PREVIA TECH
AI-Powered Infrastructure Solutions

Road Audit AI System

Technical Documentation & Model Architecture Report

Comprehensive overview of AI models for road surface classification, pothole detection, and infrastructure monitoring.

Version 1.0 | February 2026

Prepared for: [Client Name]

Prepared by: Previa Tech AI Engineering Team

Table of Contents

1. Executive Summary

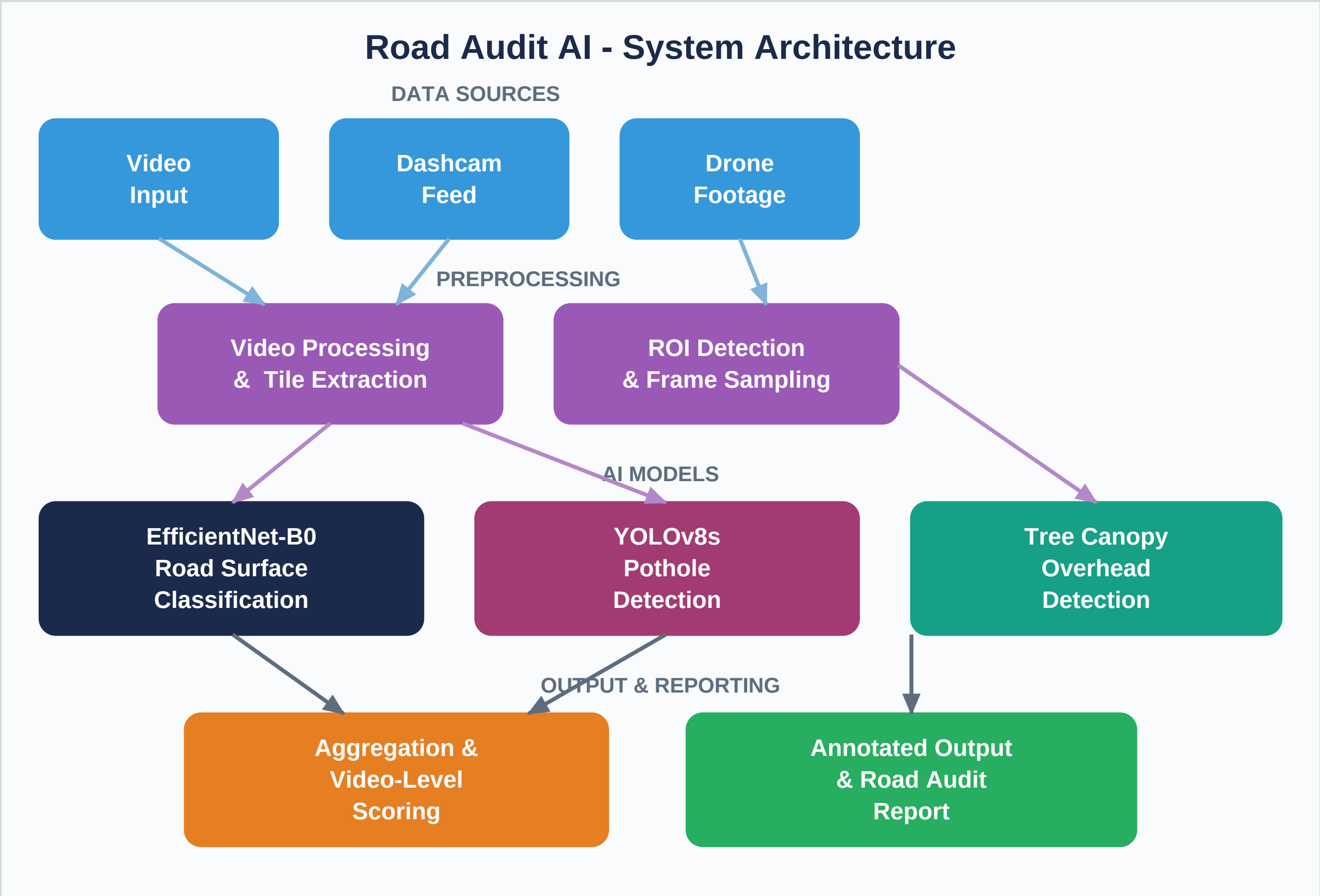
Previa Tech has developed an end-to-end AI-powered Road Audit System designed to automate the assessment of road infrastructure conditions. The system processes dashcam and drone video footage through a suite of deep learning models to detect and classify road surface degradation, potholes, and overhead tree obstructions in real time.

