

Pothole Road Surface Segmentation Using Classical Machine Learning: A Comparative Study from Random Forest to Gradient Boosting.

Dian Rakhmawati Lestari¹, Fadhlan Nur Rachman², Matthew Ken Susanto³,
Nasauramecca Nour Haqqanshah Shodiqin⁴, Nicholas⁵

^{1,2,3,4,5}*School of Computer Science, Computer Science – Intelligent System*

Abstract

Potholes are a persistent hazard to traffic safety and road infrastructure. This paper presents a comparative study of classical machine-learning approaches for pixel-wise pothole segmentation from road-surface images, without relying on deep learning. The first project (Random Forest + SLIC) implements two unsupervised baselines, Adaptive Gaussian Thresholding and K-Means clustering, followed by a supervised Random Forest classifier trained on Simple Linear Iterative Clustering (SLIC) superpixels with nine hand-crafted features. Following a recommendation from the course lecturer to explore gradient-boosting methods, a second pipeline (Boosting) was developed, implementing XGBoost, CatBoost, and LightGBM within a redesigned pixel-level pipeline comprising 36 engineered features and hard-negative road images from the RDD2022 India dataset. Experiments on the ARA 7.0 pothole dataset have 498 images, 80:20 train-validation split, show that LightGBM with the 36-feature pixel-level pipeline achieves the best segmentation quality: mean Intersection over Union (mIoU) of 0.592, Dice coefficient of 0.451, and pixel accuracy of 87.4%, compared with mIoU 0.518 and Dice 0.277 for the Random Forest baseline. Two Streamlit web applications are deployed for interactive demonstration of both approaches.

Keywords: Pothole Detection, Road Surface Segmentation, SLIC Superpixels, Random Forest, Gradient Boosting, LightGBM, XGBoost, CatBoost, Classical Machine Learning.

I. Introduction

a. Background

Highways play a vital role in sustaining social mobility and the engine of the economy (Badan Pusat Statistik, 2023). However, in Indonesia, approximately 140,000 to 160,000 kilometers of regional roads 30%–40% of the total road network are currently in a damaged condition. Potholes are a form of infrastructural damage that often happens due to extreme weather, vehicle weight, and lack of road maintenance (Eriksson et al., 2008). This issue poses a considerable safety risk for road users, such as accidents, vehicle damage, and heavy traffic. Currently, road damage monitoring is done manually, which is a time consuming and labor intensive process. Thus, this project aims to develop a lightweight and efficient automatic pothole segmentation system using machine learning to precisely detect potholes.

b. Research Problem

Manual road inspection is inefficient and requires significant time and human resources, making it difficult to monitor extensive road networks effectively. Although deep learning-based approaches have demonstrated high performance in pothole detection, they often require large labeled datasets and substantial computational resources, limiting their applicability on low-cost or resource-constrained devices. Therefore, there is a need for a lightweight machine learning-based approach that can automatically detect potholes while maintaining acceptable detection performance and computational efficiency.

c. Objective and Significance

This study has three main objectives: first, to implement and evaluate multiple classical computer-vision models for pothole segmentation on a common benchmark dataset. Second, to investigate whether a gradient-boosting approach with enriched pixel-level features and hard-negative augmentation improves upon a Random Forest baseline trained on SLIC superpixels and third, to deploy interactive web applications for both approaches, enabling practical demonstration and further evaluation. The significance lies in providing an evidence-based model comparison that helps practitioners select the appropriate accuracy versus speed trade-off for field deployment.

II. Literature Review

a. Unsupervised vs Supervised Pothole Detection Approaches

Unsupervised learning approaches have been widely implemented for pothole detection on roadways. One such study was conducted by Buza, Omanovic, and Huseinovic (2013), who developed an automated detection method based on image processing and spectral clustering. In the initial stage, the system converts the image to grayscale, performs thresholding using Otsu's method, and removes linear objects such as cracks based on eccentricity parameters. The candidate areas are then processed using spectral clustering to reduce the dimensionality of the histogram data, followed by grouping via the K-means algorithm.

In practice, the unsupervised method that is used in the research of Buza, Omanovic, and Huseinovic (2013) is economical because it only requires a standard camera. Additionally, it also has less risk of overfitting due to eliminating the need of using training data. However, this approach has a significant weakness when implemented in real world situations because Otsu's segmentation is very sensitive to changes in brightness and lighting. To overcome said limitation, this project shifts to a lightweight supervised learning method that combines SLIC Superpixel and Random Forest, which is a more adaptive and robust approach when faced with environmental noises.

To overcome said limitation, modern researchers have shifted toward Supervised Learning Approaches, which utilize labeled training data to adaptively recognize

complex road structures under dynamic outdoor conditions. Within the supervised domain, while deep learning frameworks (like CNN or YOLO) offer high pixel-level accuracy, they demand heavy computational resources and high-end hardware, making them less ideal for real-time deployment on edge devices (Ma et al., 2022). Therefore, this project adopts a lightweight supervised machine learning framework as an optimal middle ground combining efficient regional pre-processing with a fast classifier to maintain high accuracy without high computational overhead.

b. SLIC Superpixel Segmentation

Pixel-based approaches often suffer from slow performance and are prone to generating false detections. This problem occurs because such systems are easily misled by disturbances such as shadows, grass, or other objects inadvertently captured by the camera (Safyari et al., 2024). Simple Linear Iterative Clustering (SLIC) superpixels offer a solution, as they can segment an image into hundreds of homogeneous regions based on color and spatial proximity (Achanta et al., 2012). This process isolates objects like roadside grass, vehicles, or houses into distinct groups, preventing them from interfering with the identification of pothole areas.

In brief, the asphalt image undergoes camera processing, involving sub-pixel estimation and angle adjustment all before being segmented by SLIC. Once the road surface is partitioned, extraneous objects such as grass or vehicles are isolated into separate groups, ensuring they do not distort the pothole area. The system can then adaptively detect potholes by applying thresholding to each individual region (Ma et al., 2022).

c. Random Forest Classification Approach

Random Forest is an ensemble-based supervised learning algorithm that operates by constructing a collection of decision trees randomly during the training process (Breiman, 2001; Mienye & Sun, 2022). When combined with Simple Linear Iterative Clustering (SLIC) superpixels, Random Forest becomes significantly more efficient, instead of processing millions of raw pixels individually, it classifies nine-dimensional feature vectors derived from hundreds of homogeneous regions previously grouped by SLIC (Csillik, 2017). These nine features encompass color aspects derived from the Blue-Green-Red (BGR) color space, color-space transformation features from the Hue-Saturation-Value (HSV) representation, edge and texture information captured through Gradient Magnitude (Sobel operator), morphological texture features extracted via a Blackhat transform, and local texture patterns encoded by Local Binary Pattern (LBP) descriptors.

