Dissertation Title:
Road Quality Assessment System Using Computer Vision and GPS-Based Mapping

Course No. : SEZG628T

Course Title: Dissertation

Dissertation Done by:

Student Name: ANISH A.

BITS ID: 2024TM93051

Degree Program: M.Tech in Software Engineering

Research Area: Machine Learning

Dissertation carried out at:

BizIntelligence Technologies Pvt. Ltd., Thiruvananthapuram, Kerala

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE,
PILANI
VIDYA VIHAR, PILANI, RAJASTHAN - 333031

May 2026

Abstract

Road quality assessment in India remains largely manual, infrequent, and disconnected from the navigation systems that drivers rely on daily. This dissertation presents a crowdsourced, dashcam-based road quality assessment system that classifies road segments from dashcam imagery and maps quality scores onto an OpenStreetMap road network to enable quality-aware route planning.

The system follows a six-stage pipeline: dashcam video ingestion and FFmpeg-based frame extraction, Tesseract OCR GPS parsing from dashcam overlays, automated filtering of stationary and night frames, consensus-based crowdsourced annotation, Swin Transformer classification, and GPS-snapped quality mapping with interactive visualisation and routing.

An initial object detection phase evaluated 17 YOLO variants (v8/v9/v10/v11), achieving a best mAP50 of 20.3%. The consistently poor performance and high false positive rates on Indian road scenes motivated a reformulation to direct segment classification. A full architecture sweep across 13 successful models identified the Swin Transformer as the best-performing architecture, benefiting from its hierarchical multi-scale representation. Hyperparameter tuning across 72 trials established that end-to-end fine-tuning without backbone freezing, unweighted loss, and a low learning rate of 3e-5 with cosine annealing produced the optimal configuration.

The annotated dataset comprises 3,216 consensus-validated images across 53 contributors, collected from 4 dashcam sources. Active learning targeting low-confidence predictions (max softmax < 0.7) reduced the class imbalance from an initial 9.2:1 Excellent-to-Poor ratio to 4.6:1 Good-to-Bad in the final dataset. A data-driven decision to consolidate five classes into three (Good, Bad, Invalid), justified by confusion matrix analysis and annotator consensus patterns, improved validation accuracy from 53% to 79.6% and macro F1 from 57% to 75%.

The mapping pipeline snapped GPS-linked predictions onto 808 OSM road segments using UTM-projected nearest-edge matching, with pessimistic and majority-vote quality aggregation stored in a GeoPackage for flexible downstream use. A quality-penalised routing algorithm comparing shortest and smoothest paths was deployed via a Flask navigation server with a Leaflet.js interface and REST API.

The primary limitation is the Bad class recall of 53%, reflecting the residual class imbalance and the visual ambiguity between mildly degraded and lightly damaged road surfaces at driving speed. The pessimistic aggregation strategy partially mitigates this at the segment level. The system is deployable in any region with OpenStreetMap coverage and is designed for extension through continued crowdsourced annotation and broader dashcam coverage.

Contents

Introduction

Broad Area of Work

Computer vision and deep learning have emerged as powerful tools for automated infrastructure monitoring and assessment. This research project intersects artificial intelligence, machine learning, computer vision, and geospatial information systems to address the critical problem of road quality assessment for navigation and municipal planning.

The broad scope of this work encompasses the following application areas:

- **Computer Vision**: Automated analysis of dashcam imagery to assess road surface quality, moving beyond manual inspection methods that are time-consuming, expensive, and infrequent.
- **Deep Learning**: Development and fine-tuning of neural network architectures for road condition classification, with emphasis on models that can generalise to diverse Indian road environments.
- **Transfer Learning**: Adaptation of pre-trained models (YOLO family for object detection, Vision Transformers and CNNs for classification) to Indian road conditions, which differ significantly from Western datasets in terms of surface materials, damage patterns, environmental conditions, and visual complexity.
- **Active Learning and Crowdsourcing**: Design of efficient annotation pipelines combining consensus-based crowdsourcing with active learning to create high-quality training datasets while minimising annotation effort.
- **Geospatial Mapping**: Integration of road quality assessments with GPS coordinates extracted from dashcam footage to enable quality-aware route planning and visualization of road conditions on interactive maps.

Background

Road infrastructure quality directly impacts commuter safety, vehicle maintenance costs, and municipal planning decisions. Navigation systems today optimise routes based on distance and travel time but fail to account for road surface quality, particularly pothole density and severity. This limitation forces commuters to either accept poor road conditions or rely on informal knowledge about road quality, leading to suboptimal route choices that compromise comfort, safety, and vehicle longevity.

In Indian road conditions, this problem is particularly acute due to diverse surface types (asphalt, concrete, gravel, dirt), varying maintenance standards across municipalities, monsoon impacts (waterlogging, erosion, rapid deterioration), and rapid urban development that often outpaces infrastructure maintenance. Potholes and road damage contribute to accidents, vehicle damage (estimated at thousands of rupees annually per vehicle), and increased travel discomfort. However, systematic data collection for road quality assessment remains manual, expensive, and infrequent, with municipal surveys often occurring only once or twice per year.

Recent advances in deep learning, particularly object detection architectures like YOLO (You Only Look Once) and image classification models such as Vision Transformers (ViT), have shown promise in automated defect detection and infrastructure monitoring. However, existing pothole detection models are predominantly trained on Western road datasets (Japan, United States, Europe) and perform poorly when applied to Indian roads. This performance degradation occurs due to fundamental differences in road surface materials, damage patterns, environmental conditions (high visual clutter from vendors, pedestrians, animals), and the sheer visual complexity of Indian road scenes.

The proliferation of affordable dashcams with integrated GPS capabilities provides an unprecedented opportunity to collect road condition data continuously and at scale. By combining computer vision techniques with GPS-based mapping, it becomes feasible to create crowdsourced road quality assessment systems that can inform both individual routing decisions and municipal maintenance prioritization.

This dissertation developed such a system specifically adapted for Indian road conditions. Through systematic experimentation with object detection (YOLO family, 17 variants) and classification approaches (Vision Transformers, CNNs, 15 architectures via a full architecture sweep), we

empirically established that segment-level road quality classification outperforms individual pothole localization for navigation-oriented applications. The resulting system covers the full pipeline: from dashcam video ingestion and OCR-based GPS extraction, through consensus-based crowdsourced annotation and Swin Transformer classification, to road segment quality mapping on OpenStreetMap networks and quality-aware route planning served via an interactive web interface. Our work addresses the gap between existing models trained on Western datasets and the unique characteristics of Indian road infrastructure, while introducing novel methodologies for efficient dataset creation through consensus-based crowdsourcing and active learning.

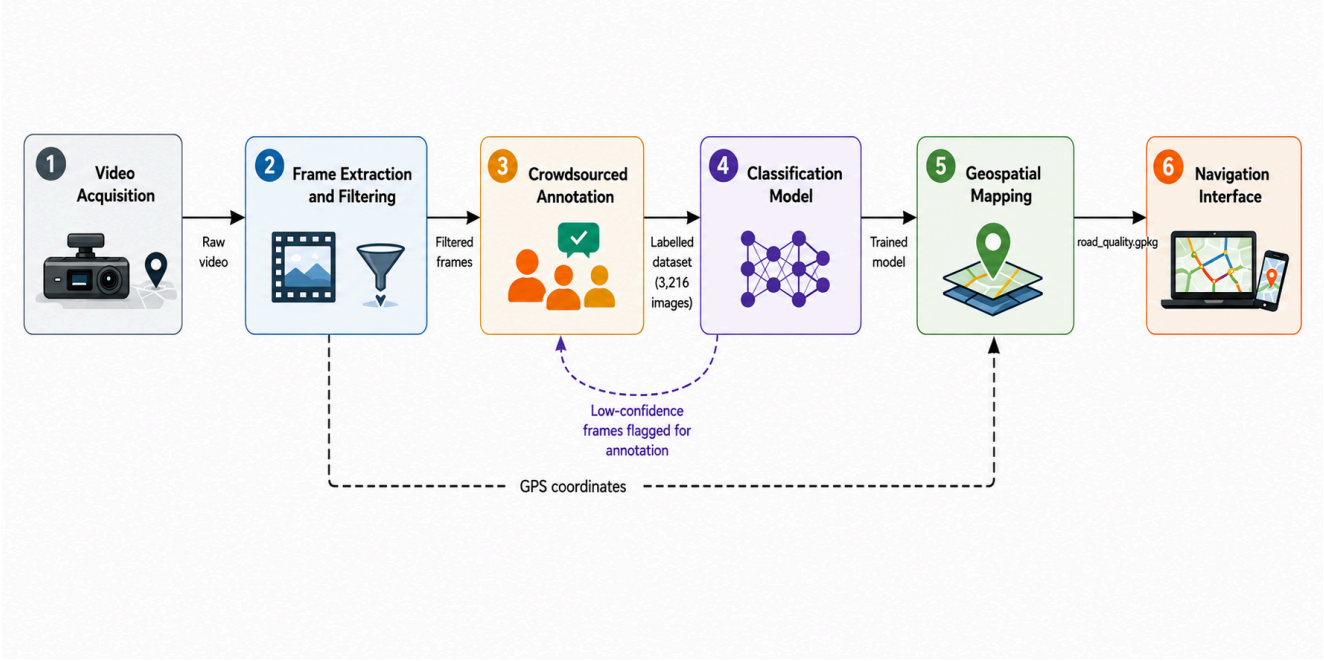


Figure 1: Overview of the road quality assessment pipeline. The full annotated diagram is presented as Figure 2 in the Methodology section.

Objectives

The objectives of this project evolved through systematic experimentation as findings from each phase informed the direction of subsequent work. The objectives as realised in the completed system are:

1. **Empirical Evaluation of Detection vs. Classification Approaches**: Systematically compare object detection (YOLO family, 17 variants across v8/v9/v10/YOLO11/YOLO26) and classification-based approaches (15 architectures via full sweep) for road quality assessment on Indian road conditions, using quantitative metrics to determine the more appropriate methodology for segment-level quality mapping.
2. **Development of High-Quality Annotated Dataset**: Create a multi-phase annotated dataset of road conditions from Kerala roads using dashcam footage, incorporating:
    a. Consensus-based crowdsourcing to ensure annotation quality, resulting in 3,216 images contributed by 53 annotators
    b. Active learning combined with multi-annotator consensus and a tiebreaker protocol to maximise labelling efficiency on minority classes while reducing noise
3. **Adaptation of Deep Learning Models for Indian Roads**: Fine-tune and compare architectures across the ResNet, EfficientNet, EfficientNetV2, ViT, ConvNeXt, and Swin Transformer families for direct road segment quality assessment. The Swin Transformer achieved the best validation accuracy of 79.6% on the final 3-class task, after class consolidation justified by confusion matrix analysis and annotation consensus patterns.
4. **Design of Efficient Annotation Pipeline**: Design an efficient annotation pipeline combining a gamified crowdsourcing platform with active learning, using iterative model-guided sample selection to reduce annotation burden on minority classes.
5. **GPS Integration and Data Preprocessing**: Implement OCR-based GPS coordinate extraction from dashcam overlays and build automated filtering pipelines for data quality assurance, covering stationary frame detection, night image removal, and GPS bounds validation.
6. **Geospatial Mapping and Quality-Aware Route Planning**: Build a complete geospatial pipeline covering the following:
    a. OSM road network download and UTM-projected GPS snapping (maximum 35m threshold) via osmnx
    b. Per-edge quality aggregation in two modes: pessimistic (worst label wins) and majority vote
    c. Interactive Folium map with toggleable layers for quality, aggregation mode, and observation density
    d. Quality-penalised routing comparing shortest path against a smoothest path computed using length × (5 − quality score) as edge weight
    e. A Flask navigation server serving live route computation and GeoJSON quality overlays via REST API

Note on Objective Evolution: The project originally targeted pothole detection using YOLOv8, with detection counts aggregated into segment-level scores. After testing 17 YOLO variants (best mAP50: 20.3%), object detection was abandoned in favour of direct segment classification. A second reformulation followed: confusion matrix analysis and annotator consensus patterns revealed consistently ambiguous boundaries between Excellent/Good and Fair/Poor, leading to consolidation from five classes to three (Good, Bad, Invalid). Validation accuracy improved from 53% (Swin-Small, 5-class) to 79.6% (Swin Transformer, 3-class). Both decisions were driven by data.

Scope of Work

The scope of this dissertation covers the design, implementation, and evaluation of an automated road quality assessment system adapted for Indian road conditions. The system spans the following components:

Data Collection and Preprocessing:

- Extract frames from dashcam video footage captured on Kerala roads at 1fps using FFmpeg
- Extract GPS coordinates and timestamps from dashcam overlays using Tesseract OCR with contrast enhancement and binarisation, validated against Kerala's geographic bounds (8°N-13°N, 74°E-78°E)
- Filter extracted frames automatically by removing stationary frames (GPS invariance and timestamp gaps), night frames (low mean pixel intensity), and frames with invalid or missing GPS data
- Organise clean frames into a structured dataset suitable for annotation and model training

Annotation Infrastructure:

- Design and implement a consensus-based crowdsourcing platform with gamification elements including annotator leaderboards and accuracy feedback
- Integrate active learning to identify and prioritise low-confidence samples (max softmax < 0.7) for annotation, targeting minority classes
- Establish quality control through a two-annotator agreement mechanism with a third-annotator tiebreaker

Model Development and Evaluation:

- Conduct systematic evaluation of 17 YOLO variants (v8/v9/v10/YOLO11/YOLO26) for pothole detection to empirically assess the viability of object detection for road quality assessment
- Evaluate 15 classification architectures across the ResNet, EfficientNet, EfficientNetV2, ViT, ConvNeXt, and Swin Transformer families using a group-aware train/val split on

video identity to prevent data leakage

- Evaluate two-phase training, label smoothing, and weighted CrossEntropyLoss via systematic hyperparameter tuning; apply the optimal configuration (end-to-end fine-tuning, unweighted loss, no label smoothing, CosineAnnealingLR at lr=3e-5) to the production model.
- Evaluate models using macro-averaged precision, recall, and F1-score to avoid minority class performance being masked by majority class dominance

Road Quality Classification:

- Classify road segments into five quality categories (Excellent, Good, Fair, Poor, Invalid) with a runtime flag supporting consolidation into a 3-class scheme (Good, Bad, Invalid) based on empirical confusion matrix analysis
- Generate segment-level quality assessments directly from dashcam imagery without intermediate pothole localisation

Geospatial Integration and Visualization:

- Download OSM drive networks for the GPS bounding box via osmnx, project to UTM, and snap each GPS point to the nearest road edge within a 35m threshold
- Aggregate per-edge quality scores using two strategies: pessimistic (worst label) and majority vote, with a minimum observation threshold per edge
- Render an interactive Folium map with toggleable layers for quality (both aggregation modes) and observation density, with hover tooltips showing street name, quality, observation count, and per-class breakdown
- Compute and compare shortest and quality-penalised smoothest routes using Dijkstra with length × (5 − quality score) as edge weight, served via a Flask navigation API

Scope Limitations:

- Geographic coverage is limited to Kerala roads; generalisation to other regions requires additional data collection and region-specific annotation
- The raw dataset comprises approximately 34,000 frames extracted per camera from dashcam footage, of which 3,216 have been annotated through consensus-based crowdsourcing. The remaining frames represent an unannotated pool available for future active learning cycles
- Data collection is limited to a single pre-monsoon season. Performance under post-monsoon conditions, where road damage patterns differ significantly due to waterlogging and accelerated surface deterioration, has not been evaluated
- The classification system addresses surface quality only and does not assess structural integrity or subsurface conditions
- Real-time inference and mobile deployment are outside the current scope; the system operates as an offline batch processing pipeline with a locally hosted navigation server
- Municipal maintenance prioritisation algorithms are not implemented; the system provides data collection, classification, and visualisation

Plan of Work

Table 1 presents the original plan of work alongside actual progress, with justifications for deviations based on empirical findings and methodological refinements.