The solution leverages state-of-the-art computer vision architectures including EfficientNet-B0 for tile-based surface classification and YOLOv8 for object detection, achieving high accuracy with low inference latency suitable for real-time deployment scenarios.

KEY HIGHLIGHTS
3 AI Models: Surface Classification | Pothole Detection | Tree Detection
Real-time Processing: 7.2ms inference per frame on GPU
Transfer Learning: Two-stage training with ImageNet pretrained backbones
Production Ready: Batch inference, frame throttling, multi-device support

2. System Architecture Overview

The Road Audit AI System follows a modular architecture with four primary layers: data ingestion, preprocessing, AI model inference, and output aggregation. Each layer is independently scalable and supports both batch and real-time processing modes.

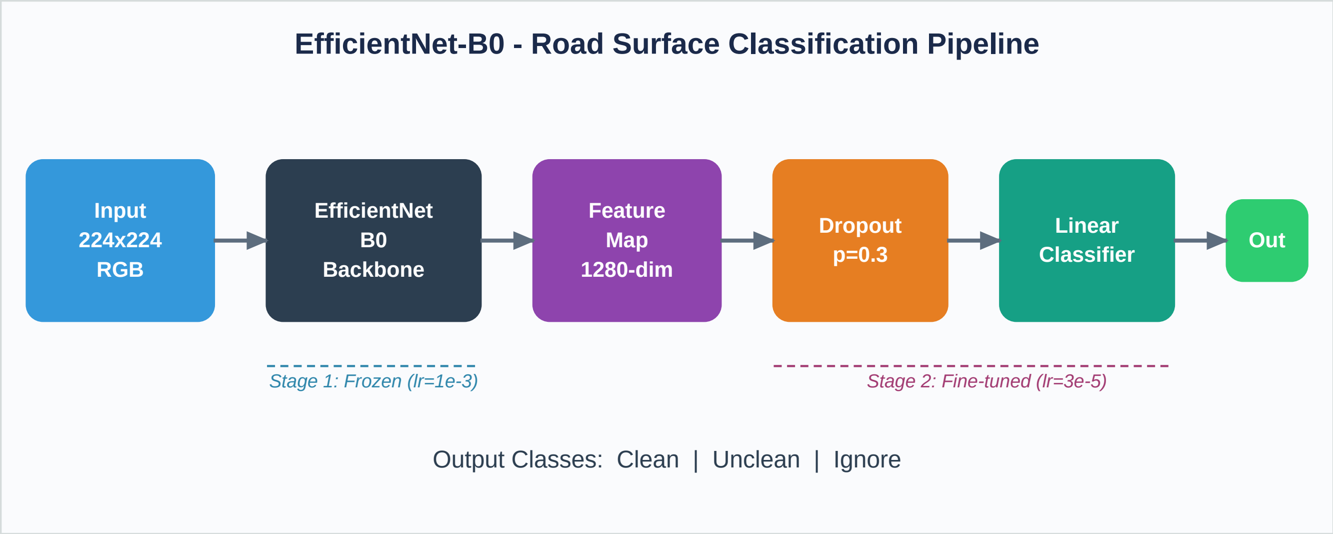


The architecture is designed for flexibility. Video inputs from dashcams, drones, or pre-recorded files are first validated through quality control checks, then processed through Region of Interest (ROI) extraction to isolate relevant road surface areas. Extracted tiles are then fed into the appropriate AI models running in parallel, and results are aggregated into video-level and road-segment-level scoring.

3. Road Surface Classification (EfficientNet-B0)

3.1 Model Architecture

The road surface classifier uses EfficientNet-B0 as its backbone, pretrained on ImageNet-1K. The model performs tile-level classification of 224x224 pixel image patches extracted from video frames, categorizing each tile as Clean, Unclean, or Ignore.