The random forest algorithm is inherently stochastic in its sample selection, demonstrating high resilience against residual noise or outliers that escaped isolation during the segmentation phase, such as faint shadow distortions or uneven asphalt texture details. By employing majority voting across nine predefined

features, the random forest is able to identify pothole characteristics with greater discriminative power. The synergy between the efficient regional partitioning provided by SLIC and the rapid execution of majority voting in the random forest results in a pipeline that is robust against environmental interference while maintaining low computational overhead.

d. Gradient Boosting Methods

Gradient boosting constructs an ensemble of shallow decision trees sequentially, with each tree correcting the residual error of its predecessor (Friedman, 2001). XGBoost (Chen and Guestrin, 2016), LightGBM (Ke et al., 2017), and CatBoost (Prokhorenkova et al., 2018) are highly optimised implementations that have achieved state-of-the-art results on structured tabular data. Their key advantages over Random Forest include superior handling of complex feature interactions, built-in regularisation to prevent overfitting, and efficient histogram-based tree construction that enables fast training on large datasets. Ma et al. (2022) identify gradient boosting as an underexplored yet promising approach for feature-based road-surface segmentation, motivating the second phase of this study.

III. Methodology

The overall research workflow is illustrated in Fig. 1. The study begins with road-surface images collected from the ARA 7.0 dataset. During preprocessing, the input images undergo illumination normalization, Contrast Limited Adaptive Histogram Equalization (CLAHE), and bilateral filtering to improve image quality while preserving edge information. A road mask extraction step is then performed to remove irrelevant regions such as the sky and vegetation, ensuring that subsequent processing focuses only on the drivable road surface. After preprocessing, the workflow is divided into two independent computer vision pipelines for comparative evaluation.

Pipeline 1 (Random Forest + SLIC) adopts a SLIC-based superpixel segmentation approach, where approximately 250 superpixels are generated for each image. A 9-dimensional feature vector consisting of BGR color channels, HSV color space, gradient information, Blackhat morphology, and Local Binary Pattern (LBP) is extracted from each superpixel and classified using a Random Forest classifier with 200 decision trees and balanced class weighting. In contrast, Pipeline 2 (Boosting) performs pixel-level classification by extracting a 36-dimensional feature vector that combines color, texture, gradient information, local statistical features, spatial priors, and scene context. The extracted features are subsequently classified using gradient boosting algorithms, including LightGBM, XGBoost, and CatBoost, together with RDD2022 hard-negative samples to improve robustness against challenging background regions. Finally, both pipelines apply postprocessing to refine the predicted segmentation masks and generate the final binary pothole mask. Their performance is evaluated using Accuracy, Precision, Recall, F1-score, Intersection over Union (IoU), Mean IoU (mIoU), and the confusion matrix.

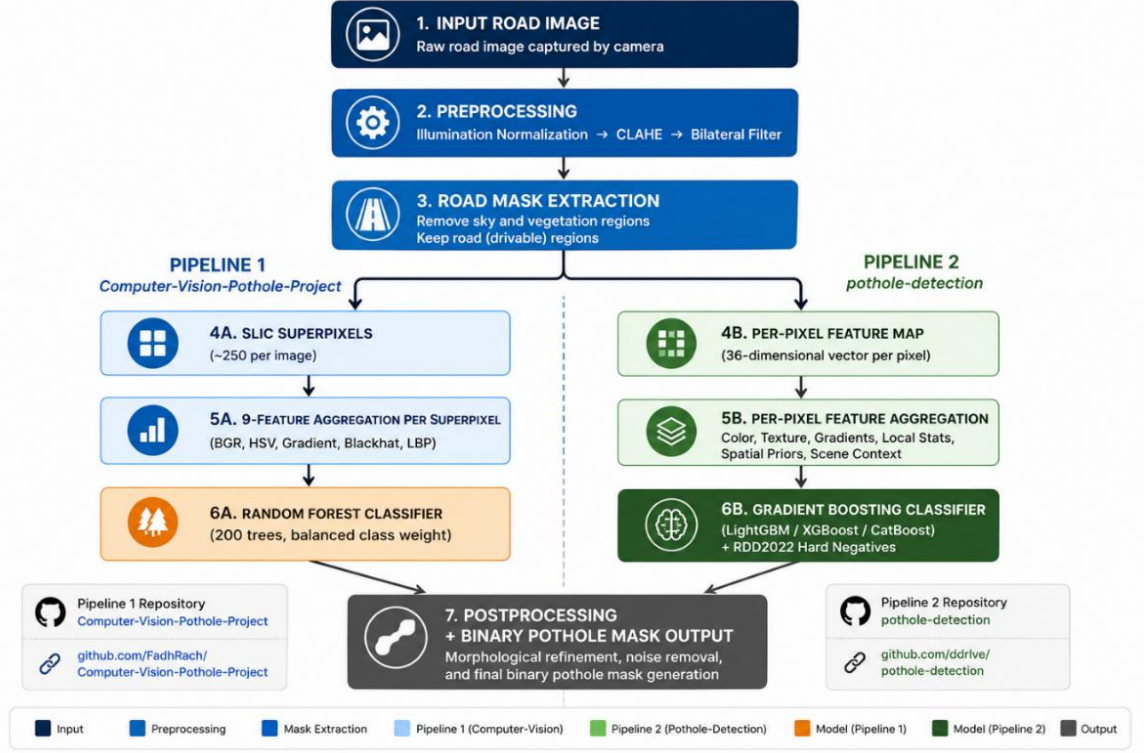


Fig. 1. Overall Methodology Pipeline for Both Pipelines

a. Dataset

This study utilizes two datasets, as summarized in Table 1, including the RDD2022 India subset used for hard-negative augmentation (Arya et al., 2022):

Dataset	Images	Split	Characteristics
ARA 7.0	793 images (498 train, 498 train mask + 295 test)	80:20 train-val (398/100)	Primary pothole segmentation (both pipelines)
RDD2022 India	700 images (hard negatives)	Training only	Hard-negative augmentation (Pipeline 2 only)

Table 1. Summary of the Dataset Used

To address the inherent class imbalance between pothole and non-pothole pixels where background pixels far outnumber pothole pixels, two mitigation strategies were employed. For the Random Forest pipeline, the `class_weight='balanced'` parameter was set, causing the algorithm to automatically adjust sample weights inversely proportional to class frequencies during training. For the LightGBM pipeline, curated hard-negative samples from the RDD2022 India subset (630,000 non-pothole pixels) were incorporated alongside pothole-positive patches, ensuring the classifier learned to distinguish genuine potholes from visually similar road distress patterns such as cracks and patches

b. Preprocessing Pipeline

All input images pass through the same three-stage preprocessing sequence before feature extraction in both Pipelines.