Table 1: Project Plan of Work Showing Original and Revised Timelines with Justifications for Deviations

Phase	Original Timeline	Actual Timeline	Work Completed	Status & Notes
Dissertation Outline	31 Jan – 7 Feb 2026	31 Jan – 7 Feb 2026	Literature review, project proposal preparation, supervisor approval	**Completed**
Data Collection & Preparation	8 Feb – 21 Feb 2026	8 Feb – 28 Feb 2026	• Dashcam video collection from Kerala roads • FFmpeg-based frame extraction at 1fps (~34,000 frames) • Tesseract OCR-based GPS and timestamp extraction with contrast enhancement and binarisation • Automated filtering covering static vehicle detection, night image removal, and GPS bounds validation • Clean dataset organised for annotation	**Completed** *Extended by 1 week to include GPS extraction (originally planned for April)*
Object Detection Exploration	21 Feb – 10 Mar 2026	21 Feb – 10 Mar 2026	• Bounding box annotation of 2,000+ images using Label Studio • Systematic evaluation of 17 YOLO variants (v8/v9/v10/YOLO11/YOLO26) • Best result: YOLOv8n at mAP50 of 20.3% • Identification of high false positive and false negative rates as fundamental limitations	**Completed** *New phase: Not in original plan; empirical validation of detection approach*

Problem Reformulation	10 Mar – 15 Mar 2026	10 Mar – 15 Mar 2026	• Quantitative analysis of YOLO limitations for segment-level assessment • Redefinition of problem as 5-class road segment quality classification (Excellent, Good, Fair, Poor, Invalid)	**Completed** *Critical methodological decision based on experimental evidence*
Annotation System Development	15 Mar – 25 Mar 2026	12 Mar – 25 Mar 2026	• Self-annotation of initial image set to establish baseline class distributions • Design and implementation of gamified annotation platform with leaderboard and accuracy feedback • Two-annotator consensus with third-annotator tiebreaker protocol • 53 contributors onboarded; 3,216 consensus-validated annotations achieved	**Completed** *New phase: Consensus-based crowdsourcing not in original plan*
Model Development - Phase 1	21 Feb – 20 Mar 2026	20 Mar – 5 Apr 2026	• Initial ViT-Small training revealing severe overfitting (97% train / 59% val) • Full architecture sweep across 15 models: ResNet, EfficientNet, EfficientNetV2, ViT, ConvNeXt, Swin Transformer families • Group-aware train/val split on video identity to prevent data leakage • Two-phase training with frozen backbone warmup followed by full fine-tuning at lr/10 • CosineAnnealingLR scheduling, label smoothing, and weighted CrossEntropyLoss applied	**Completed** *Extended to address overfitting challenges and conduct systematic architecture comparison*

Active Learning Integration	N/A (not planned)	25 Mar – 15 Apr 2026	• Low-confidence sample identification pipeline implemented (max softmax < 0.7) • Targeted annotation queue for Poor and Fair minority classes • Iterative model-guided annotation loop operational throughout crowdsourcing phase	**Completed**
Crowdsourced Annotation	N/A (not planned)	25 Mar – 15 Apr 2026	• Gamified annotation platform deployed and maintained • 3,216 confirmed consensus annotations achieved across 53 contributors • Active learning targets integrated into annotation queue throughout	**Completed**
Mid-Semester Report	15 Mar – 28 Mar 2026	15 Mar – 28 Mar 2026	• Progress documentation and preliminary results analysis • Mid-semester report preparation and viva presentation	**Completed**
Model Optimization & Evaluation	29 Mar – 20 Apr 2026	29 Mar – 25 Apr 2026	• Full architecture sweep results consolidated via summary CSV and JSON metrics • Hyperparameter tuning for selected architectures • Swin Transformer selected as final model • Per-model logs, confusion matrices, and classification reports generated • Production inference and evaluation pipelines validated against ground truth	**Completed**

Class Structure Decision Point	N/A (not planned)	15 Apr – 20 Apr 2026	• Confusion matrix analysis confirming consistent ambiguity at Excellent/Good and Fair/Poor boundaries • Annotation consensus patterns corroborated class boundary issues • Consolidation to 3-class scheme (Good, Bad, Invalid) using runtime merge flag • Validation accuracy improved from 53% (Swin-Small, 5-class) to 79.6% (Swin-Small, 3-class)	**Completed - Decision Taken** *Data-driven decision taken based on active learning results and expanded dataset characteristics*
GPS Integration & Mapping	22 Apr – 5 May 2026	20 Apr – 30 Apr 2026	• OSM road network download via osmnx with 500m bounding box buffer; GraphML cached locally • UTM projection for metre-accurate GPS snapping (maximum 35m threshold) • Per-edge quality aggregation in pessimistic and majority-vote modes • Minimum observation threshold enforced per edge (default: 3) • Outputs to road_quality.gpkg and road_quality.csv	**Completed** *Advanced from the original timeline. Scope expanded.*
Visualization Development	29 Apr – 10 May 2026	28 Apr – 7 May 2026	• Interactive Folium map with CartoDB Positron base layer • Three toggleable layers: pessimistic quality, majority quality, and observation density • Line weight scaled with observation count (3px to 7px at 50+ observations) • Hover tooltips showing street name, quality, observation count, average score, and per-class breakdown	**Completed**

			• Dynamic legend displaying only labels present in data	
Route Planning & Navigation Server	N/A (not planned)	1 May – 7 May 2026	• Quality-penalised routing comparing shortest and smoothest paths using length × (5 − quality score) as edge weight • Poor roads assigned 4× penalty; unrated edges assigned score of 2.5 with 1.2 multiplier • Folium route comparison map with info panel showing distance, average quality score, rated coverage, and per-quality breakdown • Flask navigation server with Leaflet.js UI, GeoJSON quality API, and REST route endpoint	**Completed** *Scope expanded beyond original visualisation objective*
Comprehensive Evaluation	6 May – 12 May 2026	7 May – 10 May 2026	• Evaluation across diverse road segments and conditions • Baseline comparisons (random and majority-class predictors) • Classification report and confusion matrix analysis documented • Production inference validated via evaluate_production.py against ground truth	**Completed**
Final Report Writing	22 Apr – 12 May 2026	25 Apr – 10 May 2026	• Complete dissertation documentation • Methodology and results write-up • Conclusion and future work sections • Presentation preparation	**Completed**

Key Deviations and Justifications

1. Early GPS Extraction (Advanced by 2 months)

- **Original Plan**: GPS extraction scheduled for April 22 – May 5
- **Actual**: Completed during data preparation phase (February)
- **Justification**: OCR-based GPS extraction was straightforward to implement early and enabled immediate validation of data quality. Completing this early removed it from the critical path and allowed the mapping pipeline to proceed without delays in April.

2. Addition of Object Detection Exploration Phase

- **Not in Original Plan**: Systematic YOLO experimentation across 17 variants
- **Justification**: Rigorous empirical validation of the detection approach was necessary before committing to an alternative. Testing 17 YOLO variants (v8/v9/v10/YOLO11/YOLO26) and quantifying their failure (best mAP50: 20.3%) provides a stronger methodological basis for the reformulation to classification than a theoretical argument alone would.

3. Problem Reformulation (Mid-Project reformulation)

- **Impact**: Shift from detection-based aggregation to direct segment classification
- **Justification**: YOLO experiments demonstrated consistently poor performance (mAP50 15-20%) with high false positive and false negative rates. Direct classification better addresses the navigation use case, where segment-level safety matters more than individual pothole counts, and avoids the additional uncertainty introduced by aggregating detection outputs.

4. Addition of Consensus Annotation System

- **Not in Original Plan**: Gamified crowdsourcing with multi-annotator agreement
- **Justification**: Single-annotator labelling proved insufficient for nuanced road quality assessment, where class boundaries (particularly Good/Fair and Fair/Poor) are inherently subjective. The two-annotator consensus mechanism with a third-annotator tiebreaker reduces label noise and represents a reusable methodological contribution. Gamification (leaderboard, accuracy feedback) sustained annotator engagement across 53 contributors and 3,216 annotations.

5. Active Learning Integration

- **Not in Original Plan**: Iterative model-guided annotation
- **Justification**: The initial dataset exhibited severe class imbalance (9.2:1 Excellent to Poor ratio). Random annotation of the unannotated pool would have perpetuated this imbalance. Targeting low-confidence predictions (max softmax < 0.7) directed annotation effort toward minority classes and ambiguous cases, improving model performance where it mattered most.

6. Class Structure Flexibility

- **Not in Original Plan**: Reduction from 5-class to 3-class scheme
- **Justification**: Confusion matrix analysis across all 15 evaluated architectures consistently showed high confusion at the Excellent/Good and Fair/Poor boundaries. Annotator consensus patterns corroborated that these boundaries were genuinely ambiguous rather than a modelling failure. Consolidating to three classes (Good, Bad, Invalid) using a runtime merge flag improved validation accuracy from 53% (Swin-Small, 5-class production) to 79.6% (Swin-Small, 3-class), confirming that the class boundaries were the bottleneck rather than the architecture.

7. Extended Model Development Phase

- **Original**: 1 month (Feb 21 – Mar 20)
- **Revised**: Approximately 5 weeks (Mar 20 – Apr 25)
- **Justification**: Discovery of severe overfitting in the initial ViT-Small experiment (97% train / 59% val) necessitated a full architecture sweep across 15 models, systematic hyperparameter tuning, and iterative evaluation against the expanding annotated dataset. This was a necessary investment to arrive at a defensible final model selection.

8. Addition of Route Planning and Navigation Server

- **Not in Original Plan:**Quality-penalised routing and Flask navigation server

- **Justification:** The mapping pipeline naturally enabled quality-aware routing once per-edge quality scores were available on the OSM graph. Implementing shortest versus smoothest path comparison and exposing it via a REST API substantially strengthened the practical contribution of the system, demonstrating end-to-end utility from dashcam footage to actionable navigation output.

Risk Mitigation

Table 2 summarises identified risks, mitigation strategies, and resolution status.

Table 2: Identified Project Risks, Mitigation Strategies, and Resolution Status

Risk	Mitigation Strategy	Status
Small dataset leading to overfitting	Full architecture sweep to find optimal capacity-data fit; strong augmentation; label smoothing and early stopping	**Resolved**. Swin Transformer achieved 79.6% validation accuracy
Class imbalance	Weighted CrossEntropyLoss; active learning targeting minority classes; runtime class merging flag	**Resolved**. 3-class consolidation addressed imbalance structurally
Insufficient data for 5-class problem	Continuous crowdsourced annotation; active learning; class merging as fallback	**Resolved.** 3,216 annotations achieved; class consolidation reduced per-class data requirement
Ambiguous class boundaries	Annotation consensus rate monitoring; confusion matrix analysis across architectures	**Resolved**. Data-driven consolidation to 3-class scheme
Active learning efficiency	Confidence threshold (max softmax < 0.7) for sample selection; queue prioritised by class rarity	**Resolved**
Validation methodology concerns	Group-aware train/val split on video identity; stratified split; structured JSON metrics	**Resolved**
GPS extraction accuracy	Tesseract OCR with contrast enhancement; coordinate bounds validation against Kerala limits	**Resolved**
GPS snapping errors	UTM projection; 35m maximum snapping threshold; minimum 3 observations per edge	**Resolved**
Route quality reliability	Dual aggregation modes (pessimistic and majority vote); conservative default score for unrated edges	**Resolved**

Literature References

Road Damage Detection and Infrastructure Monitoring

Maeda et al. [1] proposed a deep learning-based road damage detection system using smartphone-captured images on Japanese roads, implementing real-time processing with selective transmission of pothole data to reduce bandwidth and server load. Their work demonstrated the feasibility of crowdsourced road condition monitoring using consumer devices. Arya et al. [2] extended this work using transfer learning to a multi-country setting, including India, Japan, and the Czech Republic, demonstrating improved generalisation across diverse road conditions and establishing benchmarks for cross-dataset performance. Their findings highlighted the domain gap between Western and Asian road datasets, motivating our focus on Indian-specific adaptation.

Object Detection Architectures

Redmon et al. [3] introduced the YOLO (You Only Look Once) object detection architecture, a real-time single-stage detection framework that has been widely adopted for infrastructure and road damage monitoring applications. Our systematic evaluation of 17 YOLO family variants (YOLOv8, v9, v10, v11) [4] builds upon this foundation to empirically assess their applicability to Indian road conditions, ultimately finding the approach unsuitable for segment-level quality assessment.