3.2 Model Specifications

Parameter	Value	Notes
Architecture	EfficientNet-B0	Pretrained on ImageNet-1K V1
Input Size	224 x 224 x 3	RGB normalized
Total Parameters	~4,010,110	Full model
Trainable (Stage 1)	~2,562	Classifier head only
Output Classes	3 (or 2)	Clean / Unclean / Ignore
Dropout Rate	0.3	Before final linear layer
Feature Dimension	1,280	Output of backbone
Normalization	ImageNet Stats	mean=[0.485, 0.456, 0.406]

3.3 Two-Stage Transfer Learning Strategy

Training follows a two-stage transfer learning approach optimized for small datasets:

Parameter	Stage 1 (Frozen)	Stage 2 (Fine-tune)
Learning Rate	1e-3	3e-5
Epochs	10-15	5-10
Layers Trained	Classifier head only	All layers
Optimizer	AdamW (wd=1e-4)	AdamW (wd=1e-4)
Scheduler	CosineAnnealingLR	CosineAnnealingLR
Batch Size	64 (GPU) / 128 (Colab)	64 (GPU) / 128 (Colab)
Loss Function	CrossEntropy (weighted)	CrossEntropy (weighted)

3.4 Data Augmentation

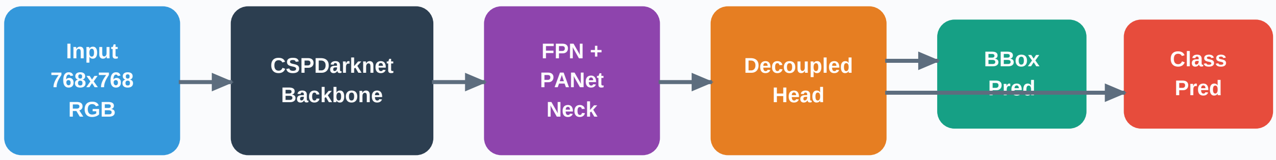
Technique	Probability	Parameters
Horizontal Flip	50%	Random left-right flip
Brightness/Contrast	80%	Factor range: 0.85 - 1.15
Color Saturation	50%	Factor range: 0.9 - 1.1

4. Pothole Detection (YOLOv8s)

4.1 Model Architecture

For localized pothole detection, the system employs YOLOv8s (You Only Look Once v8 - Small variant) from Ultralytics. This single-stage object detector provides real-time bounding box predictions with high precision, identifying the exact location and extent of potholes in road imagery.

YOLOv8s - Pothole Detection Pipeline



Single Class: Pothole | 93 Layers | 25.8M Parameters | 78.7 GFLOPs

Augmentation: Mosaic + MixUp + HSV + Horizontal Flip + Scale + Rotation

4.2 Model Specifications

Parameter	YOLOv8s (Primary)	YOLOv8m (Secondary)
Architecture Layers	93	~196
Parameters	25,840,339	~25.9M
GFLOPs	78.7	~78.9
Input Size	768 x 768	768 x 768
Detection Classes	1 (Pothole)	1 (Pothole)
Pretrained On	COCO (yolov8s.pt)	COCO (yolov8m.pt)
Batch Size	16	8

4.3 Training Configuration

Parameter	Value
Optimizer	AdamW

Parameter	Value
Initial Learning Rate	0.003
LR Schedule	Cosine Annealing
Total Epochs	120
Early Stopping Patience	25 epochs
Warmup Epochs	3
Image Caching	Enabled (RAM)
Data Split	80% Train / 10% Val / 10% Test

4.4 Data Augmentation Pipeline

Category	Technique	Value
Color	HSV Hue	0.015
Color	HSV Saturation	0.7
Color	HSV Value	0.4
Geometric	Rotation	+/- 5 degrees
Geometric	Translation	10%
Geometric	Scale	50%
Flip	Horizontal	50% probability
Mixing	Mosaic	100% (disabled last 15 epochs)
Mixing	MixUp	10%

5. Overhead Tree Detection

The Overhead Tree Detection module is designed to identify trees and vegetation that encroach over road corridors, posing hazards to tall vehicles and reducing visibility. This module extends the same tile-based classification infrastructure used for road surface analysis.