1) Illumination Normalization

Raw road images frequently suffer from uneven lighting caused by shadows, glare, or overexposure. A locally normalized intensity image is produced by dividing the grayscale channel by a Gaussian-blurred version of itself (sigma equal to 101 in Pipeline 1, sigma equal to 25 in Pipeline 2). This suppresses large-scale illumination gradients while preserving local texture differences, which are the primary cue for distinguishing potholes from intact road surface.

2) CLAHE (Contrast Limited Adaptive Histogram Equalization)

CLAHE is applied to the grayscale channel with a tile grid of 8 by 8 pixels and a clip limit of 2.5. This step enhances subtle pothole edges and makes darkened road regions more salient, improving both unsupervised thresholding and supervised classification.

3) Bilateral Filtering (Pipeline 1 only)

In Pipeline 1, a bilateral filter with diameter d equal to 9 pixels, σ_{color} equal to 50, and σ_{space} equal to 50 is applied to smooth noise while preserving sharp pothole edges. Unlike Gaussian blur, bilateral filtering is edge-aware: it averages pixels based on both spatial proximity and intensity similarity.

c. Model Architecture Pipeline 1 (Random Forest + SLIC)

Pipeline 1 implements three progressively more capable models. All operate on images downscaled to 0.5 working scale and share the same preprocessing and postprocessing pipeline.

1) Baseline: Adaptive Gaussian Thresholding

The baseline model is fully unsupervised and requires no training. After preprocessing and road-region extraction, an Adaptive Gaussian Threshold is applied to the L channel with a block size of 35 pixels and a constant C equal to 8. Pixels significantly darker than their local neighbourhood are flagged as pothole candidates, and small connected components with area below 1,000 square pixels are removed by morphological filtering. This method serves as the lower-bound reference for all other models.

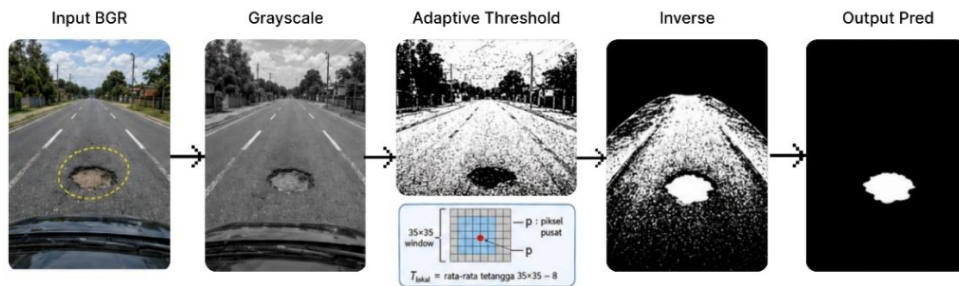


Fig. 2. Baseline: Adaptive Thresholding

2) Advanced Model: K-Means Pixel Clustering

The advanced model replaces the fixed threshold with K-Means clustering (K=3) applied to the preprocessed grayscale intensity values of road pixels. The darkest cluster center is assumed to correspond to pothole regions. The model operates at 0.5x working scale for efficiency, with K-Means initialized via k-means. It is more adaptive than Adaptive Thresholding but still requires no labelled data.

3) Main Supervised Model: Random Forest + SLIC Superpixels

Each image is segmented into approximately 250 SLIC superpixels with compactness equal to 10.0 and sigma equal to 1.0. Per-superpixel feature vectors are constructed by mean-pooling nine pixel-level features across each superpixel region, as listed in Table 2. A Random Forest with 200 decision trees, unbounded tree depth, a minimum of 3 samples per leaf, and balanced class weighting is trained on the 498 training images. Superpixels with more than 40% of their pixels overlapping the ground-truth pothole mask are labelled as pothole. At inference, each superpixel receives a binary label and the label map is upsampled back to original resolution using morphological close and open operations with kernel size 5.

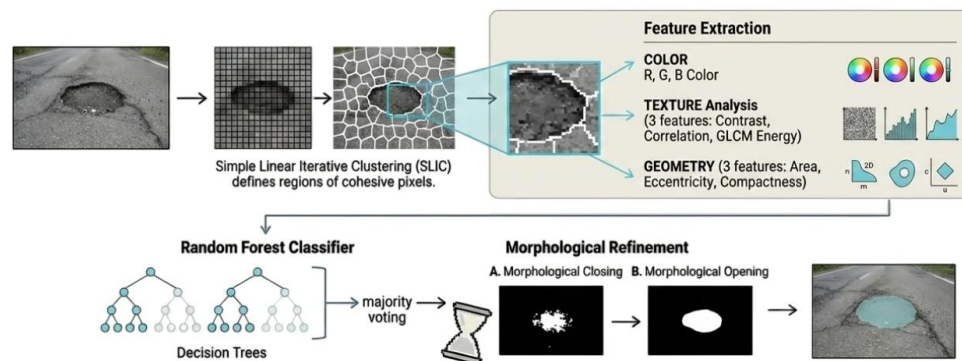


Fig. 3. Random Forest + SLIC Superpixel

d. Model Architecture Pipeline 2 (Gradient Boosting)

Following the course lecturer's recommendation to explore gradient-boosting methods, a second pipeline was developed with a completely redesigned architecture. Images are resized to a 320 by 320 pixel working resolution, and a 36-dimensional feature vector is computed for every pixel. The full list of 36 features covers nine groups: RGB colour, HSV colour, CIE L*a*b* colour, intensity (grayscale, CLAHE, illumination-normalized), edges and gradients (Sobel magnitude, Sobel angle, Laplacian), local statistics (mean and standard deviation at 7 px and 15 px radii), texture (LBP, Blackhat transforms at 15, 31, and 61 pixel scales, and Gabor filters at four orientations), spatial priors (normalized x/y position, bottom prior, center-x distance), and scene context (road mask, wet-like, shadow-like, dark-edge, specular-like indicators).