Vision Transformers and Image Classification

Dosovitskiy et al. [5] introduced Vision Transformer (ViT), demonstrating that transformer architectures originally designed for natural language processing could achieve state-of-the-art performance on image classification tasks when pre-trained on large datasets. Touvron et al. [6] proposed Data-efficient Image Transformers (DeiT), introducing improved training strategies and distillation techniques that enable Vision Transformers to achieve competitive performance with less training data. Initial experiments with ViT-Small in this project revealed severe overfitting (97% train / 59% val) due to the capacity-data mismatch, motivating the evaluation of hierarchical alternatives. Liu et al. [7] proposed the Swin Transformer, a hierarchical vision transformer using shifted windows that achieves strong performance across a range of image recognition tasks while requiring fewer resources than ViT. The Swin Transformer emerged as the best-performing model in our architecture sweep, achieving 79.6% validation accuracy on the final 3-class task.

Efficient Convolutional Architectures

He et al. [8] introduced Deep Residual Networks (ResNet), demonstrating that very deep networks could be effectively trained using skip connections to address the vanishing gradient problem. ResNet architectures served as CNN baselines in our architecture comparison. Tan and Le [9] proposed EfficientNet, using neural architecture search to systematically scale network width, depth, and

resolution, achieving superior accuracy-efficiency tradeoffs. Liu et al. [10] introduced ConvNeXt, a purely convolutional architecture modernised to match the design decisions of vision transformers, offering competitive performance with simpler training requirements. Both EfficientNet and ConvNeXt variants were included in our 15-architecture sweep.

Transfer Learning and Domain Adaptation

Yosinski et al. [11] investigated the transferability of features learned by deep neural networks across different tasks and datasets, establishing that transfer learning is most effective when source and target domains share visual similarities. Pan and Yang [12] provided a comprehensive survey of transfer learning techniques, categorising approaches for domain adaptation under dataset shift conditions, directly applicable to adapting ImageNet-pretrained models to Indian road infrastructure.

Active Learning and Annotation Strategies

Settles [13] provided a comprehensive survey of active learning methodologies, demonstrating that strategically selecting informative samples for labelling can reduce annotation costs by 50-90% compared to random sampling while maintaining model performance. Our active learning pipeline targets frames where max(softmax) < 0.7, consistent with uncertainty sampling principles described in this work. Roh et al. [14] surveyed sample selection strategies in active learning for deep neural networks, informing our design of confidence-threshold-based sample selection for minority class annotation.

Crowdsourcing and Data Quality

Snow et al. [15] demonstrated that non-expert crowdsourced labels can achieve quality comparable to expert annotations when using appropriate aggregation strategies such as majority voting. Our consensus-based annotation mechanism extends these principles to road quality assessment across 53 contributors and 3,216 images. Sheng et al. [16] analysed strategies for learning from multiple noisy annotators, showing that weighted voting improves label quality, motivating our multi-annotator consensus approach.

Class Imbalance in Deep Learning

Buda et al. [17] systematically studied the impact of class imbalance on deep neural network performance, comparing oversampling, undersampling, and cost-sensitive learning approaches. Their finding that class-balanced loss functions generally outperform sampling strategies informs our use of weighted CrossEntropyLoss. Wang et al. [18] proposed focal loss for dense object detection, addressing extreme class imbalance by down-weighting well-classified examples.

Data Augmentation and Regularisation

Shorten and Khoshgoftaar [19] surveyed data augmentation techniques for image classification, demonstrating that augmentation strategies must be task-appropriate to improve generalisation. Our

augmentation pipeline (RandomResizedCrop, horizontal flip, rotation, ColorJitter, GaussianBlur) is informed by their recommendations for outdoor scene understanding. Zhang et al. [20] introduced mixup, a data augmentation technique training networks on convex combinations of image pairs, representing a potential future enhancement for the road classification system.

Geospatial Analysis and Road Network Processing

Boeing [21] introduced OSMnx, a Python package for downloading, modelling, analysing, and visualising street networks from OpenStreetMap data. OSMnx forms the core of our geospatial pipeline, enabling road network download, UTM-projected GPS snapping, and quality-weighted route computation. Newson and Krumm [22] proposed a Hidden Markov Model approach to map matching for GPS trajectories, establishing principled methods for associating noisy GPS observations with road network edges. Our snapping pipeline draws on similar principles, projecting GPS points to UTM coordinates and assigning each point to the nearest road edge within a 35m threshold.

Methodology

System Overview

The road quality assessment system follows a six-stage pipeline: (1) dashcam video acquisition and frame extraction, (2) GPS coordinate extraction and automated data filtering, (3) multi-phase annotation with consensus validation, (4) model training and evaluation across a full architecture sweep, (5) batch inference and GPS-linked quality mapping on the OpenStreetMap road network, and (6) quality-aware route planning served via an interactive navigation interface. The methodology evolved through empirical experimentation, with an initial object detection phase providing quantitative justification for the transition to a classification-based approach.

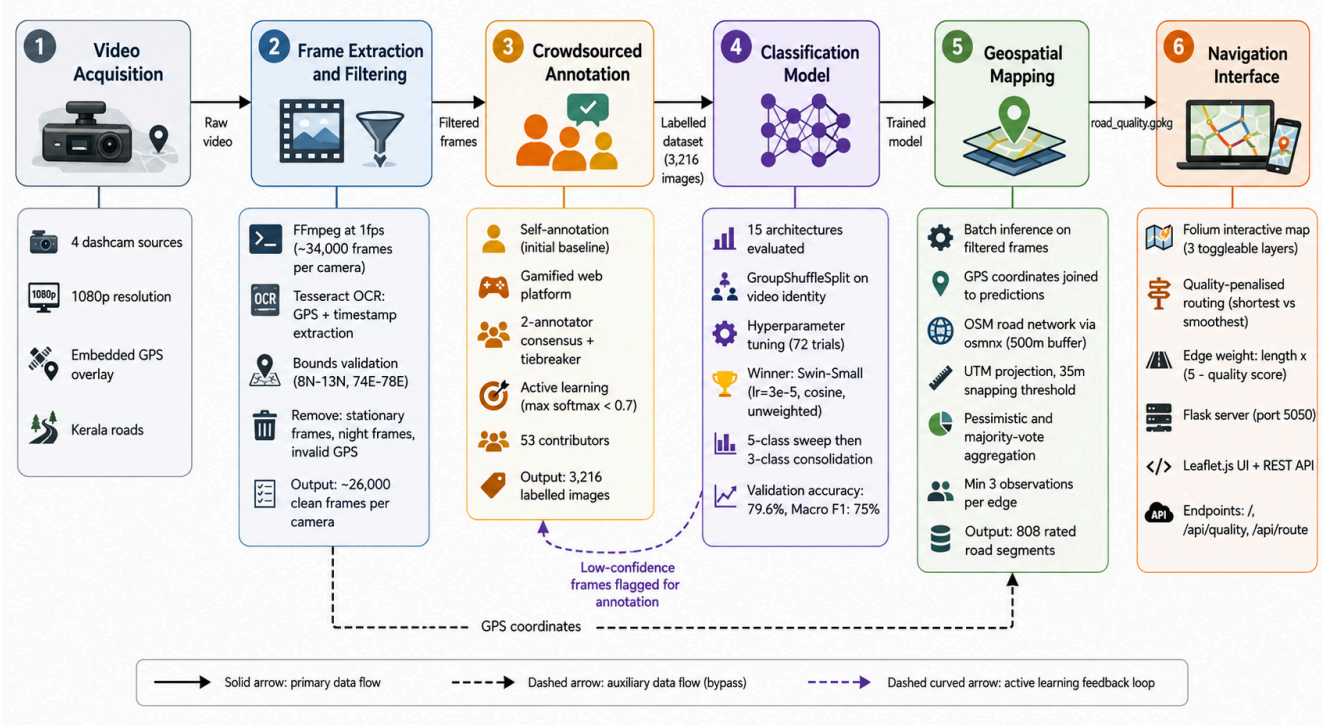


Figure 2: End-to-end road quality assessment pipeline. Solid arrows indicate primary data flow; dashed arrows indicate auxiliary bypass paths; the curved dashed arrow represents the active learning feedback loop.

Data Collection and Preprocessing

Video Acquisition

Dashcam footage was collected from diverse road types across Kerala, India, including urban arterial roads, residential streets, rural highways, and poorly maintained municipal roads. Video was captured at 1080p resolution with embedded GPS overlays containing coordinates and timestamps.

Frame Extraction and GPS Processing

Frames were extracted from video footage at 1fps using FFmpeg, yielding approximately 34,000 raw frames per camera across all collected footage. Each frame underwent OCR-based processing using Tesseract with custom preprocessing (contrast enhancement and binarisation) to extract GPS coordinates and timestamps from dashcam overlays. Extracted coordinates were validated against Kerala's geographic bounds (8°N-13°N, 74°E-78°E) to identify and remove corrupted data.

Automated Filtering Pipeline

Three automated filters were applied sequentially to ensure dataset quality:

1. **Static Vehicle Detection**: Frames captured when the vehicle was stationary, identified through timestamp gaps greater than 5 seconds or GPS coordinate invariance, were removed to focus on driving conditions.
2. **Night Image Removal**: Frames with mean pixel intensity below a fixed threshold, indicating nighttime capture with insufficient illumination for reliable road surface assessment, were excluded.
3. **GPS Validation**: Frames with invalid GPS formats, out-of-bounds coordinates, or missing location data were filtered out.

This pipeline reduced the raw frame pool to a clean annotatable set, from which 3,216 images were subsequently labelled through consensus-based crowdsourcing.

Annotation Methodology

Phase 1: Object Detection Exploration (YOLO Annotation)

Initial experimentation focused on pothole detection using the YOLO framework. Over 2,000 images were manually annotated with bounding boxes around visible potholes and road defects using Label Studio, following YOLO format specifications (class ID, normalised centre coordinates, width, height). This phase enabled systematic evaluation of detection-based approaches before committing to an alternative formulation.

Phase 2: Problem Reformulation to Classification

Following empirical findings that object detection achieved only 15-20% mAP50 with high false positive and false negative rates, the problem was reframed as a 5-class road segment quality classification task. This reformulation aligned better with the navigation use case, where segment-level quality matters more than individual pothole counts. Five quality classes were defined as shown in Table 3:

Table 3: Five-Class Road Quality Taxonomy with Annotation Criteria and Navigation Impact

Class	Criteria	Navigation Impact
Excellent	Smooth surface, no visible defects	High-speed safe
Good	Minor imperfections, small cracks	Comfortable driving
Fair	Noticeable defects, shallow potholes	Reduced comfort
Poor	Severe damage, deep potholes	Safety concern, slow navigation required
Invalid	Non-road content (sky, blur, obstructions)	Not applicable

Subsequent confusion matrix analysis and annotation consensus patterns revealed persistent ambiguity at the Excellent/Good and Fair/Poor boundaries, leading to a data-driven decision to consolidate to three classes: Good (Excellent and Good merged), Bad (Fair and Poor merged), and Invalid. The classifier supports both schemes via a runtime flag, allowing the 5-class and 3-class pipelines to coexist without reprocessing.

Phase 3: Self-Annotation

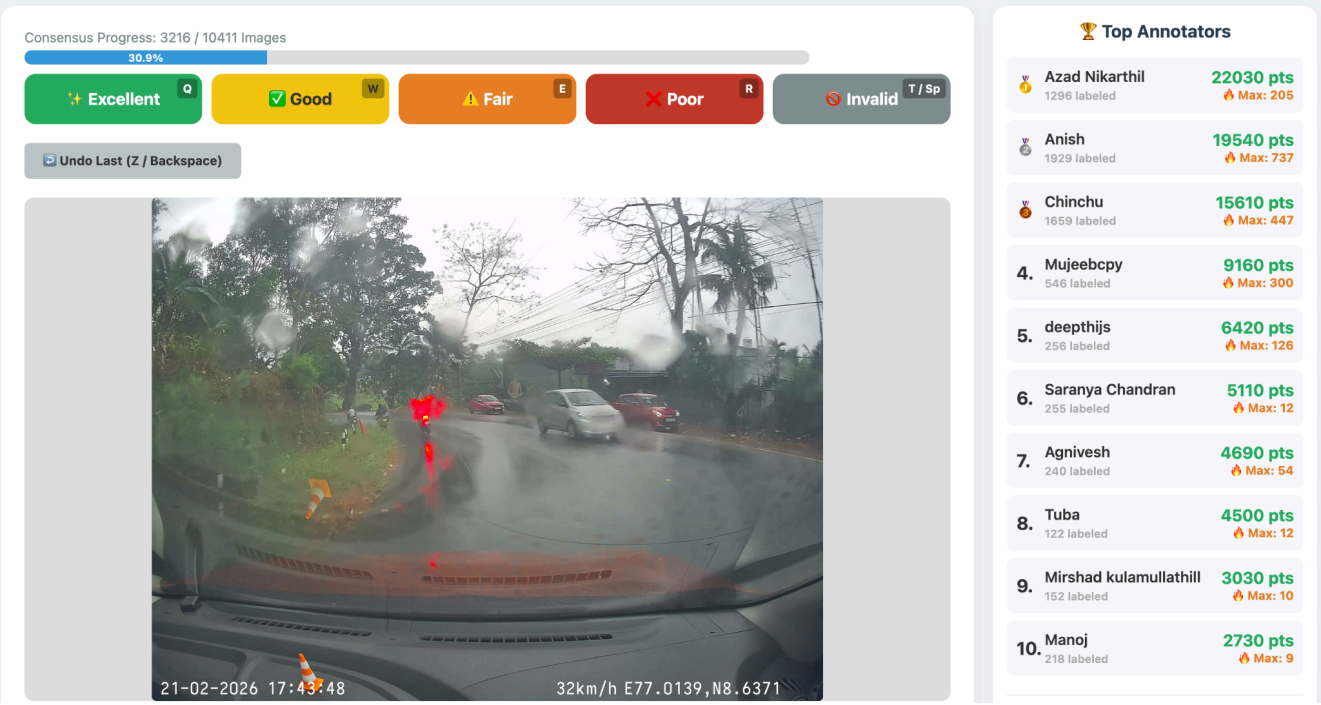
An initial classification dataset was self-annotated to establish baseline class distributions and validate the taxonomy. This phase provided early insights into annotation difficulty and inter-class boundary ambiguity, motivating the development of the consensus-based validation approach.