DEVELOPMENT STATUS
Current Phase: Research & Prototyping
Approach: Tile-based classification using shared EfficientNet backbone
Integration: Will leverage existing ROI extraction and inference pipeline
Target: Classify overhead canopy cover as Safe / Hazardous / Trimming Required

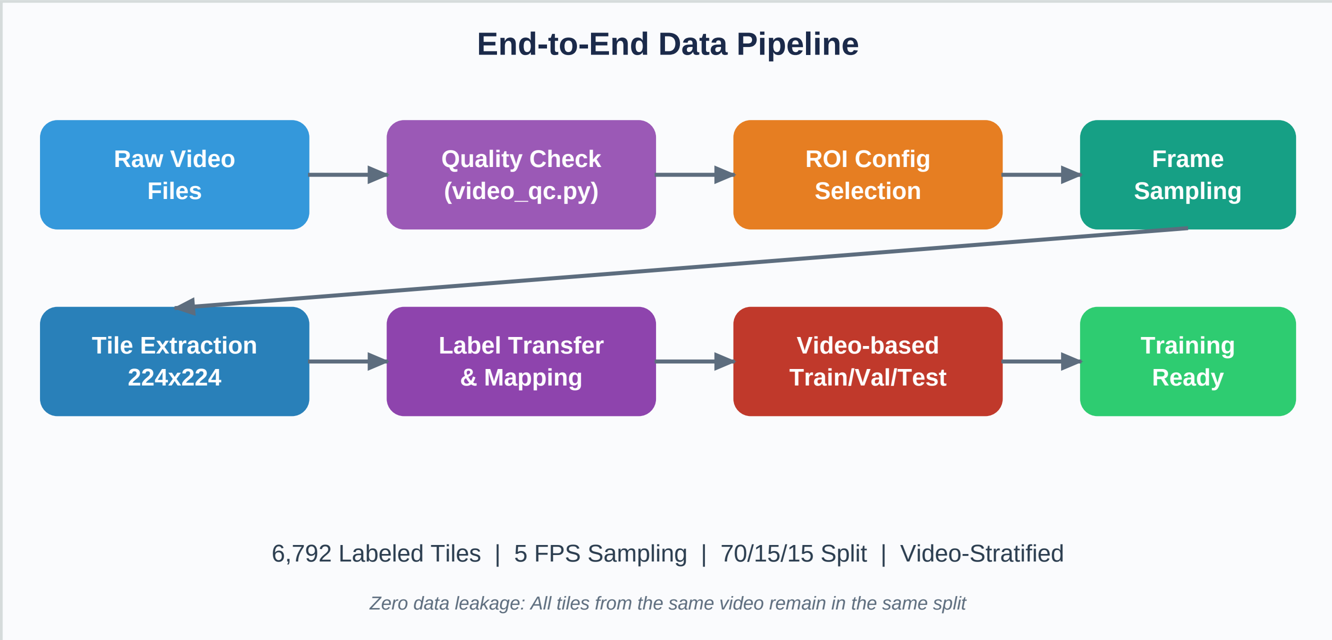
5.1 Planned Architecture

The tree detection model will reuse the proven two-stage transfer learning framework with EfficientNet-B0, adapted with a modified ROI configuration targeting the upper portion of video frames (sky-canopy region). The model will classify tiles extracted from the overhead zone to detect vegetation density and encroachment levels.

Specification	Details
Backbone	EfficientNet-B0 (shared architecture)
Input Size	224 x 224 tiles from overhead ROI
Target Classes	Clear Sky / Light Canopy / Dense Canopy / Hazardous
ROI Region	Upper frame (sky and tree canopy zone)
Training Approach	Two-stage transfer learning (same as surface model)
Integration	Parallel inference alongside surface & pothole models

6. Data Pipeline & Preprocessing

The data pipeline transforms raw road survey videos into model-ready training tiles through a robust, automated workflow with built-in quality controls and zero data leakage guarantees.



6.1 Video Quality Control

Every input video undergoes automated quality assessment before entering the processing pipeline. The QC module (video_qc.py) analyzes multiple quality dimensions including brightness levels, blur detection via Laplacian variance, and black/occluded frame identification. Each video receives a KEEP or REMOVE recommendation along with a visual montage preview for manual review.