Three gradient-boosting classifiers were trained and compared: XGBoost, CatBoost, and LightGBM. Training was performed on a pixel-level dataset combining ARA 7.0 training images and 700 hard-negative images from RDD2022 India. A probability threshold search (values 0.65, 0.70, 0.80) combined with morphological postprocessing was used to find the optimal operating point. LightGBM was selected as the primary model due to its best validation performance. The Streamlit application for Pipeline 2 supports two inference modes: Real-time (single fast LightGBM model) and Accuracy (full model artifact).

Feature Group	Features Included	Pipeline 1 (RF + SLIC, 9f)	Pipeline 2 (Boosting, 36f)
Color (RGB/BGR)	R, G, B channels	Yes	Yes
Color (HSV)	Hue, Saturation, Value	Yes	Yes
Color (LAB)	L*, a*, b* channels	No	Yes
Intensity	Grayscale, CLAHE, illumination-normalized	Partial	Yes
Edges / Gradients	Gradient magnitude, Sobel angle, Laplacian	Gradient only	Yes
Local Statistics	Mean and Std at 7 px and 15 px window	No	Yes
Texture	LBP, Blackhat (3 scales), Gabor (4 filters)	LBP + 1 Blackhat	Yes
Spatial Priors	x/y normalized, bottom prior, center-x	No	Yes
Scene Context	Road mask, wet-like, shadow-like, dark-edge, specular	No	Yes
Total Features		9	36

Table 2. Feature Sets for Both Pipeline

e. Evaluation Metrics

All models are evaluated at the pixel level using five standard segmentation metrics:

1. Mean Intersection over Union (mIoU):

The primary metric. Computed as the average of IoU for the pothole class and the background class. A higher mIoU indicates better segmentation of both classes. IoU per class equals:

$$mIoU = \frac{1}{C} \sum_c^i IoU$$

2. Pothole IoU:

IoU computed for the pothole class only, providing a focused view of pothole detection quality independent of background accuracy.

$$IoU = \frac{TP}{(TP + FP + FN)}$$

3. Dice Coefficient (F1-Score):

Measures similarity rate between prediction result and real data to evaluate the number of correctly identified potholes.

$$F1 = \frac{2TP}{2TP + FP + FN}$$

Where:

TP = True Positive, FP = False Positive, FN = False Negative

4. Pixel Accuracy:

Fraction of all pixels correctly classified. High pixel accuracy can be misleading when one class dominates, so it is reported alongside mIoU and Dice. Calculates pixel percentage of correctly classified pothole into 2 categories (1: pothole, 0: not pothole).

5. Precision and Recall:

Precision measures the fraction of predicted pothole pixels that are true potholes. Recall measures the fraction of actual pothole pixels that are detected.

IV. Implementation and Result

a. System Implementation

Both pipelines were implemented in Python 3.11 using the following primary libraries: OpenCV 4.8 for image I/O and classical computer-vision operations, scikit-image 0.21 for SLIC superpixel generation and LBP texture extraction, scikit-learn 1.3 for Random Forest and preprocessing utilities, XGBoost 2.0, LightGBM 4.1, and CatBoost 1.2 for gradient-boosting classifiers in Pipeline 2, and Streamlit 1.29 for the interactive web interface. Training was performed on standard laptop CPUs without GPU acceleration.

Pipeline 1 (Random Forest + SLIC) provides a Streamlit application supporting all three models: Adaptive Threshold, K-Means, and Random Forest. Users can upload a road image, select the model from a sidebar, view the overlay of detected potholes, and optionally upload a ground-truth mask for metric evaluation. Pipeline 2 (Boosting) provides an enhanced Streamlit application with two inference modes, a probability heatmap alongside the binary mask, and fully configurable

postprocessing parameters including probability threshold (default 0.80), minimum component area (default 1200 px), morphological kernel sizes, and a maximum predicted-area ratio guard (default 0.18) that prevents over-labelling.

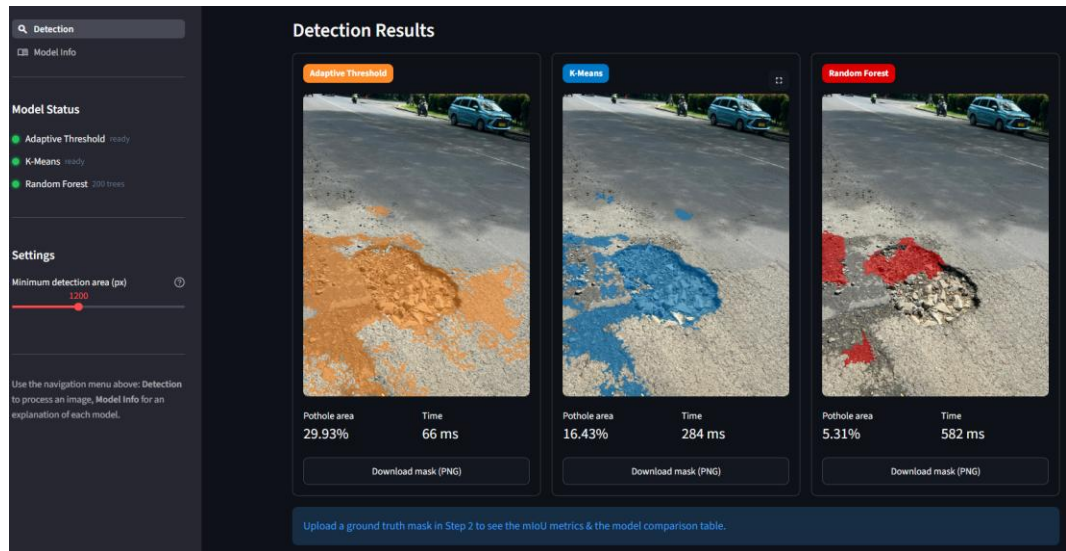


Fig. 4. Pipeline 1 Application Interface (Random Forest + SLIC)