Phase 4: Consensus-Based Crowdsourcing

To improve annotation quality and reduce individual bias, a gamified web-based annotation platform was developed with the following features:

- **Multi-Annotator Consensus**: Each image is independently labelled by two annotators. Agreement confirms the label; disagreement triggers a third annotator as tiebreaker.
- **Gamification**: Annotators view a leaderboard showing contribution counts and accuracy scores measured against confirmed consensus labels, incentivising participation and quality.
- **Quality Control**: Annotators receive immediate feedback on their accuracy relative to consensus, enabling learning and improvement over time.

This approach yielded 3,216 consensus-validated annotations across 53 contributors, with reduced labelling noise compared to single-annotator methods.



Phase 5: Active Learning Integration

To address class imbalance, an active learning pipeline was implemented alongside crowdsourced annotation. After each training cycle, the model performed inference on the unannotated pool. Frames where max(softmax) < 0.7 were flagged and prioritised for annotation, as these represent genuinely uncertain or underrepresented cases. This targeted sampling directed annotation effort toward minority classes (Poor, Fair) and improved annotation efficiency compared to random selection.

Model Development

Object Detection Phase: YOLO Family Evaluation

Seventeen YOLO variants were systematically evaluated to assess the viability of object detection for road quality assessment:

- YOLOv8: n, s, m, l variants (3.2M to 43.7M parameters)
- YOLOv9: t, s, m, c variants
- YOLOv10: n, s, m variants
- YOLO11: n, s, m variants
- YOLO26: n, s, m variants

Models were trained for 70 epochs initially, with extended training (200+ epochs) for select variants to assess convergence. Performance was evaluated using standard object detection metrics: mAP50, mAP50-95, precision, and recall.

Classification Phase: Architecture Comparison

Following the reformulation to classification, 15 architectures were evaluated via a full sweep using the timm library:

- ResNet family: ResNet18 (11M parameters), ResNet34 (21M), ResNet50 (25M)
- EfficientNet family: EfficientNet-B0 (5M), B1 (8M), B2 (9M)
- EfficientNetV2 family: EfficientNetV2-S (22M), EfficientNetV2-M (54M)
- Vision Transformers: ViT-Small (22M), ViT-Base (86M)
- ConvNeXt family: ConvNeXt-Tiny (28M), ConvNeXt-Small (50M), ConvNeXt-Base (89M)
- Swin Transformer family: Swin-Tiny (28M), Swin-Small (50M)

All models were initialised with ImageNet-21k pre-trained weights and fine-tuned on the road quality dataset. The Swin Transformer achieved the best validation accuracy of 79.6% on the 3-class task and was selected as the final production model.

Training Configuration

Data Split: A group-aware train/validation split was implemented using GroupShuffleSplit on video identity, ensuring that frames from the same dashcam recording cannot appear in both the training and validation sets. This prevents artificially inflated validation metrics caused by near-duplicate frames from the same video sequence.

Augmentation (training set only):

- RandomResizedCrop to 224×224
- Random horizontal flip (p=0.5)

- Random rotation (±8°)
- ColorJitter (brightness, contrast, saturation, hue)
- RandomGrayscale
- GaussianBlur

Two-Phase Training:

- Phase 1: Backbone frozen; only the classification head is trained for a warm-up period
- Phase 2: All layers unfrozen; full fine-tuning at one-tenth of the Phase 1 learning rate

Training Hyperparameters:

- Optimizer: Adam
- Learning rate: 1×10^{-4} (Phase 1); 1×10^{-5} (Phase 2)
- Batch size: 32
- Maximum epochs: 100
- Scheduler: CosineAnnealingLR (ReduceLROnPlateau available as an alternative)
- Early stopping: Patience of 30 epochs on validation loss
- Label smoothing: 0.1
- Loss function: Weighted CrossEntropyLoss with class weights computed as inverse class frequency

Class Merging: Both 5-class and 3-class schemes are supported throughout the pipeline via a runtime flag (--merge-classes). When enabled, Excellent and Good are merged into Good, and Fair and Poor are merged into Bad, with Invalid retained. All downstream components including inference, evaluation, mapping, and routing adapt to whichever scheme is active.

This two-phase configuration was used as the default for the architecture sweep. Hyperparameter tuning subsequently established that end-to-end fine-tuning without backbone freezing, at a learning rate of 3e-5 with cosine annealing, outperformed this approach; the production model was trained accordingly.

Evaluation Metrics

Models were evaluated on the held-out validation set using:

- Overall accuracy
- Macro-averaged precision, recall, and F1-score (unweighted by class frequency, ensuring minority class performance is not masked)
- Per-class precision, recall, and F1-score
- Confusion matrix

Per-model results were emitted as structured JSON for automated parsing and consolidated into a summary CSV via the architecture sweep tooling.

Hardware and Software Environment

- **Hardware**: AMD GPU with ROCm support for accelerated training
- **Software**: Python 3.10, PyTorch 2.0, timm (pre-trained model library), OpenCV (preprocessing), Tesseract (OCR), osmnx (road network), Folium (map rendering), Flask (navigation server & gamification server)
- **Training time**: Approximately 1,200 to 1,300 seconds per model for 100 epochs with early stopping

Geospatial Pipeline and Navigation

Batch Inference and GPS Association

Following model training and selection, the trained Swin Transformer was applied to all filtered frames at scale via a dedicated batch inference script. Predictions were joined with the GPS coordinates extracted during preprocessing, producing a flat CSV file mapping each frame to its predicted quality class, confidence score, and geographic coordinates.

Road Network Download and GPS Snapping

The OSM drive network for the GPS bounding box was downloaded via osmnx with a 500m buffer and cached locally as a GraphML file to avoid repeated downloads. Coordinates were projected to UTM for metre-accurate distance computation. Each GPS point was snapped to the nearest road edge within a maximum threshold of 35m; points beyond this threshold were discarded to avoid incorrect road assignments.

Per-edge quality scores were aggregated using two independent strategies:

- Pessimistic: The worst quality label observed on an edge is assigned, prioritising safety for routing decisions
- Majority vote: The most frequent label wins, with ties broken in favour of the worse class

A minimum observation threshold (default: 3) was enforced per edge, filtering out segments with insufficient data. Both aggregation results are stored together in a GeoPackage file (road_quality.gpkg) alongside a plain CSV export, allowing the rendering and routing layers to switch between modes without reprocessing.

A numeric quality score was assigned to each class to enable arithmetic comparisons: Poor/Bad = 1, Fair = 2, Good = 3, Excellent = 4.

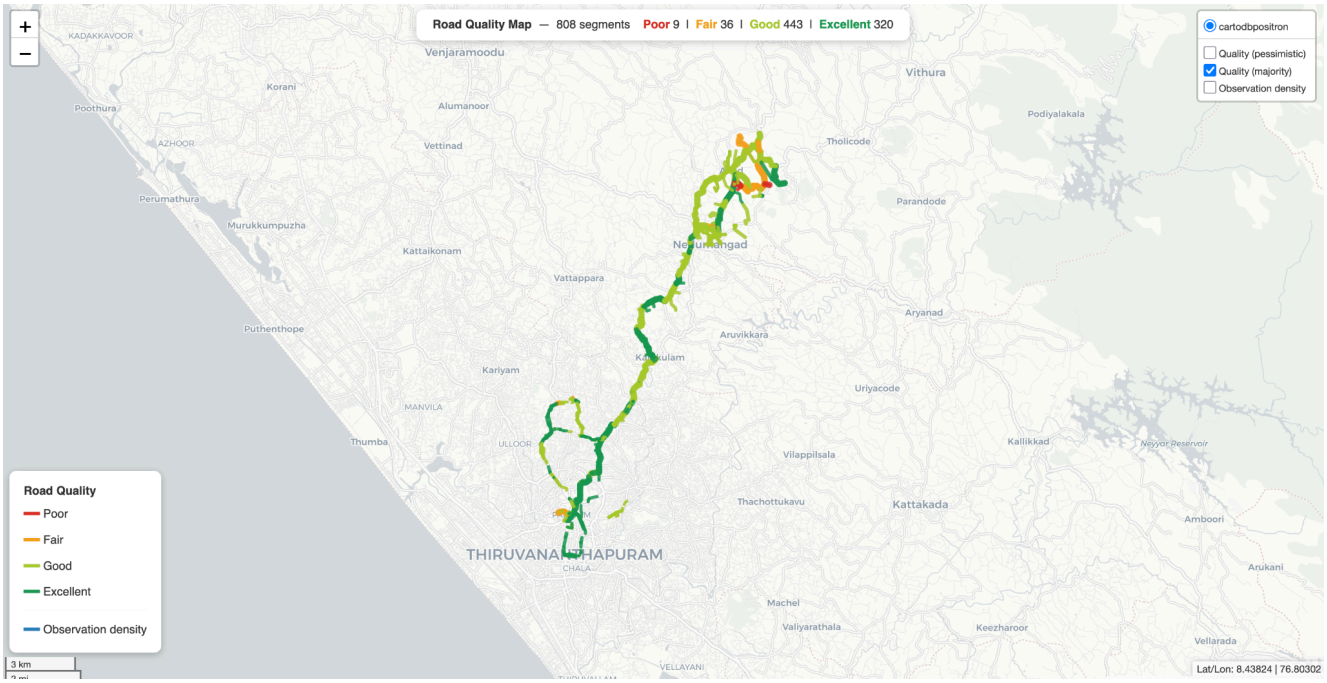
Interactive Map Rendering

An interactive HTML map was generated using Folium with a CartoDB Positron base layer. The map exposes three independently toggleable layers:

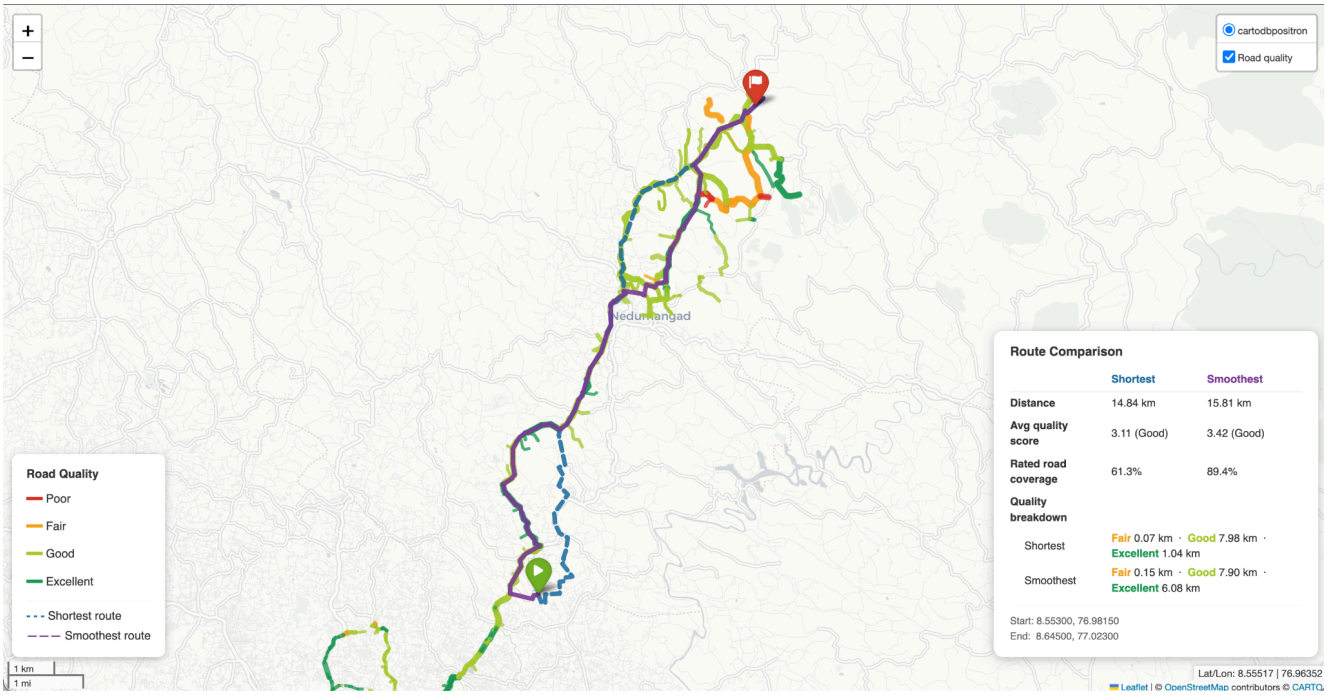
- Quality (pessimistic aggregation)
- Quality (majority vote aggregation)
- Observation density

Road segment line weight scales with observation count, ranging from 3px at low coverage to 7px at 50 or more observations. Each segment carries a hover tooltip showing street name, quality label, observation count, average score, and per-class breakdown. The legend is dynamically generated to display only quality labels present in the data.

The colour scheme follows intuitive traffic-light conventions: Poor/Bad in red, Fair in amber, Good in yellow-green, and Excellent in dark green.