6.2 ROI Configuration

Region of Interest (ROI) defines the precise area of each video frame to analyze. The system supports multiple pre-configured ROI templates optimized for different road types:

ROI Config	Use Case	Frame Size	Coverage
roi_highway.json	Full highway view	848 x 480	Wide trapezoid
roi_highway_shorter.json	Narrow highway focus	848 x 478	Reduced perspective
roi_highway_blackline.json	Highway with center line	848 x 480	Center-masked
roi_sector.json	Urban/street sectors	848 x 480	Street-level view

6.3 Tile Extraction & Labeling

Step	Description	Output
Frame Sampling	Extract frames at 5 FPS	JPEG frames
Tile Extraction	Cut 224x224 tiles from ROI	6,792 labeled tiles
Label Transfer	Map labels with overlap > 0.3	CSV with metadata
Data Splitting	Video-stratified 70/15/15	train/val/test CSVs

DATA INTEGRITY GUARANTEE
All splits are performed at the video level: every tile from the same video stays in the same split. This prevents data leakage and ensures the model is evaluated on truly unseen road segments.

7. Real-Time Inference Pipeline

The system includes three inference deployment modes optimized for different performance requirements, from basic frame-by-frame processing to high-throughput batch inference with intelligent frame throttling.

Real-Time Inference Pipeline

Devices: CUDA (GPU) | MPS (Apple Silicon) | CPU (Fallback)

Confidence Threshold: 0.7 | Batch Processing | Frame Throttling Option

Color Coding: Green = Clean | Red = Unclean | Cyan = ROI Boundary

7.1 Deployment Modes

Mode	Script	Best For
Basic	deploy_inference_realtime.py	Debugging, single frame analysis
Batch	deploy_inference_realtime_fast.py	GPU acceleration, 2-3x faster
Throttled	deploy_inference_realtime_fast_throttled.py	Real-time streaming, configurable FPS

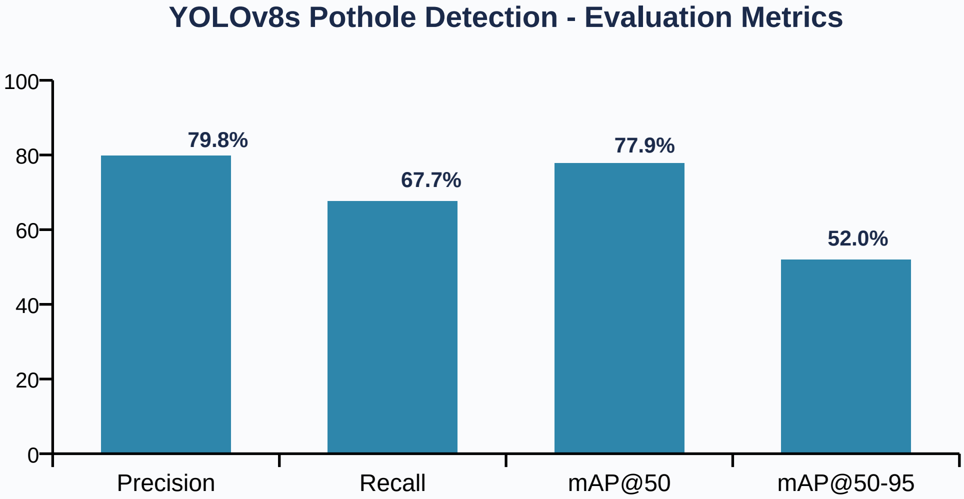
7.2 Inference Configuration

Parameter	Default	Description
Confidence Threshold	0.7	Minimum softmax probability for detection
Device Priority	CUDA > MPS > CPU	Automatic device selection
FPS Display	30-frame rolling avg	Real-time performance monitor
Output Encoding	BGR Annotated Video	Bounding boxes + labels overlay
Screenshot Capture	Press "S" key	On-demand frame capture

8. Evaluation Results & Metrics

8.1 YOLOv8s Pothole Detection Results

The YOLOv8s model was evaluated on a held-out test set of 67 images containing 161 pothole instances. The model achieves strong precision with competitive recall across multiple IoU thresholds.



