and execution of the simulated edge deployment and ensuring system functionality.
- **Responsibilities:** Focusing on final model deployment and integration, setting up the necessary backend structure for offline evaluation, and ensuring the system's operational functionality and scalability.

Kisalay – MLOps & System Reliability Lead

- **Milestone Focus:** Establishing a sustainable MLOps framework for model lifecycle management, monitoring, and version control.
- **Responsibilities:** Designing and managing the MLOps pipeline, implementing model versioning and performance tracking, handling CI/CD integration for automated testing, and ensuring overall system reliability and optimisation.

Collaborative Responsibilities

All technical team members jointly contribute to core technical implementation activities, including data engineering processes, model training and validation, and the final deployment and testing of the system.

10. References Papers

1. Ma, N., Zhang, J., & Ishikawa, T. (2022). RDD2022: A multi-national image dataset for automatic road damage detection. arXiv preprint arXiv:2205.11538.
2. Sekiba, H., & Ueda, K. (2022). Road Damage Detection Challenge 2022 (CRDDC'22): Methodology and results. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops.
3. Jocher, G., Chaurasia, A., & Qiu, J. (2023). YOLO by Ultralytics (Version 8.0). GitHub repository: https://github.com/ultralytics/ultralytics
4. Li, X., et al. (2021). Real-time road damage detection using YOLOv4. IEEE Transactions on Intelligent Transportation Systems.
5. Zhang, H., et al. (2025). SEA-YOLOv8: Road surface damage detection based on improved YOLOv8. Scientific Reports, 15(1), 14461.
6. Nepal, R., et al. (2025). YOLO-RD: Road damage detection with selective optimization modules. PMC Articles PMC11902777.
7. Zhang, L., Yang, F., Zhang, Y. D., & Zhu, J. (2016). Road crack detection using deep convolutional neural network. IEEE International Conference on Image Processing.
8. Fan, R., et al. (2022). Road crack detection using deep convolutional neural network and adaptive thresholding. Computers & Structures, 260, 106717.
9. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. European Conference on Computer Vision.
10. Mathavan, S. K., et al. (2024). Domain adaptation for road damage detection: India vs global datasets. Journal of Infrastructure Systems.

11. Shivhare, A., & Sharma, N. (2023). Lightweight road damage detection for Indian roads using MobileNet. International Conference on Computer Vision Applications.
12. Ministry of Road Transport and Highways (MoRTH). (2024). Annual Road Accidents Report 2020-2024. Government of India.
13. Howard, A. G., et al. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.
14. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. IEEE CVPR.

Benchmark & Competition Results

15. CRDDC'22 Technical Report. (2022). Road Damage Detection Challenge 2022 Results. https://rdd2022.sekilab.com/
16. Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., & Omata, H. (2018). Road damage detection and classification using deep neural networks with smartphone images. Computer-Aided Civil and Infrastructure Engineering, 33(12).
17. Zadeh, S. H., et al. (2024). RT-DETR for real-time road surface monitoring. IEEE Transactions on ITS.
18. Wang, J., et al. (2025). Multi-scale attention for pothole detection. Pattern Recognition Letters.
19. Kumar, R., & Patel, V. (2025). Indian road damage detection using transfer learning. Journal of Intelligent Transportation Systems.
20. Chen, Y., et al. (2026). Weather-robust road damage detection. ISPRS Journal of Photogrammetry.

Additional Baselines & Metrics

21. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. MICCAI.
22. Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. IEEE CVPR.
23. Lin, T. Y., et al. (2014). Microsoft COCO: Common objects in context. European Conference on Computer Vision.
24. Ishikawa, T., et al. (2023). Cross-country analysis of RDD2022 dataset. Transportation Research Part C.
25. Takahashi, K., & Sekiba, H. (2024). Benchmarking YOLO variants on RDD2022. Pattern Recognition.

26. Gupta, A., & Singh, R. (2025). Domain gap analysis: RDD2022 India performance. Indian Conference on Computer Vision.

11. Team Consent

Name	Roll No.	Sign
Yash Kumar	22F3000472	Y.K
Nishant Kumar	22f3003042	N.K
Rahul Yadav	22F1001680	R.Y
Neeraj Yadav	21F1005729	N.Y
Kisalay Pan	22F2001094	K.P