Quality-Aware Route Planning



Two routes are computed between any given origin and destination using standard Dijkstra on the OSM graph:

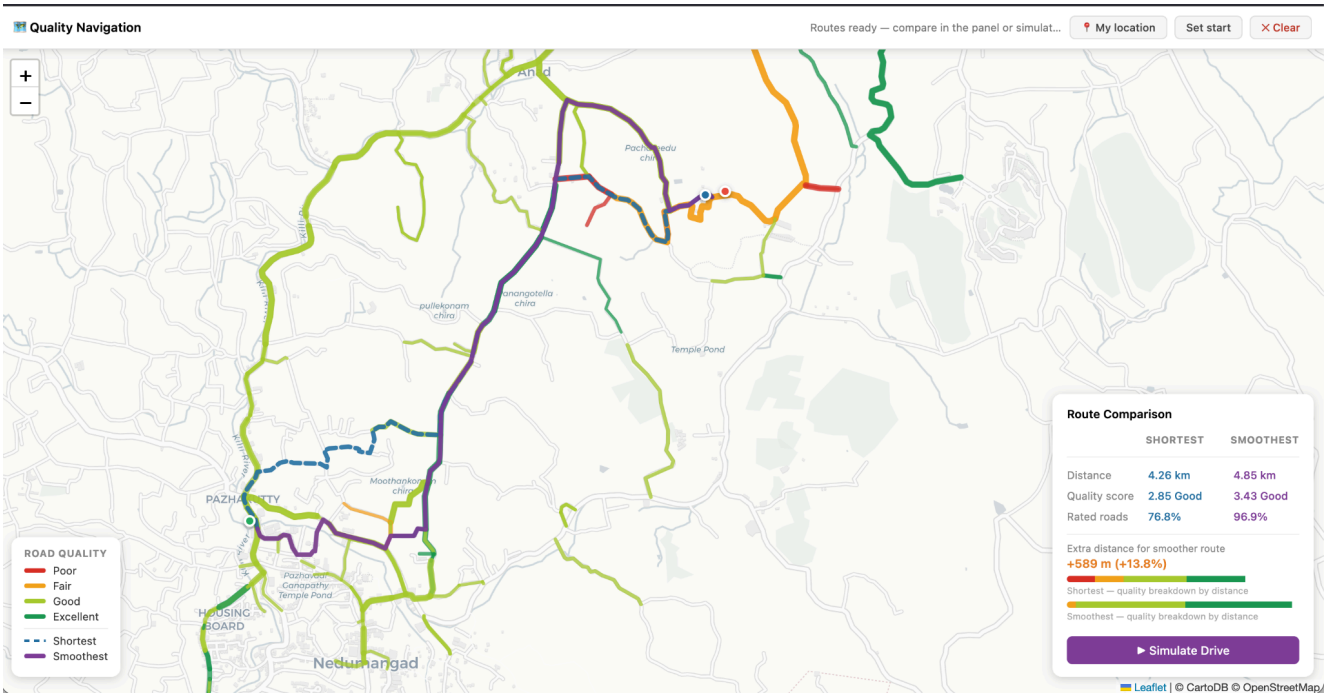
- Shortest path: Minimises total distance
- Smoothest path: Minimises a quality-penalised cost using edge weight = length × (5 − quality score)

Under this formula, a Poor road (score 1) carries a 4× length penalty relative to an Excellent road (score 4), which carries a 1× penalty. Unrated edges, where insufficient observations were available, are assigned a conservative default score of 2.5 with an additional 1.2 multiplier. Both routes are overlaid on the quality map layer (shortest in blue dashed, smoothest in purple solid) with a comparison panel showing distance, average quality score, rated coverage percentage, and per-quality segment breakdown for each route.

Navigation Server

A Flask application (port 5050) loads the OSM graph and road quality data at startup and injects quality weights into the graph. It exposes three endpoints:

- GET /: Leaflet.js navigation UI for interactive use
- GET /api/quality: GeoJSON of all rated road segments
- POST /api/route: Accepts start and end coordinates and returns both shortest and smoothest routes with full metadata including coordinates, distance, average quality score, rated coverage percentage, and per-edge quality breakdown



Experimental Design Rationale

The methodology reflects an iterative, empirically-driven research process:

1. **YOLO experimentation** provided quantitative evidence that object detection underperforms for segment-level quality assessment, with the best result of 20.3% mAP50 across 17 variants.
2. **Classification reformulation** directly addresses the navigation use case without requiring an intermediate aggregation step from detection counts to quality scores.
3. **Group-aware splitting** on video identity prevents data leakage that would otherwise inflate validation metrics on near-duplicate frames from the same recording.
4. **End-to-end fine-tuning** from the start, without backbone freezing, outperformed two-phase training across all tested durations. The substantial domain gap between ImageNet and Indian dashcam footage appears to require unrestricted early-layer adaptation, making backbone freezing counterproductive in this setting.
5. **Consensus annotation** ensures higher label quality than single-annotator approaches, critical in a small-dataset regime where noisy labels have an outsized effect on model behaviour.
6. **Active learning** maximises annotation efficiency by directing human effort toward the frames that most benefit the model, rather than sampling the unannotated pool randomly.
7. **Systematic comparison** across 15 architectures establishes a well-evidenced basis for final model selection, rather than relying on a single architecture chosen a priori.
8. **Dual aggregation modes** (pessimistic and majority vote) stored together in the GeoPackage mean that the routing and rendering layers can switch strategies without rerunning the snapping pipeline, keeping the system modular and reproducible.

Experimental Results

Object Detection Phase: YOLO Family Evaluation

Performance Summary

Seventeen YOLO variants were systematically evaluated on the pothole detection task using 2000+ annotated images with bounding boxes. Table 4 presents performance metrics for representative models across YOLO generations.

Table 4: YOLO Model Performance on Indian Road Pothole Detection

Model	Parameters	mAP50	mAP50-95	Precision	Recall	Training Time (70 ep)
YOLOv8n	3.2M	0.203	0.073	0.316	0.265	339s
YOLOv8s	11.2M	0.176	0.057	0.323	0.229	467s
YOLOv8m	25.9M	0.148	0.043	0.223	0.254	856s
YOLOv8l	43.7M	0.174	0.055	0.369	0.205	1292s
YOLOv10n	3.6M	0.189	0.069	0.362	0.215	507s
YOLO11s	9.4M	0.185	0.060	0.392	0.205	586s
YOLO26s	11.1M	0.175	0.061	0.296	0.234	706s

Best Performing Model: YOLOv8n achieved the highest mAP50 of 20.3%, despite being the smallest model tested.

Key Findings

1. Inverse Scaling Phenomenon: Larger models consistently underperformed smaller variants. YOLOv8l (43.7M parameters) achieved only 17.4% mAP50 compared to YOLOv8n's (3.2M) 20.3%. This suggests overfitting on the limited dataset, where larger capacity models memorise training patterns without learning generalisable features.

2. Low Absolute Performance: The best mAP50 of 20.3% is significantly below acceptable thresholds for practical deployment (typically >60% for production systems). This indicates fundamental challenges in applying object detection to Indian road conditions.

3. High False Positive Rate: Precision values (22-39%) reveal that 60-78% of detections are false positives. Common misclassifications include shadows cast by trees/buildings, water puddles reflecting light, road markings (lane dividers, crosswalks), and tar patches used for temporary repairs.

4. High False Negative Rate: Recall values (20-27%) indicate that 73-80% of actual potholes are missed. Undetected potholes include shallow/small defects, edge-of-frame instances with partial visibility, potholes obscured by shadows or water, and instances with unusual shapes deviating from training distribution.

5. Extended Training Ineffective: Select models trained for 200+ epochs (selected variants across families) showed minimal improvement over 70-epoch baselines, achieving only 19.0% mAP50, indicating the performance ceiling is data-limited rather than convergence-limited.

Sample YOLO Predictions

Figure 3: Representative detection results highlighting systematic failure modes.

Row 1: Correct Detections

- Image A: Urban road with a large pothole correctly detected
- Image B: Urban road with a large irregular pothole correctly detected

Row 2: False Positives

- Image C: A water droplet on the windshield misclassified as a pothole
- Image D: Road marking detected as pothole, incorrectly

Row 3: False Negatives

- Image E: Only a portion of potholes are detected. Large ones are skipped.
- Image F: Large pothole detected, but a deeper smaller hole is missed.

Representative YOLO detection results showing correct predictions (row 1), false positives (row 2), and false negatives (row 3). The model struggles with visual ambiguity inherent to Indian road scenes.

Implications for Road Quality Assessment

The low mAP50 (15-20%) and high error rates render object detection unsuitable for navigation-oriented road quality assessment. Even if individual pothole localization were improved, aggregating detections into segment-level quality scores introduces additional uncertainty. A road segment with 5 false positives and 3 missed potholes provides unreliable quality information for routing decisions. This empirical evidence motivated the reformulation to direct segment classification.

Classification Phase: Results

Dataset Characteristics

The annotated dataset comprises 3,216 consensus-labelled images, each with a single tie-broken label reflecting agreement across multiple annotators. Data was collected from 4 dashcam sources. Raw frame extraction yielded approximately 34,000 frames per camera; after automated filtering, approximately 26,000 frames per camera were retained as the annotatable pool. The train/validation split used a 70/30 ratio applied using GroupShuffleSplit on video identity, ensuring frames from the same recording cannot appear in both splits.

Table 5 presents the 5-class distribution of the final consensus dataset before merging.

Table 5: Dataset Statistics (5-class)

Class	Images	Percentage
Excellent	1,069	33.2%
Good	1,307	40.6%
Fair	399	12.4%
Poor	112	3.5%
Invalid	329	10.2%
Total	**3,216**	**100%**

The dataset reflects a 9.5:1 Excellent-to-Poor imbalance in the 5-class scheme. After class consolidation, Table 6 shows the 3-class distribution used for final model training.

Table 6: Dataset Statistics (3-class)

Class	Composition	Images	Percentage
Good	Excellent + Good	2,376	73.9%
Bad	Fair + Poor	511	15.9%
Invalid	Invalid	329	10.2%
Total		3,216	100%

The Good-to-Bad ratio of 4.6:1 in the merged scheme represents a meaningful improvement over the initial 9.2:1 Excellent-to-Poor ratio in the early 5-class dataset (1,140 images), achieved through active learning targeting minority classes throughout the crowdsourcing phase.

Initial Experiments: Vision Transformer

Initial experiments with ViT-Small on the 5-class dataset established the severity of the overfitting problem that motivated the full architecture sweep.

ViT-Small Initial Results

Table 7 summarises the initial training results.

Table 7: ViT-Small Baseline Training Results on the 5-Class Dataset

Model	Parameters	Epochs	Train Acc	Val Acc	Overfitting
vit_small_patch16_224	22.1M	34	96.96%	58.97%	37.99%

Validation Metrics (Macro-Averaged):

- Precision: 49.93%
- Recall: 50.51%
- F1-Score: 49.47%

Analysis: The 38% train/validation gap confirmed a fundamental capacity-data mismatch. Early stopping triggered at epoch 4 with no further validation improvement, and macro-averaged metrics near 50% indicated near-random performance across classes despite reasonable overall accuracy driven by majority class dominance.

Architecture Sweep Results

A full sweep across 15 architectures was conducted with fixed hyperparameters (lr=1e-4, weighted loss, 30% validation split, patience=10), completing in approximately 1.6 hours. EfficientNetV2-S and EfficientNetV2-M failed to initialise and were excluded. Table 8 presents results for the remaining 13 architectures, ranked by macro F1.

Table 8: Architecture Sweep Results (5-class, fixed lr=1e-4, weighted loss)

Rank	Architecture	Accuracy	Macro F1	Precision	Recall
1	Swin-Small	53.80%	47.00%	46.40%	48.40%
2	Swin-Tiny	51.50%	46.10%	45.90%	50.60%
3	ResNet34	51.10%	45.40%	47.70%	48.20%
4	ConvNeXt-Tiny	50.40%	44.80%	47.10%	52.90%
5	ResNet18	49.70%	44.40%	46.30%	49.30%
6	ViT-Small	49.80%	44.00%	45.00%	45.40%
7	EfficientNet-B1	49.40%	41.40%	43.00%	45.40%
8	EfficientNet-B2	49.80%	41.30%	41.90%	41.70%
9	ResNet50	48.20%	41.20%	42.60%	46.90%
10	ConvNeXt-Small	49.70%	40.50%	49.60%	41.50%
11	ViT-Base	49.70%	40.40%	43.30%	41.70%
12	EfficientNet-B0	45.40%	39.50%	39.30%	41.20%
13	ConvNeXt-Base	48.90%	39.50%	44.30%	39.20%
	EfficientNetV2-S	FAILED	N/A	N/A	N/A
	EfficientNetV2-M	FAILED	N/A	N/A	N/A

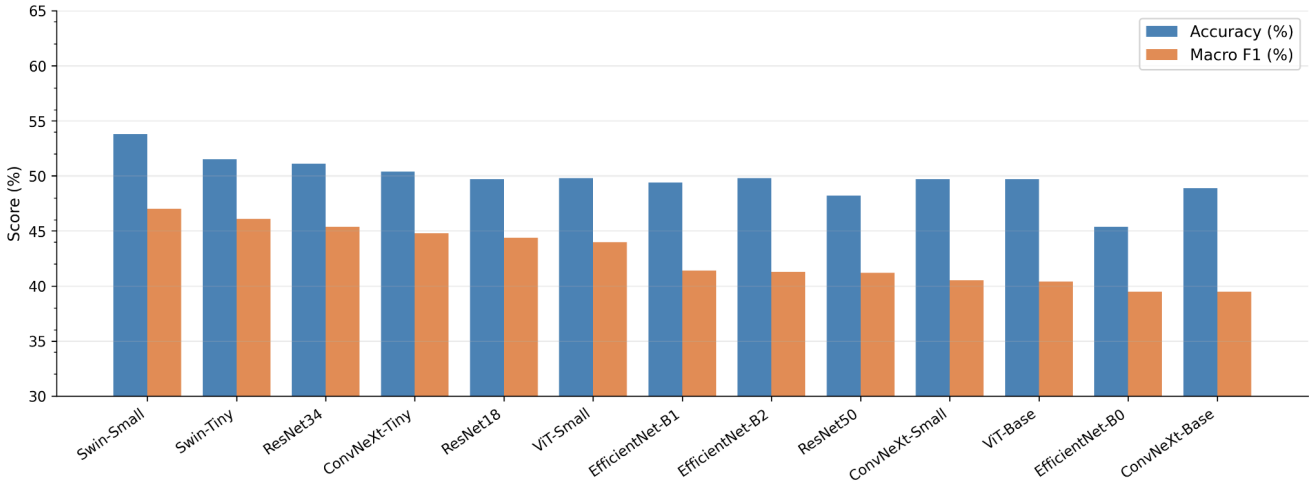


Figure 4: Validation accuracy and macro F1 across 13 architectures in the sweep (5-class, fixed lr=1e-4, weighted loss). Swin-Small ranked first on both metrics.