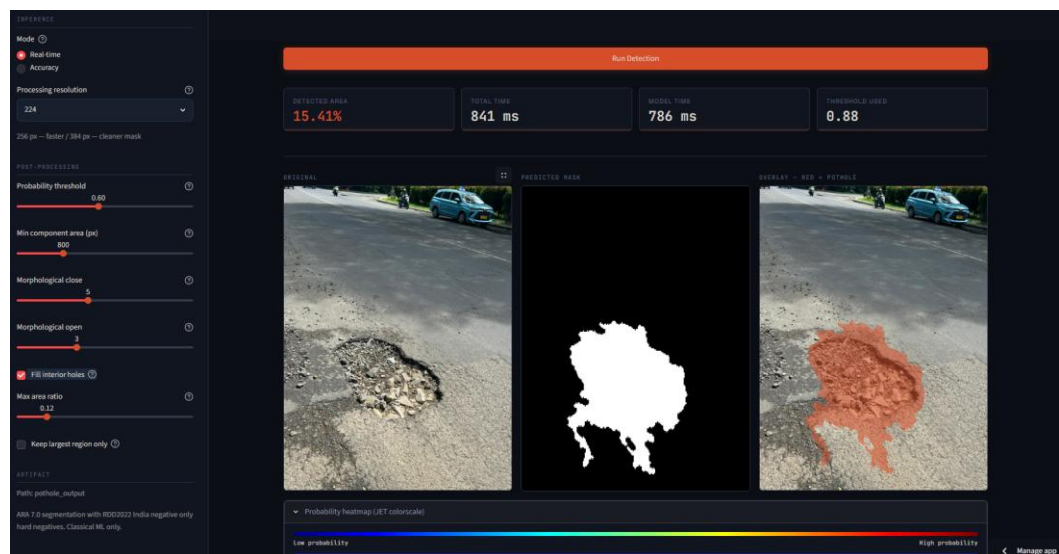


Fig. 5. Pipeline 2 Application Interface (LightGBM + 36 Features)

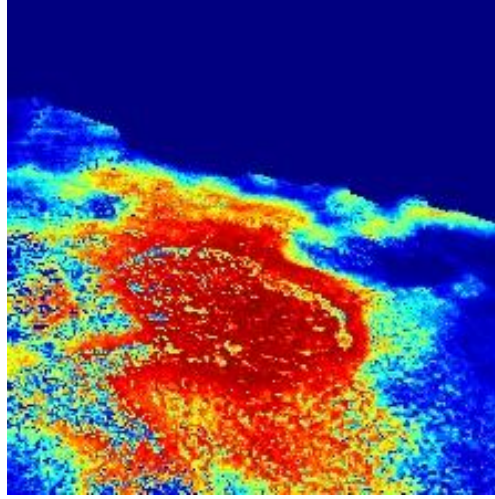


Fig. 6. Pipeline 2 Probability heatmap (LightGBM + 36 Features)

b. Evaluation Result

Table 3 presents the evaluation metrics on the 100 validation images. Adaptive Threshold and K-Means are reported with mIoU only, as they are unsupervised models without systematic per-image metric logging for all measures. Random Forest (Pipeline 1) and LightGBM (Pipeline 2) are fully evaluated.

Model	IoU Pothole	mIoU	Dice / F1	Pixel Acc.	Precision	Recall
Adaptive Threshold (Baseline)	0.1827	0.403	0.2795	0.6676	0.2386	0.565
K-Means Clustering (Advanced)	0.1546	0.444	0.2421	0.7503	0.2448	0.365
Random Forest + SLIC (9 features)	0.183	0.518	0.277	0.863	0.449	0.238
LightGBM + Pixel-Level (36 features)	0.320	0.592	0.451	0.874	0.452	0.619

Table 3. Evaluation Results on 100 Validation Images (ARA 7.0 Dataset)

The unsupervised baselines achieved moderate mIoU values of 0.403 for Adaptive Threshold and 0.444 for K-Means. Both methods are sensitive to illumination changes and frequently confuse shadows with pothole regions. The Random Forest model (Pipeline 1) achieves mIoU of 0.518, pixel accuracy of 86.3%, and Dice of 0.277 on the validation set. While mIoU and pixel accuracy are relatively high, the low Dice and pothole IoU indicate that the model misses entirely in 23 out of 100

validation images (IoU equal to zero for those images), reflecting conservative predictions under challenging conditions.

The LightGBM model in Pipeline 2, trained with 36 pixel-level features and RDD2022 hard negatives, improves all metrics simultaneously: pothole IoU rises from 0.183 to 0.320, mIoU from 0.518 to 0.592, Dice from 0.277 to 0.451, and pixel accuracy from 86.3% to 87.4%. This confirms that the richer feature set, the pixel-level resolution, and hard-negative augmentation collectively drive the accuracy improvement.

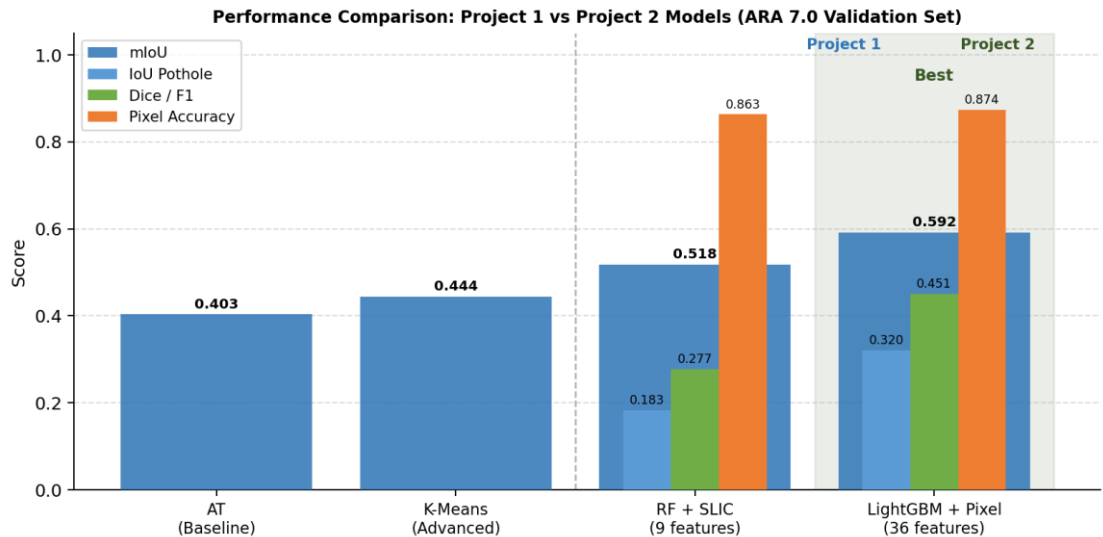


Fig. 7. Performance Comparison Across All Models (ARA 7.0 Validation Set)

c. Gradient Boosting Algorithm Comparison (Pipeline 2)

Within Pipeline 2, three gradient-boosting algorithms were compared on a held-out sample of 30 training images using the same 36-feature pixel-level pipeline. Results are presented in Table 4.