Test Set: 67 images, 161 pothole instances | Inference: 7.2ms/image

Metric	Value	Description
Precision	79.8%	Of predicted potholes, 79.8% are correct
Recall	67.7%	67.7% of actual potholes are detected
mAP@50	77.9%	Mean Average Precision at IoU=0.50
mAP@50-95	52.0%	Mean AP across IoU thresholds 0.50-0.95

8.2 Inference Speed Benchmarks

Stage	Time (ms/image)	Notes
Preprocessing	4.0 ms	Image resize and normalization
Model Inference	7.2 ms	Forward pass on GPU
Postprocessing	3.0 ms	NMS and bbox decoding
Total Pipeline	14.2 ms	~70 FPS throughput

8.3 Road Surface Classification

The EfficientNet-B0 surface classifier is trained with class-balanced weighting to handle the natural imbalance between clean (54.4%) and unclean (45.6%) road tiles. The video-level aggregation uses Top-K mean scoring (top 15% of tiles) to provide robust road segment scores, minimizing the effect of individual tile noise on the final assessment.

Aggregation Metric	Method	Purpose
score_topk_mean	Mean of top 15% unclean tiles	Robust contamination score
dirty_frac_p>0.5	Fraction with p(unclean)>0.5	Overall dirtiness ratio
n_tiles_used	Count of non-ignored tiles	Coverage metric

9. Deployment Architecture

The system is designed for flexible deployment across edge devices and cloud infrastructure. The modular design allows individual models to be deployed independently or as a unified pipeline.

Component	Technology	Details
Model Format	PyTorch (.pt)	CPU/CUDA compatible checkpoints
Object Detection	Ultralytics YOLOv8	Native ONNX export supported
GPU Support	CUDA / MPS / CPU	Auto-detection at runtime
Video Processing	OpenCV (cv2)	Real-time capture and annotation
Batch Inference	PyTorch DataLoader	Configurable batch size
Frame Throttling	Custom sampler	Adjustable processing FPS
Checkpoint Structure	Dict with metadata	Model + optimizer + history

9.1 Model Checkpoint Contents

```
CHECKPOINT STRUCTURE (.pt files)
  epoch:        Training epoch number
  stage:        Training stage (1=frozen, 2=fine-tuned)
  model:        Model state dictionary (weights)
  optimizer:    Optimizer state for resume
  class_names:  ['clean', 'unclean', 'ignore']
  val_loss:     Best validation loss achieved
  val_acc:      Best validation accuracy
  history:      Full training curves (loss, acc per epoch)
```

10. Technology Stack & Dependencies

Category	Technology	Purpose
Deep Learning	PyTorch + torchvision	Model training and inference
Object Detection	Ultralytics YOLOv8	Pothole detection framework
Computer Vision	OpenCV (cv2)	Video processing and annotation
Image Processing	Pillow (PIL)	Image loading and transforms
Data Handling	pandas + NumPy	Dataset management and operations
ML Evaluation	scikit-learn	Metrics, confusion matrix, reports
Progress Tracking	tqdm	Training and inference progress bars
Model Backbone	EfficientNet-B0	ImageNet pretrained features

11. Future Roadmap

The Road Audit AI System is designed for continuous improvement and extension. The following enhancements are planned for upcoming releases:

Phase	Feature	Description	Status
Phase 1	Road Surface Model	EfficientNet-B0 classification	Complete
Phase 1	Pothole Detection	YOLOv8s bounding box detection	Complete
Phase 2	Tree Detection	Overhead canopy classification	In Progress
Phase 2	ONNX Export	Cross-platform model deployment	Planned
Phase 3	Crack Detection	Fine-grained road damage types	Planned
Phase 3	Dashboard & API	Web-based reporting interface	Planned
Phase 4	Edge Deployment	Embedded device optimization	Planned
Phase 4	GPS Integration	Geo-tagged road condition maps	Planned

This document provides a comprehensive technical overview of the Previa Tech Road Audit AI System. For questions, demonstrations, or integration discussions, please contact the Previa Tech engineering team.

CONTACT INFORMATION
Company: Previa Tech Pvt. Ltd.
Email: shubhamfufal8@gmail.com
Document: Road Audit AI - Technical Documentation v1.0
Date: February 2026