Key observations from the sweep:

1. Swin Transformer variants ranked first and second, with Swin-Small leading at 53.8% accuracy and 47.0% macro F1. The hierarchical shifted-window attention mechanism appears better suited to this task than isotropic designs such as ViT-Small and ViT-Base.
2. Larger models did not outperform smaller variants within the same family. ConvNeXt-Base (89M parameters) ranked 13th while ConvNeXt-Tiny (28M) ranked 4th. ViT-Base (86M) ranked 11th against ViT-Small (22M) at 6th. This inverse scaling pattern, also observed in the YOLO experiments, reflects the data-limited training regime.
3. Accuracy values were tightly clustered (45-54%) across all successful models, indicating that the 5-class task itself was the bottleneck rather than any individual architecture.

Hyperparameter Tuning

The top two models from the sweep (Swin-Small and Swin-Tiny) were carried into a 72-trial hyperparameter search covering learning rate, freeze epochs, label smoothing, class weights, and scheduler, completing in approximately 14 hours.

Table 9 presents the average macro F1 for each hyperparameter value across all 72 trials.

Table 9: Average Macro F1 by Hyperparameter Value

Parameter	Value	Avg Macro F1	
Learning rate	**3.00E-05**	51.90%	Best
	1.00E-04	51.70%	
	3.00E-04	51.70%	
Freeze epochs	**0**	52.10%	Best
	5	51.70%	
	10	51.40%	
Label smoothing	**0**	51.90%	Best
	0.1	51.60%	
Class weights	**Unweighted**	52.00%	Best
	Weighted	51.50%	
Architecture	**Swin-Small**	52.40%	Best
	Swin-Tiny	51.10%	

Table 10 lists the five highest-ranked configurations by macro F1.

Table 10: Top 5 Hyperparameter Tuning Trials Ranked by Macro F1 (5-class)

Rank	Model	LR	Freeze	Smoothing	Weights	Accuracy	Macro F1
1	***Swin-Small***	***3.00E-05***	***0***	***0***	***Unweighted***	***53.20%***	***56.80%***
2	Swin-Small	3.00E-04	5	0	Unweighted	52.30%	56.70%
3	Swin-Small	1.00E-04	0	0.1	Unweighted	52.90%	55.90%
4	Swin-Small	1.00E-04	0	0.1	Weighted	51.60%	55.80%
5	Swin-Tiny	3.00E-04	5	0	Unweighted	51.40%	55.80%

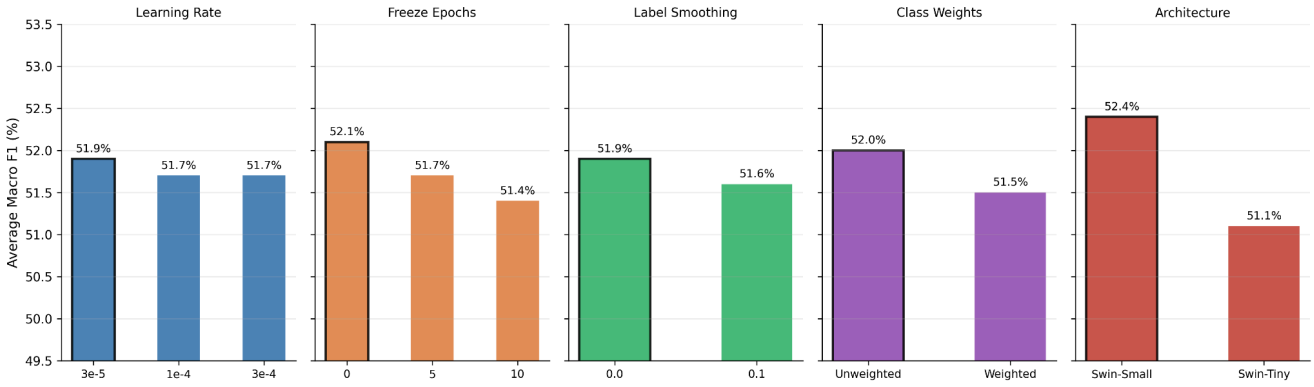


Figure 5: Average macro F1 by hyperparameter value across 72 trials. Bold outline indicates the best value per parameter. Lower learning rate, no backbone freezing, unweighted loss, and Swin-Small consistently outperformed alternatives.

Three findings from the tuning are worth noting. First, a lower learning rate (3e-5) consistently outperformed higher values, reflecting the sensitivity of Swin Transformer to learning rate when fine-tuning from ImageNet weights. Second, freezing backbone layers hurt performance across all tested durations; fine-tuning all layers from the start proved more effective on this dataset, contrary to the typical two-phase training assumption. Third, label smoothing and class weighting provided no measurable benefit, suggesting the architecture was already learning reasonably balanced representations without these corrections.

Class Structure Analysis and Consolidation

Confusion matrix analysis across all architectures in the sweep consistently showed high confusion at two boundaries: Excellent/Good and Fair/Poor. Annotator disagreement data from the crowdsourcing platform corroborated this: inter-annotator conflict was concentrated at the same boundaries, confirming that the ambiguity was genuine rather than a modelling failure.

A data-driven decision was taken to consolidate from five classes to three: Good (Excellent and Good merged), Bad (Fair and Poor merged), and Invalid retained. This was implemented as a runtime flag (--merge-classes) throughout the pipeline, preserving the 5-class option for reference.

Performance Progression

Table 11 shows accuracy and macro F1 at each stage of the modelling process.

Table 11: Performance Progression Across Training Stages

Stage	Accuracy	Macro F1
Architecture sweep best (5-class)	53.80%	47.00%
After hyperparameter tuning (5-class)	53.20%	56.80%
Production training (5-class)	53.00%	57.00%
Production training (3-class merged)	79.60%	75.00%

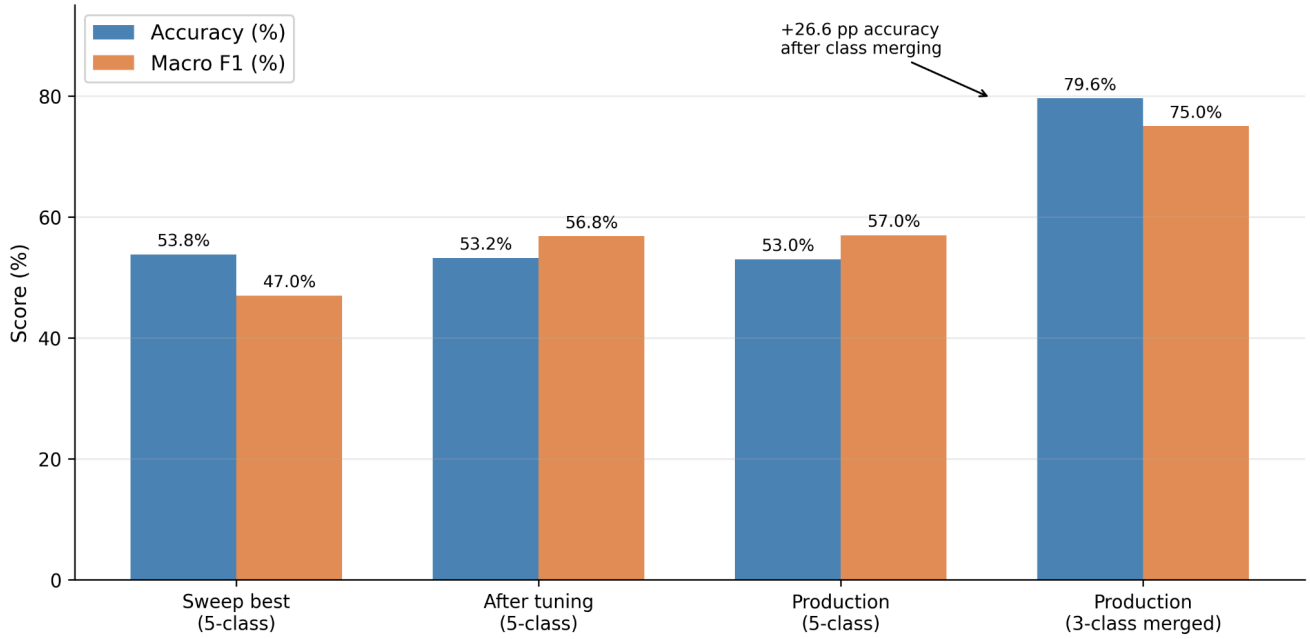


Figure 6: Accuracy and macro F1 at each stage of the modelling process. The largest gain came from class consolidation rather than architectural or hyperparameter improvements.

The improvement in macro F1 from sweep to tuning (47.0% to 56.8%) without a proportional accuracy gain reflects better minority class calibration rather than overall improvements. The jump to 79.6% accuracy and 75.0% macro F1 after class merging confirms that the 5-class boundaries were the primary bottleneck.

Final Model Performance

The production Swin-Small model trained on the 3-class consolidated dataset was evaluated on 941 held-out validation samples. Table 12 presents the per-class classification report, and Table 13 the confusion matrix.

Table 12: Per-Class Classification Report (3-class, Invalid excluded from evaluation)

Class	Precision	Recall	F1	Support
Bad	83%	53%	65%	342
Good	78%	94%	85%	599
Macro avg	80%	73%	75%	941
Weighted avg	80%	79%	78%	941
Overall accuracy	79.6%			941

Table 13: Confusion Matrix

	Predicted Bad	**Predicted Good**
True Bad	182	160
True Good	38	561

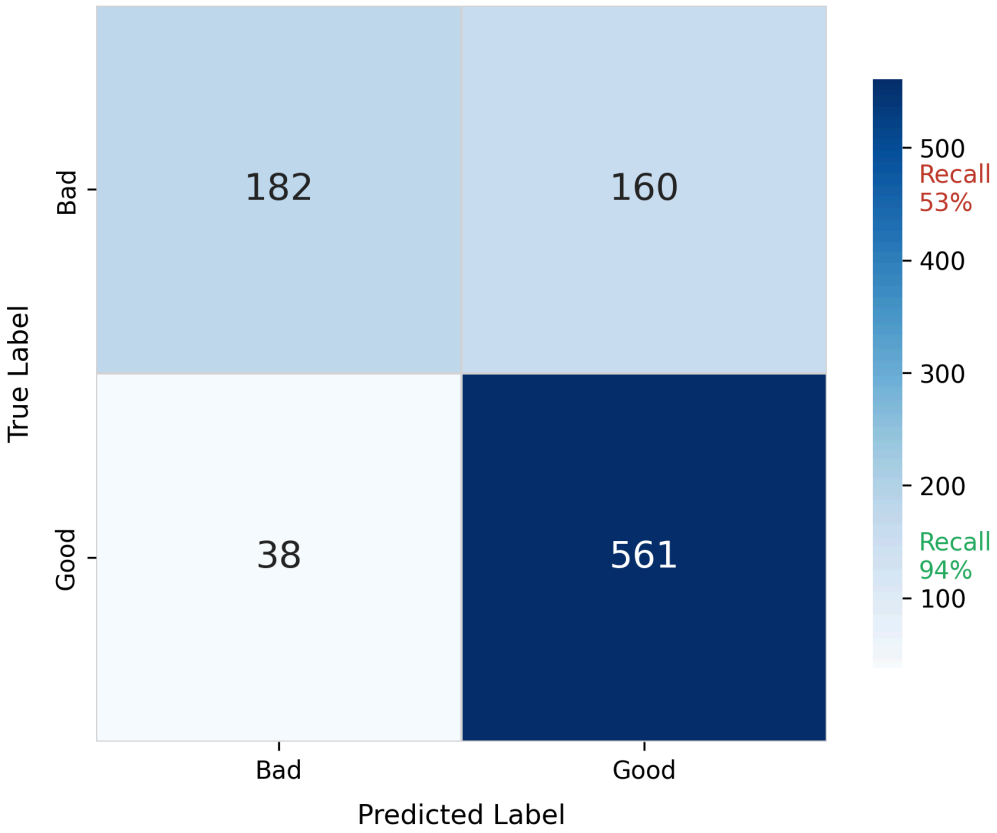


Figure 7: Confusion matrix for the final Swin-Small model on 941 validation samples (Bad and Good only; Invalid excluded from evaluation). The model achieves 94% recall on Good roads but misses 47% of Bad segments.

The model performs well on Good roads (94% recall) but misses 47% of Bad road segments, classifying 160 Bad samples as Good. This asymmetry has practical implications for the navigation use case: the model is conservative in flagging bad roads, which means routes computed by the quality-penalised algorithm may occasionally route drivers through segments worse than predicted. This is discussed further in the Discussion section.

The training log shows validation accuracy converging to 79.6% over 14 epochs before early stopping triggered on validation loss (patience=10, best validation loss of 0.5807 at epoch 4).

Active Learning Impact

The active learning pipeline, which targeted frames where max(softmax) < 0.7 for priority annotation, grew the dataset from the initial 1,140-image 5-class baseline to 3,216 unique images with improved minority class representation. The Bad class, which comprised only 4.1% of the initial dataset (47 Poor samples out of 1,140), grew to 511 images (15.9% of the final dataset), a direct result of targeted annotation toward low-confidence and minority class samples throughout the crowdsourcing phase.

Comparison: Object Detection vs. Classification

Table 14 summarises the key differences between the two approaches across the dimensions evaluated.

Table 14: Object Detection vs. Classification Approach Comparison

Criterion	**YOLO (Object Detection)**	**Classification (Final)**
Primary Metric	mAP50: 20.3%	Val Accuracy: 79.6%, Macro F1: 75.0%
Best Model	YOLOv8n (3.2M parameters)	Swin-Small (50M parameters)
Models Evaluated	17 variants	13 successful out of 15
Annotation Effort	High (bounding boxes per pothole)	Medium (image-level labels)
False Positive Pattern	Shadows, puddles, road markings	Good class over-prediction (conservative on Bad)
Alignment with Use Case	Indirect; requires aggregation	Direct; segment quality in single forward pass
Road Segments Mapped	Not applicable	808 segments on OSM network
Final Status	**Completed; approach abandoned**	**Completed; Swin-Small deployed in production pipeline**

Sample Classification Predictions (Earlier ViT-Small Model)

Figure 8 presents representative predictions from the earlier ViT-Small model, demonstrating both capabilities and failure modes.