Model	Macro F1	Pothole F1	Precision	Recall
LightGBM	0.827	0.822	0.777	0.872
Soft Ensemble (all 3 models)	0.822	0.819	0.767	0.879
CatBoost	0.817	0.815	0.758	0.882
XGBoost	0.816	0.814	0.758	0.880

Table 4. Boosting Algorithm Comparison within Pipeline 2 (30-Image Sample)

LightGBM achieves the highest macro F1 of 0.827 and pothole F1 of 0.822, making it the recommended model for deployment. The soft ensemble, which averages predicted probabilities from all three models, performs slightly below LightGBM alone while requiring approximately three times the inference time. XGBoost and CatBoost are within 1 to 2 percentage points of LightGBM on all

metrics, confirming that the feature engineering contributes more to performance than the specific boosting algorithm chosen.

d. Feature Importance Analysis

LightGBM's built-in feature importance metric (total gain) for Pipeline 2 reveals the most informative features, as shown in Figure 8. The vertical pixel position (y_norm) is the single most important feature, reflecting the fact that potholes tend to appear in the lower portion of road images captured from a forward-facing camera. The illumination-normalized intensity and the large-scale blackhat transform at 61 pixels form the next tier, capturing the characteristic dark appearance of potholes relative to their surroundings. Scene context indicators such as wet-like (rank 4) and shadow-like (rank 9) were also highly ranked, showing that the model learns to adjust predictions based on local surface conditions.

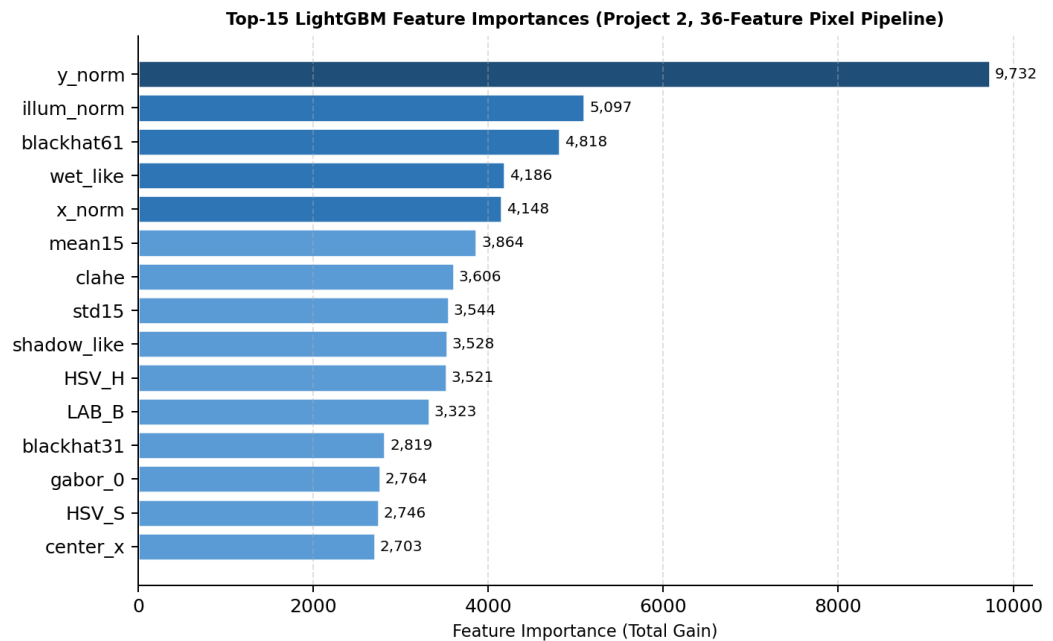


Fig. 8. Top-15 LightGBM Feature Importances (Pipeline 2, Total Gain)

e. Inference Speed Comparison

The two pipeline architectures exhibit a fundamental accuracy versus speed trade-off, as summarized in Table 5. Pipeline 1 reduces each image to approximately 250 superpixel feature vectors before classification, enabling fast inference. Pipeline 2 computes 36 features for every pixel at 320 by 320 resolution (102,400 pixels per image) and therefore requires substantially more computation per image.

Configuration	Classification Unit	Predictions per Image	Relative Speed
RF + SLIC (Pipeline 1)	Superpixel	approx. 250	Fast

LightGBM Pixel-Level, Real-time (Pipeline 2)	Pixel	102,400 (256 px)	Moderate
LightGBM Pixel-Level, Accuracy (Pipeline 2)	Pixel	102,400 (256 px)	Slower

Table 5. Qualitative Inference Speed Comparison

To mitigate the latency of Pipeline 2, the Streamlit application provides two inference modes. The Real-time mode uses a compressed single LightGBM model optimised for lower latency, while the Accuracy mode uses the full model artifact for the best segmentation quality. Both modes operate at a 256 by 256 pixel application resolution compared to the 320 by 320 pixel training resolution, further reducing inference time.

V. Discussion and Limitations

a. Unsupervised versus Supervised Models

The unsupervised baselines, Adaptive Threshold and K-Means, were consistently outperformed by the Random Forest model on the ARA 7.0 validation set. K-Means produced more false positives than Adaptive Threshold in well-lit conditions but was more robust to illumination variation. Neither method achieved adequate pothole IoU for practical deployment, thus confirming that labelled training data is necessary for reliable pothole segmentation on a heterogeneous dataset with diverse lighting and road surface conditions.

b. Pipeline 1 to Pipeline 2: From SLIC to Pixel-Level Boosting

The transition from Pipeline 1 (Random Forest + SLIC, 9 features) to Pipeline 2 (LightGBM + pixel-level, 36 features) improved all metrics simultaneously. The addition of LAB color space features, multi-scale local statistics, multi-scale blackhat morphological transforms, Gabor texture filters, spatial priors, and scene context indicators gave the model a richer per-pixel representation. The gradient-boosting classifier is able to exploit interactions among these 36 features more effectively than a Random Forest trained on only 9 features aggregated over superpixel regions.