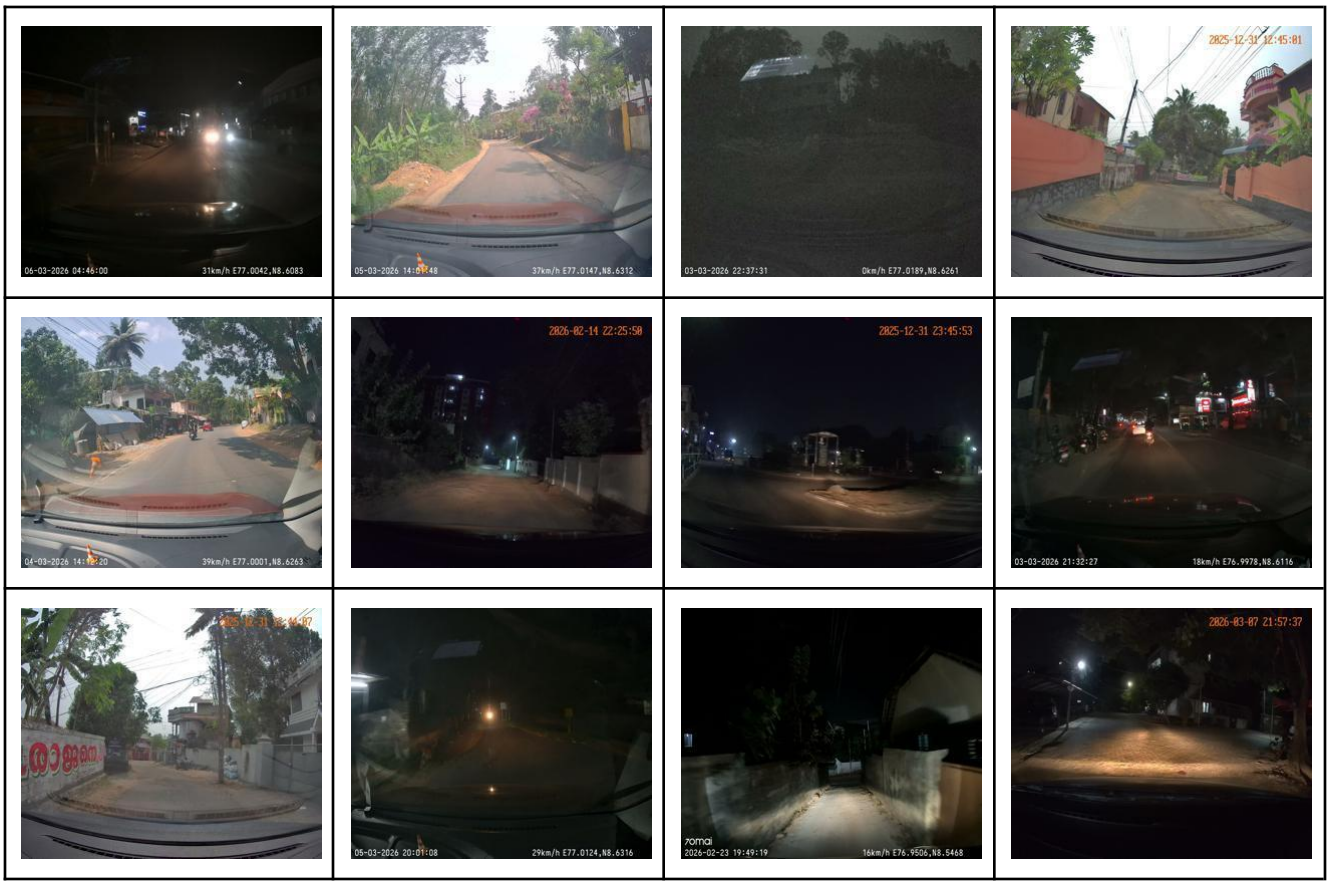


Figure 8: Sample predictions from ViT-Small demonstrating correct classifications (Row 1), high-confidence errors from overfitting (Row 2), and low-confidence predictions targeted for active learning (Row 3). Confidence scores <0.5 are flagged for human review.

Note: Row 1, Col 3 and Row 2 images appear dark as they represent nighttime/invalid captures.

Correct Predictions (Row 1):

- Col 1: **Excellent** | Pred: Excellent (0.64) | GT: Excellent ✓
- Col 2: **Good** | Pred: Good (0.85) | GT: Good ✓
- Col 3: **Invalid** | Pred: Invalid (0.995) | GT: Invalid ✓
- Col 4: **Fair** | Pred: Fair (0.84) | GT: Fair ✓

High-Confidence Errors (Row 2):

- Col 1: **Boundary Confusion** | Pred: Good (0.69) | GT: Excellent ✗
- Col 2: **Class Bias** | Pred: Good (0.68) | GT: Fair ✗
- Col 3: **Majority Class Bias** | Pred: Excellent (0.62) | GT: Good ✗
- Col 4: **Overfitting Pattern** | Pred: Excellent (0.73) | GT: Good ✗

Low-Confidence Predictions (Row 3):

- Col 1: **Critical Misclassification** | Pred: Invalid (0.39) | GT: Poor ✗
- Col 2: **Underconfident Correct** | Pred: Good (0.47) | GT: Good ✓
- Col 3: **Minority Class Confusion** | Pred: Fair (0.48) | GT: Poor ✗
- Col 4: **Active Learning Target** | Pred: Good (0.33) | GT: Good ✓

Discussion

Why Object Detection Failed

The systematic evaluation of 17 YOLO variants yielded consistently poor performance (mAP50 15-20%). Several factors explain this:

Visual Complexity: Indian roads present high visual clutter including vendors, pedestrians, animals, shadows, and water puddles. Precision values of 22-39% across all variants reveal that 60-78% of detections were false positives, with shadows and puddles being the primary confounders.

Pothole Variability: Indian potholes vary dramatically in size (5cm to 2m), depth, shape, fill state (water, debris, tar), and surface type (asphalt, concrete, gravel). This high intra-class variance challenges models trained on more uniform damage patterns.

Task Misalignment: Detection answers "where are potholes?" but navigation requires "is this segment safe to drive?" A road with 10 small potholes may be safer than one with 2 deep failures. Aggregating detection counts into segment-level quality scores introduces additional uncertainty and requires ad-hoc heuristics with no principled basis.

Inverse Scaling: Larger YOLO variants (YOLOv8l: 43.7M parameters) consistently underperformed smaller ones (YOLOv8n: 3.2M), confirming that performance was data-limited rather than capacity-limited. Extended training to 200+ epochs produced no meaningful improvement, further confirming this ceiling.

Why Vision Transformers Overfit

ViT-Small (96.96% train / 58.97% val) exhibited a textbook capacity-data mismatch. With 22M parameters and a limited training set, the parameter-to-sample ratio far exceeded what can be sustained without memorisation.

Unlike convolutional architectures which have built-in inductive biases (local connectivity, translation equivariance), isotropic vision transformers rely on learned attention from scratch at each layer. This flexibility, an advantage at ImageNet scale, becomes a liability when fine-tuning on a small domain-specific dataset. The model memorised training patterns rather than learning transferable features, as evidenced by validation improvement stopping at epoch 4 while training accuracy climbed to 97% over 34 epochs.

The same pattern repeated with ViT-Base (86M parameters), which ranked 11th in the architecture sweep despite having four times the capacity of ViT-Small. Across all 13 successful architectures in the sweep, the top performers were all hierarchical or residual architectures (Swin-Small, Swin-Tiny, ResNet34, ConvNeXt-Tiny) rather than the deeper or larger variants in each family.

Why Swin Transformer Won

The Swin Transformer's hierarchical design, processing image patches at multiple scales through shifted windows, appears well-suited to road quality assessment. Road surface condition manifests at multiple spatial scales simultaneously: fine-grained texture (crack patterns, surface roughness) and coarser structure (pothole extent, patch repairs, lane-level damage). Swin's multi-scale representation captures both, whereas isotropic transformers operate at a single resolution throughout.

Additionally, Swin's shifted window mechanism constrains attention to local neighbourhoods in early layers before expanding scope in deeper layers, providing a degree of locality bias that reduces the data requirement compared to global self-attention. This makes it more practical for domain-specific fine-tuning on datasets of a few thousand images.

The Class Merging Decision

The consolidation from five classes to three (Good, Bad, Invalid) produced the single largest performance gain in the project: accuracy increased from 53% to 79.6% and macro F1 from 57% to 75%. This gain is not simply a consequence of having fewer classes to distinguish; it reflects that the original 5-class boundaries were genuinely ambiguous rather than learnable.

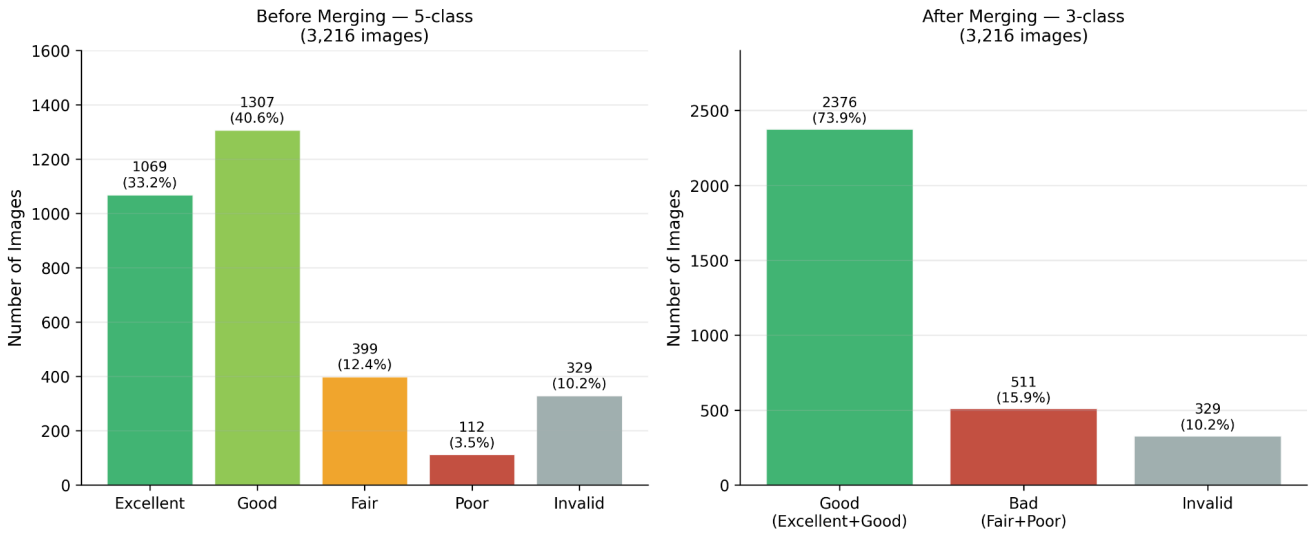


Figure 9: Dataset class distribution before and after consolidation. Merging Excellent and Good into Good, and Fair and Poor into Bad, reduced the number of decision boundaries while better reflecting the navigation use case.

Confusion matrix analysis across all 13 architectures in the sweep consistently showed the highest off-diagonal mass at the Excellent/Good and Fair/Poor boundaries. Annotator consensus data corroborated this: inter-annotator disagreement was disproportionately concentrated at these same boundaries. When two experienced annotators consistently disagree on whether a road is Excellent or

Good, or Fair versus Poor, the distinction is either too subjective for image-level labelling or requires information (e.g. tactile feedback, vehicle speed, road context) unavailable from dashcam frames alone.

From a navigation standpoint, the 3-class scheme is also more practically useful. A routing system needs to know whether a segment is safe and comfortable (Good), degraded and potentially damaging (Bad), or not a road at all (Invalid). The finer-grained distinctions within Good and within Bad add precision that the system cannot reliably deliver and the navigation use case does not require.

The Bad Recall Problem

The most significant limitation of the final model is its Bad class recall of 53%: nearly half of Bad road segments are classified as Good. Of 342 Bad samples in the validation set, 160 were incorrectly predicted as Good. This asymmetry has direct implications for navigation safety. The quality-penalised routing algorithm relies on per-segment scores to identify smoother paths; segments misclassified as Good receive no penalty, meaning the smoothest route may occasionally include Bad segments that the model failed to flag.

Two factors contribute to this. First, the training dataset remains imbalanced even after class merging: Good comprises 73.9% of images and Bad 15.9%, a 4.6:1 ratio that biases the model toward the majority class. Second, the visual boundary between a mildly degraded Good road and a lightly damaged Bad road is inherently ambiguous in dashcam images, particularly at driving speed where motion blur reduces surface texture visibility.

The pessimistic aggregation mode in the mapping pipeline partially mitigates this: when a road segment has multiple observations, the worst label wins, which can recover Bad classifications from individual frames even if some were incorrectly predicted Good. However, this only helps for segments with sufficient observation coverage (minimum 3 observations per edge).

Hyperparameter Tuning Surprises

The 72-trial hyperparameter search produced several findings that contradict common practice:

Freezing backbone layers hurt performance. The intuition behind two-phase training (freeze backbone, warm up head, then fine-tune) is that pre-trained features are valuable and should be protected during early training. On this dataset, however, end-to-end fine-tuning from the start consistently outperformed any degree of backbone freezing. This may reflect the substantial domain gap between ImageNet and Indian dashcam footage: the pre-trained features require significant adaptation, and restricting early updates slows this adaptation without benefit.

Class weights gave no advantage. Weighted CrossEntropyLoss, which up-weights minority class loss contributions, is a standard response to class imbalance. Here it provided no benefit over unweighted loss, and in several trials slightly hurt macro F1. This suggests the Swin Transformer's attention

mechanism was already distributing representational capacity across classes more evenly than the loss weighting assumed.

Label smoothing made no difference. With a small dataset, label smoothing is often recommended to prevent overconfident predictions. The tuning results showed negligible difference between smoothing=0.0 and smoothing=0.1, suggesting the model was not overconfident in the way label smoothing is designed to address.

Methodological Contributions

Empirically-Driven Problem Reformulation: The reformulation from object detection to classification was justified by quantitative evidence from 17 YOLO variants rather than theoretical reasoning. Similarly, the subsequent consolidation from 5 to 3 classes was driven by confusion matrix patterns and annotator consensus data rather than convenience. Both decisions are fully documented and reproducible.

Consensus Annotation for Subjective Labels: Road quality assessment involves inherently subjective judgements, particularly at class boundaries. The two-annotator consensus mechanism with a third-annotator tiebreaker reduces noise from individual bias. The gamification layer (leaderboard, per-annotator accuracy feedback) sustained engagement across 53 contributors, demonstrating that non-expert crowdsourcing can produce usable training data for nuanced visual tasks with appropriate quality controls.

Active Learning for Minority Class Recovery: Starting from a 9.2:1 class imbalance (Excellent to Poor) in the initial 1,140-image dataset, active learning targeting low-confidence predictions grew the Bad class from 47 images (4.1%) to 511 images (15.9% of 3,216) in the final dataset. This is a more targeted approach than random annotation and significantly reduced the imbalance ratio from 9.2:1 to 4.6:1.

End-to-End Navigation Pipeline: The complete system from dashcam footage to interactive route planning via a REST API represents a practical, replicable template for road quality monitoring. The dual aggregation strategy (pessimistic and majority vote stored together in GeoPackage) and the quality-penalised Dijkstra routing are design decisions with clear engineering rationale that generalise beyond this specific deployment.

Limitations and Mitigations

Bad Class Recall (53%): The model misclassifies nearly half of Bad segments as Good, which affects routing reliability on less-observed segments. Mitigation: pessimistic aggregation recovers some Bad predictions at the segment level; a larger and more balanced training dataset would be the most effective long-term fix.

Geographic and Seasonal Coverage: The dataset covers Kerala roads in a single pre-monsoon

season. Road conditions during and after the monsoon, when damage accelerates significantly, are not represented. Mitigation: additional data collection across seasons and regions; the pipeline infrastructure is in place to incorporate new footage without architectural changes.

Dataset Size: 3,216 images is modest for a deep learning classification task, particularly for the Bad class (511 images). Mitigation: the annotatable pool of approximately 104,000 filtered frames across 4 cameras provides substantial headroom for continued active learning without additional data collection.

Minimum Observation Threshold: Road segments with fewer than 3 GPS-matched observations are excluded from the quality map. In areas with sparse dashcam coverage, this leaves gaps in the quality layer that the routing algorithm fills with a conservative default score. Mitigation: additional camera passes on under-covered routes.

Key Lessons

1. Problem formulation matters more than architecture: the 26-percentage-point accuracy gain from class merging dwarfs any architectural improvement observed in the sweep (1-3% range). Defining the right task is the highest-leverage decision in the pipeline.
2. Capacity must match data: the inverse scaling pattern appeared in both the YOLO experiments and the classification sweep. On datasets of a few thousand images, mid-capacity architectures (28-50M parameters) consistently outperform larger ones.
3. Conventional mitigations do not always apply: class weights, label smoothing, and backbone freezing, all standard practices for small imbalanced datasets, provided no benefit here. Hyperparameter tuning on the actual data is more reliable than applying rules of thumb.
4. Label quality over label quantity: 3,216 consensus-validated images produced a more useful model than 2,000+ noisily annotated YOLO bounding boxes. In low-data regimes, annotation quality has a larger effect on model performance than raw count.
5. Pessimistic aggregation is appropriate for safety-critical routing: given the model's tendency to under-predict Bad segments, routing decisions should prefer the worst observed label on a segment rather than the average or majority.

Implications for Deployment

The system demonstrates that a crowdsourced, dashcam-based road quality assessment pipeline is practically feasible with consumer hardware and open-source tools. The OSM-based mapping layer means the system is immediately deployable in any region with road network data, which covers virtually all of India. The Flask navigation server, while currently a local prototype, provides the API surface needed for integration with a mobile application or third-party navigation platform.

For municipal use, the GeoPackage output provides a directly usable dataset for maintenance prioritisation: Bad-rated segments with high observation counts represent high-confidence targets for repair, while low-observation segments indicate areas needing additional survey coverage.

The primary barrier to broader deployment is the Bad recall limitation. For a navigation system, falsely routing drivers through Bad roads is a more serious error than unnecessarily avoiding Good ones. Addressing this through continued active learning on Bad class samples, and potentially through a lower classification threshold for the Bad class, should be the first priority in any production deployment.

Research Contributions

Empirical Validation: Systematic comparison of 17 detection and 13 classification architectures (30 models total) with quantified performance metrics, establishing that segment-level classification outperforms defect detection for navigation-oriented road quality assessment on Indian roads.

Data Quality Methodology: Consensus-based crowdsourcing framework with gamification and active learning integration, scaling to 3,216 consensus-validated images across 53 contributors while reducing the Excellent-to-Poor class imbalance from 9.2:1 to 4.6:1 through targeted annotation.

End-to-End Pipeline: Complete implementation from dashcam video ingestion through OCR-based GPS extraction, classification, OSM road network snapping, interactive quality mapping, and quality-penalised route planning served via REST API. The pipeline is modular, reproducible, and directly extensible to new geographic regions and camera sources.

Domain-Specific Findings: Quantification of the performance gap between standard training practices (class weights, label smoothing, backbone freezing) and their actual effectiveness on Indian dashcam data, providing empirical guidance for future work in this domain.

Conclusion and Future Work

Conclusion

This dissertation set out to develop a road quality assessment system for Indian road conditions using dashcam imagery and GPS-based mapping, with the end goal of enabling quality-aware navigation. What began as a pothole detection problem evolved, through systematic empirical investigation, into a substantially different and more effective formulation.

The initial phase established that object detection is poorly suited to the navigation use case on Indian roads. Testing 17 YOLO variants across four model generations yielded a best mAP50 of 20.3%, with high false positive rates driven by visual clutter and high false negative rates on small or occluded potholes. More fundamentally, aggregating per-pothole detections into segment-level quality scores introduces compounding uncertainty that undermines routing reliability. This empirical evidence, rather than theoretical preference, motivated the reformulation to direct segment classification.

The classification phase evaluated 13 architectures, with Swin-Small emerging as the best performer. A second data-driven reformulation, consolidating five classes into three (Good, Bad, Invalid) based on confusion matrix evidence and annotator consensus patterns, improved accuracy from 53% to 79.6% and macro F1 from 57% to 75%. The dataset grew from 1,140 images with a 9.2:1 class imbalance to 3,216 consensus-validated images across 53 contributors, with the Good-to-Bad ratio reduced to 4.6:1 through active learning. These results confirm that problem formulation and data quality have a larger impact on outcomes than architectural choice.

The complete pipeline was fully implemented and validated, mapping 808 road segments on the OpenStreetMap network and serving quality-penalised route comparisons via a Flask navigation API. The primary limitation is the Bad class recall of 53%: nearly half of degraded road segments are classified as Good, which affects routing reliability on less-observed segments. The pessimistic aggregation strategy partially mitigates this at the segment level, but improving Bad recall remains the most important open problem for production deployment.

Overall, the project makes a practical and replicable contribution to infrastructure monitoring: an open-source pipeline deployable in any region with OSM road data, operated with consumer dashcams, and scalable through crowdsourced annotation without expert labellers.

Future Work

This project was developed as a proof of concept, demonstrating the feasibility of a crowdsourced, dashcam-based road quality assessment pipeline. Several directions for extending the system follow from this foundation.

Mobile Navigation Application

The most impactful next step is a mobile application that integrates the quality-aware routing capability into a driver-facing interface. The Flask navigation server already exposes the route planning functionality via REST API, providing a ready backend for a mobile frontend. Such an application would allow drivers to request quality-penalised routes in real time, choosing between the shortest and smoothest path based on their preference. Passive footage collection from the same application would simultaneously feed new dashcam frames into the pipeline, creating a self-reinforcing data collection loop.

Expanded Dashcam Coverage

The current dataset covers footage from 4 cameras across a limited set of routes in Kerala. Broader coverage across more roads, drivers, and time periods would improve both model robustness and map completeness. In particular, monsoon and post-monsoon footage is entirely absent from the current dataset, despite representing the conditions where road quality is most variable and most relevant to drivers. Additional cameras would also increase observation counts on under-covered segments, reducing the reliance on default quality scores for unrated edges.

Wider Geographic Deployment

The pipeline infrastructure is built on OpenStreetMap and is not geographically restricted. Extending deployment beyond the current coverage area, initially to other parts of Kerala and subsequently to other states, would validate the system's generalisability and demonstrate its utility as a practical road monitoring tool at scale. The current implementation serves as a proof of concept; the architecture is designed to accommodate this expansion without fundamental changes.

References

[1] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road damage detection using deep neural networks with images captured through a smartphone," arXiv preprint arXiv:1801.09454, 2018.

[2] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, and Y. Sekimoto, "Transfer learning-based road damage detection for multiple countries," arXiv preprint arXiv:2008.13101, 2020.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788.

[4] Ultralytics, "YOLOv8 Documentation," 2023. [Online]. Available: https://docs.ultralytics.com/

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in Proc. Int'l Conf. Learning Representations (ICLR), 2021.

[6] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in Proc. Int'l Conf. Machine Learning (ICML), 2021, pp. 10347-10357.

[7] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical vision transformer using shifted windows," in Proc. IEEE Int'l Conf. Computer Vision (ICCV), 2021, pp. 10012-10022.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778.

[9] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. Int'l Conf. Machine Learning (ICML), 2019, pp. 6105-6114.

[10] Z. Liu, H. Mao, C. Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2022, pp. 11976-11986.

[11] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in Advances in Neural Information Processing Systems (NeurIPS), 2014, pp. 3320-3328.

[12] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Trans. Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345-1359, 2010.

[13] B. Settles, "Active learning literature survey," Computer Sciences Technical Report 1648, University of Wisconsin-Madison, 2009.

[14] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: A big

data-AI integration perspective," IEEE Trans. Knowledge and Data Engineering, vol. 33, no. 4, pp. 1328-1347, 2021.

[15] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng, "Cheap and fast — but is it good? Evaluating non-expert annotations for natural language tasks," in Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP), 2008, pp. 254-263.

[16] V. S. Sheng, F. Provost, and P. G. Ipeirotis, "Get another label? Improving data quality and data mining using multiple, noisy labelers," in Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, 2008, pp. 614-622.

[17] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," Neural Networks, vol. 106, pp. 249-259, 2018.

[18] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in Proc. IEEE Int'l Conf. Computer Vision (ICCV), 2017, pp. 2980-2988.

[19] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," Journal of Big Data, vol. 6, no. 1, pp. 1-48, 2019.

[20] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in Proc. Int'l Conf. Learning Representations (ICLR), 2018.

[21] G. Boeing, "OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks," Computers, Environment and Urban Systems, vol. 65, pp. 126-139, 2017.

[22] P. Newson and J. Krumm, "Hidden Markov map matching through noise and sparseness," in Proc. ACM SIGSPATIAL Int'l Conf. Advances in Geographic Information Systems, 2009, pp. 336-343.

Particulars of the Supervisor and Examiner

	Supervisor	**Additional Examiner**
Name	Dr. Kavya Manohar	Mr. Ashik Salahudeen
Qualification	PhD Electronics and Communication Engineering	Bachelor of Technology in Electrical Engineering
Designation	ML Researcher	Senior software Engineer
Employing Organization and Location	Adalat AI, Thiruvananthapuram, Kerala	Auxmoney GmbH, Franz-Jacob-Str 3, 10369 Berlin
Phone No.(with STD Code)	+91 9400044565	+4916099581092
Email Address	sakhi.kavya@gmail.com	aashiks@gmail.com

Remarks of the Supervisor

The final report shows how much this project has grown since the interim stage, and it is good to see that most of the earlier concerns have been acknowledged and genuinely addressed. The dataset has grown from a small initial set to 3,216 images contributed by 53 people, making sure the harder, less-represented road conditions got proper attention. Model accuracy has improved considerably, reaching 79.6% with a macro F1 of 75%, owing to a smart, data-driven decision to consolidate five classes into three rather than forcing the model to learn distinctions that even human annotators consistently disagreed on. The annotation process is now backed by a multi-annotator consensus mechanism, making it much more reliable than basic labelling. The student has also gone beyond the original scope — evaluating 30 models in total, running 72 hyperparameter trials and delivering a working navigation interface that maps quality scores onto real OpenStreetMap roads and compares routes .The one honest limitation — that bad roads are still missed about half the time — is clearly stated and well explained, which reflects good research thinking. Overall, this is a practically grounded, and academically solid dissertation with a clear path toward publication and further development.

Information about the Supervisor:
Dr. Kavya Manohar holds PhD in Electronics and Communication Engineering from APJ Abdul Kalam Technological University, Kerala. She has publication track records in peer-reviewed journals and conferences. Kavya currently leads the Speech and Language Research at Adalat AI as the founding ML Researcher.

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
WORK INTEGRATED LEARNING PROGRAMMES (WILP) DIVISION
SECOND SEMESTER OF ACADEMIC YEAR 2025-2026

SEZG628T : Dissertation Final Evaluation

STUDENT ID No.	2024TM93051
NAME OF THE STUDENT	ANISH A.
STUDENT'S EMAIL ADDRESS	aneesh.nl@gmail.com
STUDENT'S EMPLOYING ORGANIZATION & LOCATION	BizIntelligence Technologies Pvt. Ltd., Thiruvananthapuram, Kerala
SUPERVISOR'S NAME	Dr. Kavya Manohar
SUPERVISOR'S EMPLOYING ORGANIZATION & LOCATION	Adalat AI, Thiruvananthapuram, Kerala
SUPERVISOR'S EMAIL ADDRESS	sakhi.kavya@gmail.com
ADDITIONAL EXAMINER'S NAME	Mr. Ashik Salahudeen
ADDITIONAL EXAMINER'S EMPLOYING ORGANIZATION & LOCATION	Auxmoney GmbH, Franz-Jacob-Str 3, 10369 Berlin
ADDITIONAL EXAMINER'S EMAIL ADDRESS	aashiks@gmail.com
DISSERTATION TITLE	Road Quality Assessment System Using Computer Vision and GPS-Based Mapping

		
Signature of Student	**Signature of Supervisor**	**Signature of Additional Examiner**
Name: Anish A.	**Name: Dr. Kavya Manohar**	**Name: Ashik Salahudeen**