Hard-negative augmentation from RDD2022 India further contributed to precision improvement. The Pipeline 1 Random Forest model had a low recall (0.238) because it classifies conservatively on the 9-feature SLIC representation, it correctly identifies background but misses many potholes. Pipeline 2, exposed to 630,000 additional road pixels without potholes during training, learned to better discriminate between potholes and visually similar road features, raising recall to 0.619 while maintaining precision at 0.452.

c. **Role of Feature Engineering in Pipeline 2**

Feature importance analysis confirms that scene context features, particularly wet-like (rank 4) and shadow-like (rank 9), were especially valuable in allowing the model to adjust predictions based on surface conditions that mimic pothole appearance. The dominance of `y_norm` as the top feature reflects the camera perspective: potholes are physically located on the road surface, which in a forward-facing camera image occupies the lower portion of the frame. Illumination normalization (rank 2) and large-scale blackhat transforms (rank 3) capture the fundamental dark appearance of potholes relative to surrounding road surface. These findings confirm that structured feature engineering, rather than the specific choice of boosting algorithm, is the primary driver of Pipeline 2's accuracy improvement.

d. **Limitations**

This experiment faced several limitations:

1. **Dataset size:**

The ARA 7.0 training set contains 498 images, which is relatively small for pixel-level supervised learning. A larger and more diverse dataset would likely improve generalization, especially for unusual pothole shapes and rare lighting conditions.

2. **No deep-learning comparison:**

This study deliberately restricts itself to classical machine learning. A benchmark comparison against lightweight segmentation networks would contextualize the performance gap to state-of-the-art methods.

3. **Inference speed not precisely measured:**

Due to hardware variations across team members' machines, inference times are reported qualitatively rather than with precise millisecond measurements.

4. **Unsupervised model metrics partially reported:**

Baseline and K-Means results are reported with mIoU only. Systematic per-image metric logging for all measures would strengthen the baseline comparison.

5. **Generalization to other road types:**

Both models were trained and evaluated exclusively on Indonesian road data. Performance on roads from other countries with different surface materials, damage patterns, and weather conditions has not been evaluated.

6. **Fixed working resolution:**

Pipeline 1 operates at 0.5x scale, Pipeline 2 at 320 by 320 pixels. Very high-resolution camera images may contain fine-grained details that are lost during rescaling.

e. **Ethical Considerations and Responsible AI**

The deployment of automated pothole detection systems raises important ethical considerations that must be addressed for responsible AI development. First, regarding algorithmic bias, the training datasets comprising images primarily from

Indonesian urban roads and the RDD2022 India dataset may not adequately represent rural road conditions, nighttime scenarios, or road surfaces from diverse geographic regions. This geographic and environmental bias could result in reduced detection accuracy for underrepresented conditions, potentially leading to unequal infrastructure maintenance across different communities. To mitigate this, future iterations should incorporate more diverse datasets spanning varied climates, road materials, and lighting conditions.

Second, concerning transparency and explainability, the LightGBM-based pipeline provides inherent interpretability advantages over deep learning approaches, as feature importance scores (Figure 8) reveal which visual cues such as normalized luminance and blackhat texture drive classification decisions. This transparency supports auditability and allows domain experts to validate model reasoning. Third, with respect to privacy, the proposed system processes road imagery that may incidentally capture pedestrians or vehicle license plates. Any deployment at scale must implement appropriate anonymization procedures and comply with applicable data protection regulations. Finally, while this system is designed for beneficial infrastructure monitoring purposes, stakeholders must ensure that automated detection outputs are reviewed by qualified engineers before triggering maintenance decisions, preventing potential misclassification from causing resource misallocation or public safety risks

VI. Conclusion and Future Work

a. Conclusion

This project presented a two-phase study of classical machine-learning approaches for pothole road-surface segmentation. Pipeline 1 (Random Forest + SLIC) progressed from two unsupervised baselines to a supervised Random Forest classifier trained on SLIC superpixels with 9 hand-crafted features. Following the course lecturer's recommendation, Pipeline 2 (Boosting) implemented a redesigned pixel-level pipeline using gradient-boosting classifiers with 36 engineered features and hard-negative augmentation from RDD2022 India. The following conclusions are drawn:

1. Supervised models substantially outperform unsupervised baselines on the heterogeneous ARA 7.0 dataset. Adaptive Threshold and K-Means are insufficient for practical deployment.
2. The Random Forest + SLIC model (Pipeline 1) achieves mIoU of 0.518 and pixel accuracy of 86.3%, but has a low Dice coefficient of 0.277 due to conservative predictions that miss potholes in difficult conditions.
3. The LightGBM pixel-level pipeline (Pipeline 2) improves all metrics simultaneously: mIoU 0.592, Dice 0.451, and pixel accuracy 87.4%, representing a meaningful improvement over Pipeline 1.
4. LightGBM is the best individual model among the three gradient-boosting algorithms evaluated in Pipeline 2, achieving macro F1 of 0.827 on a held-out sample.

5. The accuracy gain from Pipeline 1 to Pipeline 2 comes at the cost of higher inference latency due to pixel-level feature computation, partially mitigated by the Real-time inference mode in the Pipeline 2 web application.
6. For deployment where speed is the primary constraint, Pipeline 1 with Random Forest is recommended. Where accuracy is the primary requirement, Pipeline 2 with LightGBM is the recommended choice.

b. Future Work

Several modifications and improvements could extend this work in future research:

1. Ensemble of classical and deep-learning models: A lightweight CNN feature extractor combined with a boosting classifier could combine the spatial reasoning of deep models with the interpretability and training efficiency of classical methods.
2. Temporal consistency for video: For video inputs, enforcing consistency across frames using optical flow or temporal smoothing would reduce frame-to-frame flickering in predictions.
3. Pothole severity estimation: Beyond binary segmentation, estimating pothole depth or area would enable prioritization of road-repair schedules based on severity.
4. Edge deployment: Quantization and tree-model compression techniques such as TreeLite could enable deployment on embedded hardware for real-time on-vehicle use.
5. Dataset expansion: Collecting and annotating additional road images from different provinces and weather conditions would improve model generalization across Indonesia.

VII. REFERENCES

- Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Süsstrunk, S. (2012). SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11), 2274–2282.
- Arya, D., Maeda, H., Ghosh, S. K., Toshniwal, D., Mraz, A., Kashiyama, T., & Sekimoto, Y. (2022). RDD2022: A multi-national image dataset for automatic road damage detection. *arXiv:2209.08538*.
- Badan Pusat Statistik. (2023). Statistik Transportasi Darat 2022. Jakarta: BPS-Statistics Indonesia. <https://www.bps.go.id>
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.
- Buza, E., Omanovic, S., & Huseinovic, A. (2013). Pothole detection with image processing and spectral clustering. In *Proceedings of the 2nd International Conference on Information Technology and Computer Networks* (pp. 48–53).
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794).
- Csillik, O. (2017). Fast segmentation and classification of very high resolution remote sensing data using SLIC superpixels. *Remote Sensing*, 9(3), 243.

- Eriksson, J., Girod, L., Hull, B., Newton, R., Madden, S., & Balakrishnan, H. (2008). The pothole patrol: Using a mobile sensor network for road surface monitoring. In *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services* (pp. 29–39).
- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5), 1189–1232.
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 30.
- Ma, N., Fan, J., Wang, W., Wu, J., Jiang, Y., Xie, L., & Fan, R. (2022). Computer vision for road imaging and pothole detection: A state-of-the-art review of systems and algorithms. *Transportation Safety and Environment*, 4(4), tdac026. <https://doi.org/10.1093/tse/tdac026>
- Mienye, I. D., & Sun, Y. (2022). A survey of ensemble learning: Concepts, algorithms, applications, and prospects. *IEEE Access*, 10, 99129–99149. <https://doi.org/10.1109/ACCESS.2022.3207287>
- Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: Unbiased boosting with categorical features. *Advances in Neural Information Processing Systems*, 31.
- Safyari, Y., Mahdianpari, M., & Shiri, H. (2024). A review of vision-based pothole detection methods using computer vision and machine learning. *Sensors*, 24(17), 5652. <https://doi.org/10.3390/s24175652>

VIII. APPENDIX

A. Team Contribution

Table A1 details each team member's specific contributions to this project. All team members participated in weekly online meetings, peer code review, and jointly prepared the project demonstration video.

Name	NIM	Contributions
Dian Rakhmawati Lestari	2802539085	Model training, model experimentation, final report, presentation, deployment, app testing.
Fadhlan Nur Rachman	2802491690	Model training, model experimentation, deployment, presentation, app testing.
Matthew Ken Susanto	2802407736	Model experimentation, final report, presentation, app testing.
Nasauramecca Nour Haqqanshah Shodiqin	2802541921	Model experimentation, final report, presentation, demo video.
Nicholas	2802424326	Model experimentation, final report, presentation, app testing.

Table A1. Team Contribution

B. Repository and Resources

The source code, trained model artifacts, and live applications for both pipelines are accessible through the links listed in Table A2. Version control was managed using Git with separate repositories for each pipeline.

Resource	Pipeline	Link / URL
Source Code (GitHub)	Pipeline 1 (RF + SLIC)	https://github.com/FadhRach/Computer-Vision-Pothole-Project
Source Code (GitHub)	Pipeline 2 (Boosting + 36f)	https://github.com/ddrlve/pothole-detection
Live Web App (Streamlit)	Pipeline 1	https://computer-vision-pothole-project.streamlit.app/
Live Web App (Streamlit)	Pipeline 2	https://pothole-detection-cv.streamlit.app
Live Web App (HuggingFace)	Pipeline 2	https://huggingface.co/spaces/ddrlvee/pothole-detection
Hard-Negative Dataset RDD2022 India	Pipeline 2 (training only)	https://github.com/sekilab/RoadDamageDetector
ARA 7.0 Dataset	Pipeline 1 & 2 (train-validation)	ARA 7.0 Dataset
Demo Video	YouTube	https://youtu.be/JAk7gb7jyc?feature=shared
PPT Canva	Presentation	https://canva.link/nfw5gd1jus2nqk8
Drive Backup	Drive	Drive Backup

Table A2. Project Links and Resources

C. Key Code Snippets

a. SLIC Feature Extraction (Pipeline 1, 9 Features per Superpixel)

```
from skimage.segmentation import slic
import numpy as np

segments = slic(img, n_segments=250, compactness=10.0, sigma=1.0, start_label=0)

feature_rows = []
for sp_id in np.unique(segments):
    mask = segments == sp_id
    bgr_mean = img[mask].mean(axis=0) # B, G, R
    hsv_mean = hsv_img[mask].mean(axis=0) # H, S, V
    grad_mean = grad[mask].mean() # gradient magnitude
    bhat_mean = blackhat[mask].mean() # blackhat
    lbp_mean = lbp[mask].mean() # LBP
    feature_rows.append([*bgr_mean, *hsv_mean, grad_mean, bhat_mean, lbp_mean])
```

b. LightGBM Training with Balanced Weights (Pipeline 2, 36 Features)

```
import cv2, numpy as np

prob_map = model.predict_proba(X_pixels)[: , 1].reshape(H, W)
pred_mask = (prob_map >= threshold).astype(np.uint8)

k = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (close_k, close_k))
pred_mask = cv2.morphologyEx(pred_mask, cv2.MORPH_CLOSE, k)

n, labels, stats, _ = cv2.connectedComponentsWithStats(pred_mask)
for lbl in range(1, n):
    if stats[lbl, cv2.CC_STAT_AREA] < min_area:
        pred_mask[labels == lbl] = 0

if pred_mask.mean() > max_area_ratio:
```

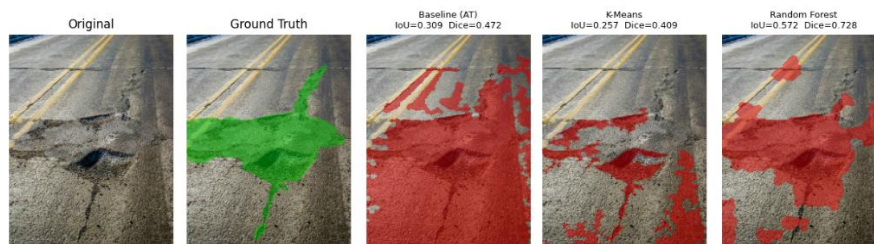
c. Postprocessing (Probability Threshold and Morphological Operations)

```
import lightgbm as lgb

scale_pos = (y_train == 0).sum() / (y_train == 1).sum() # approx. 6.5

clf = lgb.LGBMClassifier(
    n_estimators=500, learning_rate=0.05, num_leaves=63,
    scale_pos_weight=scale_pos, subsample=0.8,
    colsample_bytree=0.8, random_state=42, n_jobs=-1
)
clf.fit(
    X_train, y_train,
    eval_set=[(X_val, y_val)],
    callbacks=[lgb.early_stopping(50), lgb.log_evaluation(100)]
)
```

Comparison of models' detection capabilities (Pipeline 1):



Model Pipeline 1:

