

Practical implementation of pothole detection and dimension estimation using YOLOv8 and Mask R-CNN with a Flutter-based mobile application

Suci Dwijayanti*, Aditya Erlangga Wibowo, Bhakti Yudho Suprpto, Rendyansyah Rendyansyah, Hera Hikmarika, Robi Prasetyo

Department of Electrical Engineering, Faculty of Engineering, Universitas Sriwijaya, Indonesia

Abstract

Road surface deterioration, particularly pothole formation, poses substantial threats to traffic safety and driving comfort, highlighting the need for timely and objective road condition assessment. However, past studies have predominantly focused on pothole detection without estimating pothole dimensions, while conventional inspection methods remain manual, inefficient, and subjective. To address these limitations, a deep-learning-based framework is proposed herein for joint pothole detection and dimension estimation using monocular road images. The proposed framework integrates the You Only Look Once (YOLO)v8 and Mask Region-Based Convolutional Neural Network (Mask R-CNN) models to detect potholes and estimate their geometric dimensions from road surface images. Image data were acquired using a GoPro Hero 8 camera across multiple road segments in Sumatra Selatan, Indonesia. The models were trained on Google Colaboratory using the YOLOv8x-seg architecture and subsequently integrated into a Flutter-based mobile application for practical deployment, while Leaflet JS was employed to interactively visualize pothole locations. Experimental results demonstrated detection confidence levels exceeding 92.22%, with stable performance observed on both local computing platforms and mobile devices. Application-level evaluations further confirmed robust cross-device functionality and yielded a usability score of 4.8875. These results indicate that the proposed approach provides an accurate and reliable solution for automated road condition monitoring, supporting more objective and data-driven road maintenance planning.

Keywords:

Deep learning;
Intelligent transportation systems;
Mobile-based inspection;
Pothole detection;
Road damage assessment

Article History:

Received: March 10, 2026
Revised: June 24, 2026
Accepted: July 20, 2026
Published: October 15, 2026

Corresponding Author:

Suci Dwijayanti
Department of Electrical Engineering, Faculty of Engineering, Universitas Sriwijaya, Indonesia
Email: sucidwijayanti@ft.unsri.ac.id

This is an open-access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license



INTRODUCTION

Road surface damage, particularly potholes, causes driving discomfort and substantially increases the risk of traffic accidents. Therefore, the rapid and accurate identification of road conditions is essential for enabling timely maintenance and repair. Based on 2022 data reported by the Central Statistics Agency (BPS) of Sumatra Selatan, the province ranks among the regions with the highest levels of road damage across national, provincial, and

district road networks in Indonesia [1]. Currently, road condition assessment in Indonesia is conducted by the Indonesian Department of Public Works and Housing (PUPR) through public reporting systems, enabling residents to submit reports of road damage. However, this approach suffers from several limitations, including inconsistent data recording, incomplete damage descriptions, and a lack of real-time information. These challenges highlight the need for an automated and objective system capable of

identifying road conditions accurately and efficiently.

Road damage detection has been studied extensively using deep learning approaches, including You Only Look Once (YOLO) variants, such as YOLOv3 [2][3], YOLOv5 [4], YOLOv7 [5][6], YOLOv11 [7], YOLOX [8], YOLOv8 [9], as well as SSD-MobileNet [10] and convolutional neural networks (CNNs) [11]. These methods are capable of automated pothole detection and classification. However, despite this progress, many approaches still suffer from limited detection accuracy and inadequate real-time performance.

In a previous study [12], deep learning and computer vision techniques were employed to identify road defects, including potholes, crocodile cracks, and longitudinal cracks, using citizen-submitted smartphone reports for spatial mapping. Meanwhile, [13] combined wavelet methods with a convolutional neural network (CNN) to improve the efficiency and accuracy of crack detection in road images. Another study [14] proposed an improved CNN architecture derived from VGG16, while a CNN-based approach was also used to classify pothole and non-pothole images [15]. Android-based applications that collected pothole and non-pothole datasets using accelerometer, gyroscope, and global positioning system sensors, followed by prediction using random forest and k-nearest neighbor algorithms, have also been reported [16][17]. Furthermore, an Android-based pothole detection application was proposed by Rahman et al. [18], employing a CNN-based SSD-MobileNet model to generate warning signals for avoidance assistance. However, these approaches have primarily focused on defect classification and have not addressed dimension estimation. Similarly, smartphone-based methods have been proposed to detect speed bumps, manholes, and potholes using camera images and acceleration data [19], while accelerometer-based classification using CNNs has also been reported [20]. Pothole detection has also been integrated into autonomous vehicle systems [21].

Different YOLO variants have been evaluated for real-time pothole detection [12, 22, 23]. However, these evaluation studies were limited to defect identification and did not incorporate geometric dimension estimation, thereby restricting their practical applicability for maintenance planning. To address this limitation, dimension estimation techniques have been explored in several studies. For instance, photogrammetric methods combined with a CNN-based RetinaNet model have been used for pothole detection and depth estimation [24], while other approaches have been used to estimate

pothole area using computer vision techniques like morphological closing operations [25].

Despite these efforts, most previous studies have focused on defect classification or localization, while accurate dimension estimation, which is an essential factor for repair prioritization and maintenance planning, remains insufficiently explored. Furthermore, many methods have been evaluated only in offline or desktop-based environments, with limited attention given to real-time operation on mobile platforms. Therefore, performance degradation under real-world conditions, including variations in illumination, road surface texture, and camera viewpoint, remains a significant challenge. Consequently, integrated real-time systems, which combine robust detection, reliable dimension estimation, and practical mobile deployment within a unified framework, remain scarce.

To address these limitations, a deep learning-based system is proposed herein for pothole detection and dimension estimation implemented in a mobile application called POTION AI. The proposed system integrates the YOLOv8 and Mask Region-Based Convolutional Neural Network (Mask R-CNN) models, combining the strengths of both approaches while automatically selecting the most reliable prediction for each inspection session. This integration enables accurate pothole detection and geometric dimension estimation. The dataset was collected using a GoPro Hero 8 camera across multiple road segments in Sumatra Selatan, Indonesia. Model training was performed on Google Colaboratory using the YOLOv8x-seg architecture, which comprises approximately 71 million parameters. A comparison of representative state-of-the-art methods is presented in Table 1.

For the practical implementation of maintenance actions and the reduction of accident risk, a pothole monitoring system must be capable of real-time defect detection and dimension estimation [26]. The proposed system is expected to assist relevant authorities by accelerating maintenance workflows and mitigating pothole-related traffic hazards via the provision of accurate and timely road condition information.

The main contributions of this study are as follows:

1. A unified framework for joint pothole detection and dimension estimation.
2. The integration of YOLOv8 and Mask R-CNN to improve detection and measurement performance.
3. A practical mobile-based implementation for road monitoring applications.

Table 1. State-of-the-art comparison of deep learning models for object detection

Method	Detection strategy	Segmentation capability	Decision strategy	Advantages	Limitation
Traditional image processing (edge detection, thresholding)	Hand-crafted features	No	Fixed thresholds	Simple implementation and low computational cost	Highly sensitive to illumination, shadows, and road texture variations
Mask R-CNN	Instance segmentation	Yes	Single segmentation model	Accurate boundary extraction and estimation	Slower inference and reduced real-time capability
YOLOv5	One-stage object detection	No	Single detector	Fast inference, lightweight architecture, and real-time application suitability	Less accurate boundary representation for irregular potholes compared with instance segmentation methods
YOLOv8	One-stage object detection	No	Single detector	Higher accuracy, faster inference, and improved robustness compared with earlier YOLO versions	Performance may decrease for complex pothole geometries or severe occlusions
Proposed POTION AI	Independent detection using YOLOv8 and Mask R-CNN	Yes	Adaptive model selection based on average confidence/score	Combines the strengths of both models while automatically selecting the most reliable prediction for each inspection session; improves robustness under varying road conditions	Requires the maintenance of two trained models and a slightly higher memory requirement

The remainder of this paper is organized as follows. The next section describes the proposed methodology, including data acquisition, model architecture, training procedures, and system integration within a Flutter-based application. This is followed by the presentation of the experimental results and performance evaluation, along with a discussion of their implications for real-world road condition monitoring. The paper concludes with a summary of the findings and directions for future research.

METHODOLOGY

Dataset Collection

This study employed primary data consisting of video footage captured via a GoPro Hero 8 camera mounted on a moving vehicle. Data acquisition was conducted at multiple locations across Sumatra Selatan under daylight conditions. The recorded videos were stored in MP4 format and subsequently converted into image frames in PNG format. Frame extraction was performed using Shotcut software by reducing the frame rate to 10 frames per second (fps). The resulting dataset contained diverse road-related objects targeted for the object detection stage, including potholes, vehicles, and surrounding road environments. A total of 2,069 images were acquired, followed by data

augmentation procedures involving adjustments to saturation and exposure, as well as image blurring. A representative sample of the dataset is presented in Figure 1. The dataset was divided into training, validation, and testing sets in a proportion of 75%, 15%, and 10%, respectively.

Training Model

Model training employed two principal algorithms: YOLOv8 and Mask R-CNN. YOLOv8 was trained using the YOLOv8x-seg architecture over 50, 100, and 200 epochs on Google Colaboratory with a Tesla T4 GPU to detect road potholes using the collected and augmented image dataset. Mask R-CNN was implemented using Detectron2, which supports instance segmentation, enabling object detection and pixel-level segmentation mask generation for individual instances.

The training configuration used a ResNet-50 backbone with a feature pyramid network. Training was evaluated at 700, 800, and 900 iterations. The dataset followed the COCO standard, with bounding box and pixel-level mask annotations for each object. Model parameters were optimized using stochastic gradient descent (SGD), with hyperparameters configured to ensure stable convergence and optimal performance.



Figure 1. Dataset sample

Table 2 and Table 3 summarize the principal hyperparameters used for training the Mask R-CNN and YOLOv8 models, respectively. These parameters include the number of training epochs or iterations, learning rate, batch size, optimizer configuration, and backbone architecture. The selected settings were determined based on preliminary experiments and widely adopted practices in deep learning-based object detection and instance segmentation to ensure stable convergence and reliable performance.

To ensure reproducibility, all experiments were conducted using fixed model configurations and standardized training procedures. The dataset underwent consistent preprocessing and augmentation prior to training, and identical training-validation splits were applied across all experiments. Model training was performed using publicly available frameworks, including Ultralytics YOLOv8 and Detectron2, enabling the proposed approach to be replicated under comparable computational conditions.

Dimension Estimation

For dimension estimation, the width and length of each pothole were first measured manually.

Table 2. Hyperparameter Tuning of Mask R-CNN

Parameter	Iterations		
	700	800	900
Backbone	ResNet-50	ResNet-50	ResNet-50
Batch size	16	16	16
Learning rate	0.0025	0.0025	0.0025
Momentum	0.9	0.9	0.9
Weight decay	0.0001	0.0001	0.0001
Optimizer	SGD	SGD	SGD

Table 3. Hyperparameter Tuning of YOLOv8.

Tuning	Hyperparameters				
	Batch	Momentum	Weight decay	Learning rate	Optimizer
1.	16	0.937	0.0005	0.001	SGD
2.	16	0.937	0.0005	$0.001 + \cos lr$	SGD
3.	16	0.937	0.0005	0.001	AdamW
4.	16	0.937	0.0005	$0.001 + \cos lr$	AdamW

These measurements were then compared with the dimensions of the corresponding detected bounding boxes expressed in pixel units. Subsequently, the pixel dimensions were compared with the actual pothole dimensions. Based on this comparison, average scale factors were derived and applied to convert dimensions from pixels to centimeters by multiplying the pixel width and height by the corresponding scale values. The calculations are expressed as follows:

$$width_{cm} = width_{px} \times avg \ width \ scale \quad (1)$$

$$height_{cm} = height_{px} \times avg \ height \ scale \quad (2)$$

where *avg width scale* and *avg height scale* are calculated as:

$$avg \ width \ scale = \frac{\sum_{i=1}^n \frac{Width_{pred_cm_i}}{Width_{true_i}}}{n} \quad (3)$$

$$avg \ height \ scale = \frac{\sum_{i=1}^n \frac{Height_{pred_cm_i}}{Height_{true_i}}}{n} \quad (4)$$

The values of $Width_{pred_cm_i}$ and $Height_{pred_cm_i}$ are obtained by converting the predicted pixel dimensions into centimeters, where *n* represents the total number of measured potholes. The pixel values are extracted from the bounding boxes of the detected potholes and calculated as follows:

$$Width_{pred_cm_i} = width_{pred_i} \times a \quad (5)$$

$$Height_{pred_cm_i} = height_{pred_i} \times a \quad (6)$$

where $a = 0.0264583333$ is the conversion factor from pixels to centimeters.

Application Design of POTION AI

The overall architecture of the POTION AI system consists of four core modules: data acquisition, deep learning inference, application logic, and visualization. Road surface images are acquired using a GoPro Hero 8 camera mounted on a moving vehicle and transmitted to the mobile application for subsequent processing. The deep learning inference module combines YOLOv8 for pothole detection with Mask R-CNN for instance segmentation and dimension estimation. The application logic layer handles data preprocessing, inference execution, and management of output results within the Flutter framework.

The visualization module presented detection results and estimated pothole dimensions on an interactive map implemented using Leaflet JS, with geospatial positioning derived from latitude and longitude coordinates. This modular design supports efficient real-time processing and ensures seamless integration across mobile platforms. The POTION AI application is illustrated in Figure 2.

Figure 3 depicts the operational workflow of the proposed POTION AI system. The procedure begins with video acquisition using a GoPro Hero 8 camera, followed by frame extraction and image preprocessing. The processed frames are subsequently analyzed by the YOLOv8 model to detect potholes. The detected regions are then refined using the Mask R-CNN model to generate instance-level segmentation masks and estimate pothole dimensions. Within the POTION AI application, direct pothole detection is performed independently using the trained YOLOv8 and Mask R-CNN models. From the detection results of each model, a confidence rate is obtained for YOLOv8, and a score value is obtained for the Mask R-CNN model. The resulting scores are then summed and divided by the total number of detected potholes to calculate the average confidence rate for each model. Subsequently, the model with the higher average score between the two is selected as the best-performing model and displayed in the POTION AI application.

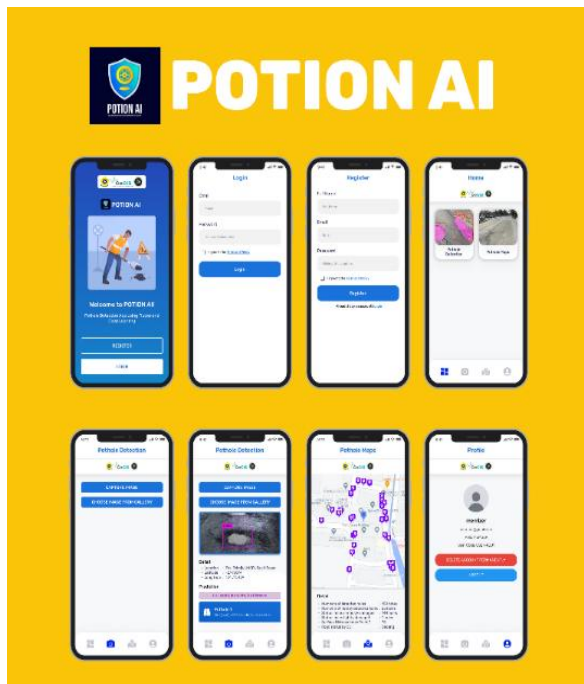


Figure 2. POTION AI application

The resulting inference outputs, including spatial location and dimensional information, were transmitted to the Flutter-based application layer. Finally, the detected potholes were displayed on an interactive map using Leaflet JS, enabling practical road condition monitoring.

RESULTS AND DISCUSSION

Training Results

The YOLOv8 model was trained for 50, 100, and 200 epochs. The best-performing configuration was obtained at 50 epochs, yielding a box mean average precision (mAP) of 0.816 and a mask mAP of 0.801. The Mask R-CNN model was trained for 700, 800, and 900 iterations, with the optimal configuration achieved at 900 iterations, corresponding to the lowest segmentation loss of 0.2028. These outcomes are critical because model performance directly affects the accuracy of pothole detection and dimension estimation. Accordingly, the best-performing configuration of each algorithm was selected for subsequent evaluation and system deployment. The corresponding results are presented in Figure 4.

As summarized in Table 4 and Table 5, the YOLOv8 model achieves strong detection performance with relatively few training epochs, whereas the Mask R-CNN model benefits from additional training iterations, resulting in improved segmentation accuracy. This behavior indicates effective convergence of both architectures under their respective optimal configurations. Overall, the results demonstrate that YOLOv8 and Mask R-CNN provide reliable performance for pothole detection and dimension estimation. The YOLOv8 model trained for 50 epochs achieves high detection accuracy with favorable computational efficiency, while the Mask R-CNN model trained for 900 iterations attains higher segmentation accuracy, as reflected by the lowest segmentation loss of 0.2028.

The observed performance metrics have direct implications for real-world applications. Higher detection accuracy and lower segmentation loss increase the reliability of pothole localization and dimension estimation under real-world operating conditions. In particular, precise instance segmentation is essential for accurate pothole size estimation, which is critical for maintenance prioritization and informed decision-making in road management systems. Consequently, the optimal configurations were selected for implementation within the proposed mobile application.

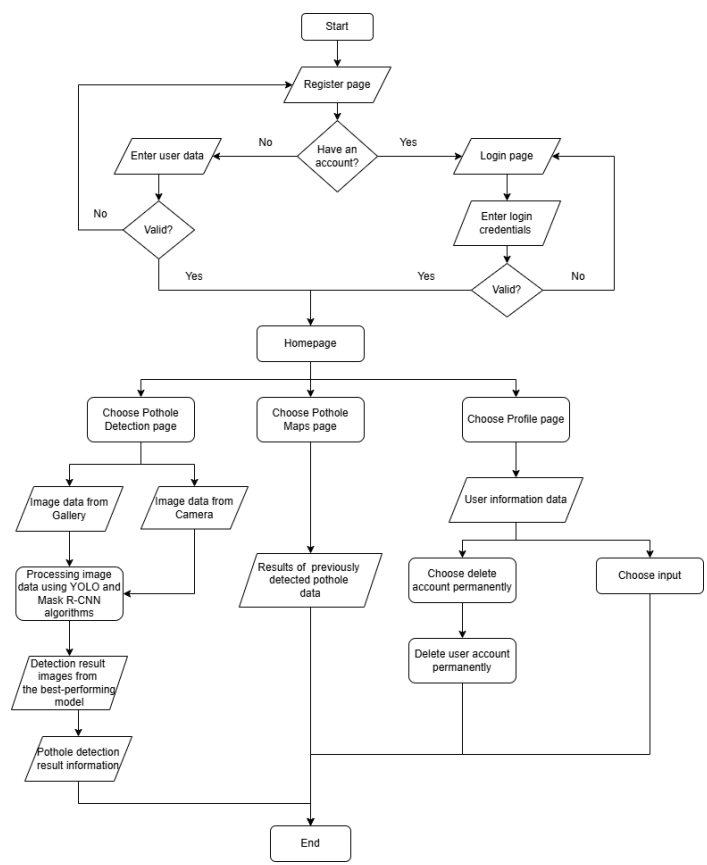


Figure 3. Workflow of the proposed POTION AI system

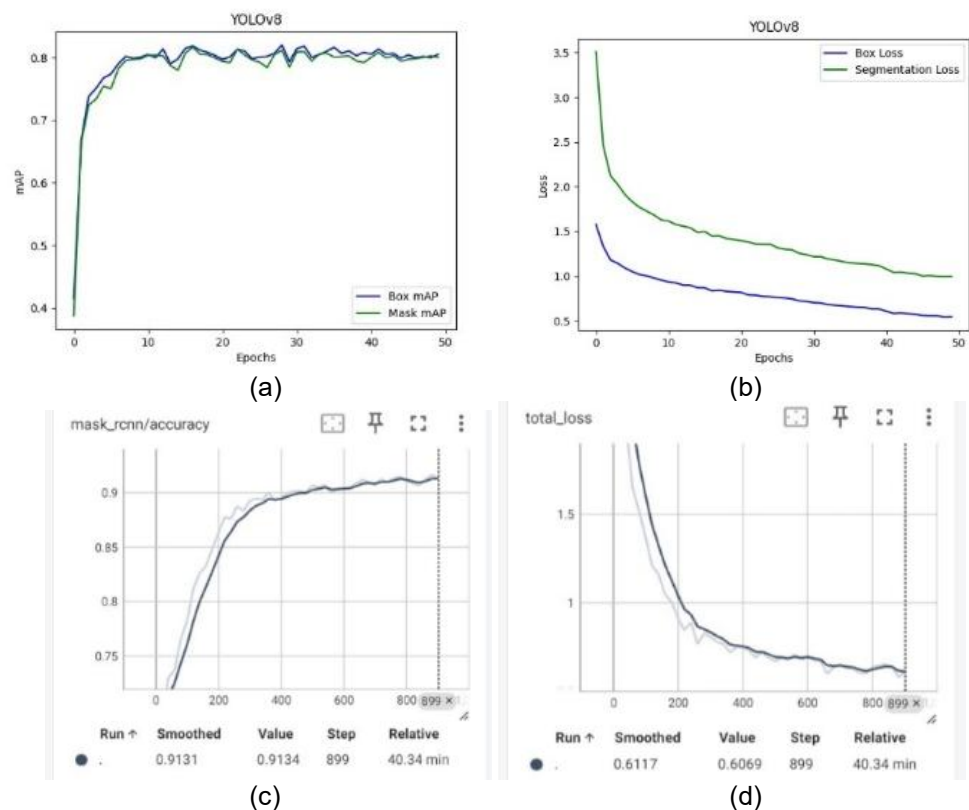


Figure 4. Training loss curves for YOLOv8 and Mask R-CNN: (a) box and mask mAP, (b) box and segmentation losses, (c) accuracy of Mask R-CNN, and (d) loss of Mask R-CNN

Table 4. Performance metrics of YOLOv8

Metric	Metrics Value		
	50 Epochs	100 Epochs	200 Epochs
Box mAP	0.816	0.817	0.812
Mask mAP	0.801	0.802	0.801
Box recall	0.746	0.764	0.747
Mask recall	0.743	0.759	0.748
Box precision	0.801	0.773	0.782
Mask precision	0.788	0.768	0.781
Average	0.7825	0.7805	0.7785

Table 5. Training loss of Mask R-CNN.

Training Loss	Iteration		
	700	800	900
Mask Loss	0.2067	0.2056	0.2028
Class Loss	0.1208	0.1130	0.1158
Box Regression Loss	0.2971	0.2783	0.2956
RPN Bbox Loss	0.0083	0.0079	0.0081
RPN Class Loss	0.0053	0.0060	0.0046

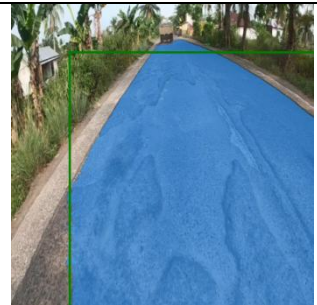
Pothole Detection and Dimension Estimation

The POTION AI application employed both YOLOv8 and Mask R-CNN models to detect potholes on road surfaces and estimate their geometric dimensions. During inference, YOLOv8 generated detection confidence scores, whereas Mask R-CNN produced segmentation-based scores. The average scores obtained from each model were compared to determine the most optimal model for application-level deployment. From the user's perspective, operation was limited to capturing a road surface image, after which the POTION AI application automatically processed the input to identify potholes. Based on comparative evaluation, the model demonstrating the best performance was selected for final detection within the application.

A comparison of the proposed POTION AI with existing methods is presented in Table 6. Specifically, YOLOv5 achieves a lower confidence score of 0.68, whereas the Segment Anything Model (SAM) may produce a higher confidence score of up to 0.84. However, because SAM functions as a general-purpose segmentation tool without target-specific optimization, it may incorrectly classify non-pothole objects as potholes. In contrast, both YOLOv8 and Mask R-CNN achieve higher confidence scores and provide more reliable pothole detection performance.

Unlike conventional approaches that rely on a single model, the proposed POTION AI performed pothole detection independently using both YOLOv8 and Mask R-CNN. The average confidence score for YOLOv8 and the average detection score for Mask R-CNN were calculated for each inspection session.

Table 6. Comparison of the proposed POTION AI with existing methods

Methods	Results
YOLOv5	 Confidence score = 68%
YOLOv8	 Confidence score = 95%
Mask R-CNN	 Confidence score = 94.04%, 90.88%, 89.99%, 79.64%, 76.15%
SAM	 Confidence score = 84%
Proposed POTION AI Framework	Compared YOLOv8 and Mask R-CNN and used the optimal model



The model with the higher average score was then automatically selected, and its prediction was used as the final detection result. For example, when YOLOv8 produced a higher average confidence score than Mask R-CNN, the YOLOv8 prediction is selected. This adaptive decision strategy enabled POTION AI to leverage the complementary strengths of both models, combining the fast and robust detection capability of YOLOv8 with the precise instance segmentation capability of Mask R-CNN. Consequently, the framework developed in this study provided a more reliable and practical solution for pothole detection and geometric dimension estimation under varying road conditions.

Furthermore, a pothole dimension estimation algorithm incorporating depth information [27] was employed to estimate pothole length, width, and distance from the camera. The estimated dimensions were validated against ground-truth measurements obtained manually using a measuring tape. The comparison between the estimated dimensions and the ground-truth measurements is presented in Table 7.

As shown in Table 7, the estimation errors for the pothole width and height are relatively higher when the pothole is located close to the camera. As the camera-to-pothole distance increases, the estimation errors gradually decrease. This trend is primarily attributed to the use of a fixed pixel-to-centimeter scale factor for dimension estimation. At close distances, perspective distortion is more pronounced, causing the relationship between image pixels and real-world dimensions to vary significantly across the image. Consequently, the fixed scale factor cannot accurately represent the true geometric projection, resulting in larger estimation errors.

As the camera moves farther from the pothole, the effects of perspective distortion are reduced, and the projected dimensions become more uniform. Under these conditions, the fixed scale factor provides a better approximation of the actual physical dimensions, affording lower estimation errors. Furthermore, the entire pothole

is more likely to be captured within the camera's field of view at greater distances, allowing the YOLOv8 and Mask R-CNN models to delineate the pothole boundaries more consistently. These factors collectively contribute to the improved accuracy of width and height estimation observed in the experimental results.

Although the proposed method achieves satisfactory performance within the evaluated distance range of 0.5–3 m, its accuracy remains dependent on the camera-to-object distance owing to the assumption of a constant scale factor. Consequently, camera calibration techniques and depth-aware geometric transformations will be studied in future to replace the fixed scaling factor, thereby improving dimension estimation accuracy over a wider range of viewing distances.

To quantitatively evaluate the performance of the proposed dimension estimation method, the Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) were calculated. The MAE values are 4.08 cm for the width and 13.88 cm for the height, while the RMSE values are 6.14 cm and 25.96 cm, respectively. Furthermore, the corresponding MAPE values are 7.36% for the width and 19.95% for the height. These results indicate that the proposed method estimates the pothole width more accurately than the height. The more pronounced errors observed in height estimation are likely attributable to the irregular geometry of potholes, variations in road surface elevation, and the sensitivity of the depth estimation process to changes in camera perspective and viewing angle.

Table 7. Comparison between the estimated dimensions and the ground-truth measurements

Actual (cm)	Predicted (cm)	Distance (m)	Width error (cm)	Height error (cm)
W: 50 H: 27	W: 42.04 H: 27.06	2.10	7.96	0.06
W: 42 H: 39	W: 42.35 H: 29.58	1.91	0.35	9.42
W: 70 H: 25	W: 73.27 H: 34.62	1.66	3.27	9.62
W: 90 H: 205	W: 86.56 H: 278.82	0.65	3.44	73.82
W: 53 H: 35	W: 52.24 H: 30.21	1.87	0.76	4.79
W: 59 H: 47	W: 74.5 H: 66.72	0.98	15.5	19.72
W: 57 H: 25	W: 59.35 H: 28.95	1.87	2.35	3.95
W: 30 H: 37	W: 32.46 H: 34.62	1.94	2.46	2.38
W: 60 H: 41	W: 60.59 H: 42.17	1.73	0.59	1.17
MAE	W = 4.08 cm			H = 13.88 cm
MAPE	W = 7.36%			H = 19.95 %
RMSE	W = 6.14 cm			H = 25.96 cm

Note: W is width and H is height

Despite these challenges, the obtained error values indicate that the proposed method provides sufficiently accurate geometric measurements for practical road inspection. The relatively low MAPE for width estimation indicates reliable measurement of pothole lateral dimensions, which are essential for assessing road deterioration. Although height estimation shows higher errors, the results remain acceptable for pavement condition assessment and maintenance planning. Moreover, integrating YOLOv8 and Mask R-CNN enables accurate pothole localization and boundary segmentation, improving the geometric features used for dimension estimation.

Overall, the experimental results demonstrate that the proposed framework not only achieves reliable pothole detection but also provides quantitative dimension estimation with acceptable accuracy under real-world operating conditions. This capability extends the functionality of conventional pothole detection systems, which typically provide only object localization without estimating the physical dimensions required for road damage assessment and maintenance prioritization.

Following detection, the application stores all relevant outputs, including the estimated pothole dimensions and geographic location information represented by latitude and longitude coordinates. An example of the pothole detection and dimension estimation results obtained using YOLOv8 is presented in Figure 5.



(a)



(b)

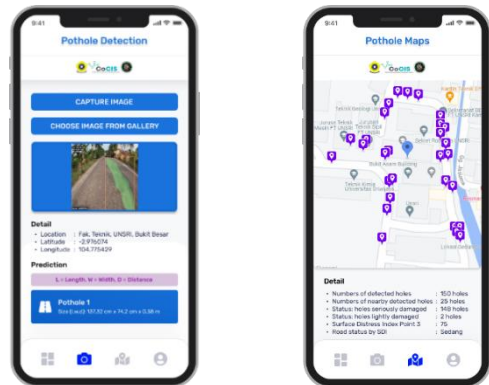
Figure 5. Pothole detection and dimension estimation results obtained using YOLOv8: (a) confidence score and (b) depth information

POTION AI Application

The POTION AI application interface is depicted in Figure 6. Previously detected potholes (Figure 6a) are displayed on the pothole map page (Figure 6b). This information is presented on an interactive map developed using Leaflet.js, a widely used JavaScript library for lightweight and user-friendly interactive mapping. The map displays the locations of detected potholes and allows users to access detailed information about nearby road conditions within a 5 km radius. This functionality enables users to track reported potholes and review maintenance actions that have been undertaken. The interactive mapping feature is particularly valuable for road authorities in planning and prioritizing maintenance activities.

Performance Evaluation of POTION AI

During the application-level evaluation, a comparative analysis was conducted between detections performed on a local computer and those executed through the POTION AI mobile application.



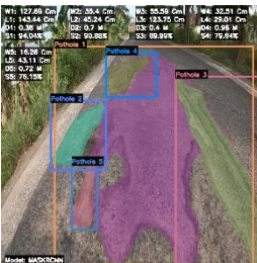
(a)

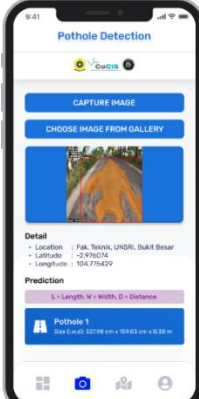
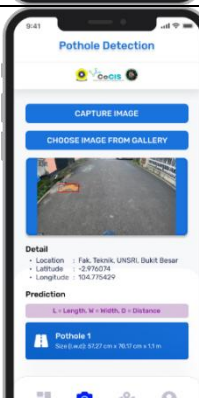
(b)

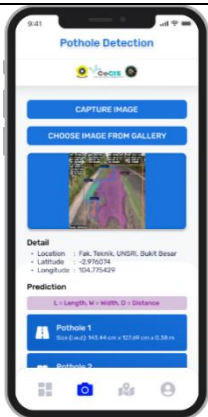
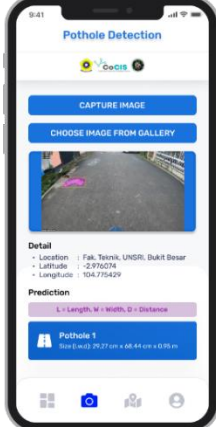
Figure 6. POTION AI application interface: (a) pothole detection, (b) pothole maps

Identical models, namely YOLOv8 and Mask R-CNN, were employed in both environments. The evaluation metrics included detection confidence and processing delay for each test scenario. The models generated bounding boxes with associated confidence scores, as shown in Table 8.

Table 8. Comparison between the local computer and POTION AI application

Type	Model	Picture	Score	Width, Length, Distance	Delay (s)
Local Computer	YOLOv8		96.05%	93.98 cm, 316.58 cm, 0.39 m	4.98
			95.03%	159.83 cm, 337.98 cm, 0.38 m	6.2
			93.04%	70.17 cm, 57.27 cm, 1.1 m	5.2
	Mask R-CNN		97.83%	74.2 cm, 137.32 cm, 0.38 m	8.9
			94.04%, 90.88%, 89.99%, 79.64%, 76.15%	P1: 127.69 cm, 143.44 cm, 0.38 m; P2: 35.4 cm, 45.24 cm, 0.7 m; P3: 55.59 cm, 123.75 cm, 0.4 m; P4: 32.51 cm, 29.01 cm, 0.96 m; P5: 16.26 cm, 43.11 cm, 0.72 m	10.8
			97.51%	68.44 cm, 29.27 cm, 0.95 m	8.7

Type	Model	Picture	Score	Width, Length, Distance	Delay (s)
Application POTION AI	YOLOv8		96.05%	93.98 cm, 316.58 cm, 0.39 m	9.11
			95.03%	159.83 cm, 337.98 cm, 0.38 m	9.92
			93.04%	70.17 cm, 57.27 cm, 1.1 m	9.02
	Mask R-CNN		97.83%	74.2 cm, 137.32 cm, 0.38 m	13.66

Type	Model	Picture	Score	Width, Length, Distance	Delay (s)
			94.04%, 90.88%, 89.99%, 79.64%, 76.15%	P1: 127.69 cm, 143.44 cm, 0.38 m; P2: 35.4 cm, 45.24 cm, 0.7 m; P3: 55.59 cm, 123.75 cm, 0.4 m; P4: 32.51 cm, 29.01 cm, 0.96 m; P5: 16.26 cm, 43.11 cm, 0.72 m	15.20
			97.51%	68.44 cm, 29.27 cm, 0.95 m	14.22
Average confidence score					92.22%
Average delay local computer					7.46
Average delay application POTION AI					11.85

The experimental results demonstrated that both models exhibited consistent performance on the local computer and within the POTION AI application. YOLOv8 and Mask R-CNN achieved identical accuracy scores across both platforms, attaining 96.05% and 97.83%, respectively.

However, notable differences were observed in processing delay, with lower latency recorded on the local computer. Specifically, YOLOv8 recorded delays of 4.98 s on the local computer and 9.11 s within the POTION AI application, while Mask R-CNN exhibited delays of 8.90 s and 13.66 s, respectively. Additional experiments revealed variations in confidence scores accompanied by corresponding changes in processing delay across both platforms; however, the overall performance trend remained consistent. The average processing delay was 7.46 s on the local computer and 11.85 s within the POTION AI application, indicating that detection accuracy was maintained while mobile deployment incurred longer processing times than local execution.

Practical Web Implementation of POTION AI

In addition to the Android-based mobile application, a web-based platform was developed, as shown in Figure 7, to support real-time pothole detection. This implementation was designed to evaluate model performance in both local and cloud-based deployment environments. The experimental results demonstrated that, when executed on a local system, the pothole detection model operated in real time with a low latency of 120 ms. By contrast, deployment on the cloud server resulted in a substantially higher latency of 9.325 ms. These results indicated that local processing is more suitable for applications requiring rapid response and real-time decision-making.

Table 9 summarizes the latency comparison between local and cloud-based deployments, highlighting the substantial performance gap under real-time operating conditions. On the real-time POTION AI website, the Mask R-CNN model achieved confidence scores of 94.50%, 91.30%, and 89.49%, with corresponding processing delays of 17.2 s, 18.9 s, and 17.1 s, respectively.

Table 9. Comparison between local computers and real-time POTION AI website.

Type	Model	Picture	Score	Width, Length, Distance	Delay (s)
Local Computer	YOLOv8		96.05%	93.98 cm, 316.58 cm, 0.39 m	4.98
			95.03%	159.83 cm, 337.98 cm, 0.38 m	6.2
			93.04%	70.17 cm, 57.27 cm, 1.1 m	5.2
	Mask CNN		97.83%	74.2 cm, 137.32 cm, 0.38 m	8.9
			94.04%, 90.88%, 89.99%, 79.64%, 76.15%	P1: 127.69 cm, 143.44 cm, 0.38 m; P2: 35.4 cm, 45.24 cm, 0.7 m; P3: 55.59 cm, 123.75 cm, 0.4 m; P4: 32.51 cm, 29.01 cm, 0.96 m; P5: 16.26 cm, 43.11 cm, 0.72 m	10.8
			97.51%	68.44 cm, 29.27 cm, 0.95 m	8.7
Website POTION AI	YOLOv8		74.89%	34.62 cm, 79.93 cm, 0.86 m	9.2
			80.34%	7.73 cm, 19.51 cm, 2.8 m	10.7

Type	Model	Picture	Score	Width, Length, Distance	Delay (s)
			84.98%	17 cm, 13.22 cm, 4 m	9.3
			94.50%	4.13 cm, 4.66 cm, 0.58 m	17.2
Mask CNN	R-		91.30%	1.03 cm, 0.74 cm, 2.96 cm	18.9
			97.47%, 89.49%	0.48 cm, 0.40 cm, 5.43 m	17.1
Average delay local computer					7.46
Average delay website POTION AI					13.73

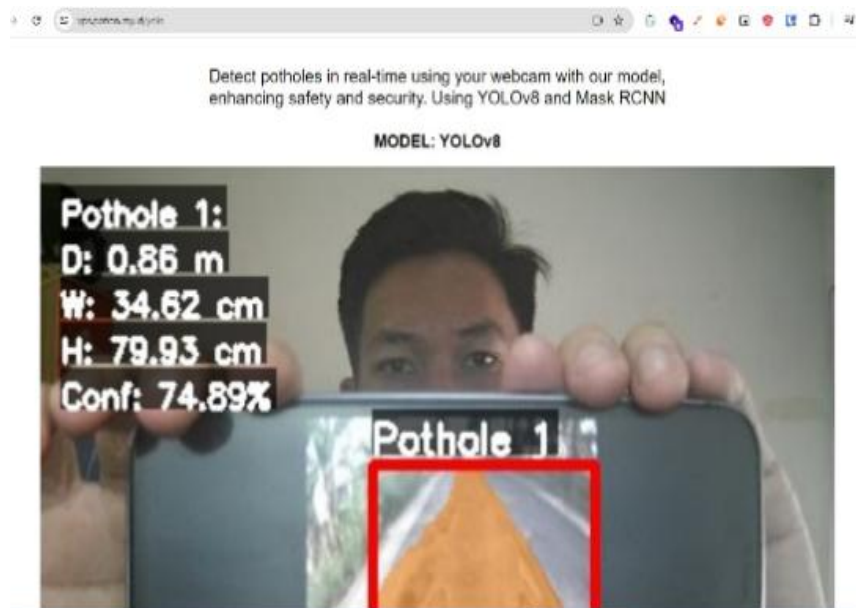


Figure 7. POTION AI Website

The final evaluation indicated that the average processing delay on a local computer was 7.46 s on the local computer and 13.73 s on the real-time POTION AI website. These results demonstrate that web-based deployment incurs substantially higher processing latency than local execution.

In addition, notable differences in detection accuracy were observed between website-based and local deployments, particularly in terms of confidence scores and depth-scale estimation. Although cloud-based

deployment offers advantages in data storage, scalability, and accessibility, limited CPU resources and the absence of GPU acceleration significantly increase computational latency. These findings suggest that local deployment is more suitable for applications requiring rapid and efficient responses. Accordingly, local processing is recommended for time-critical road monitoring tasks, whereas cloud-based deployment is better suited for non-real-time analysis or large-scale data aggregation.

The overall statistical evaluation is presented in Table 10. As shown in the table, YOLOv8 outperformed Mask R-CNN in pothole detection performance. These results suggest that YOLOv8 is more suitable for deployment in both web-based and mobile applications. In addition, YOLOv8 exhibited lower processing latency than Mask R-CNN. Furthermore, Mask R-CNN occasionally misclassifies potholes.

Surface Distress Index (SDI) Analysis

An SDI analysis was subsequently conducted to assess road damage severity based on visual indicators, including the number and size of potholes, surface cracks, and rut depth. As summarized in Table 11, 170 potholes were identified along a 2-km road segment extending from Kampung Bali through Sungai Dua, Jalan Sungai Kundur, and Sungai Kedukan in Banyuasin I Subdistrict, Banyuasin Regency, Sumatra Selatan. The calculated SDI value of 225 classified the road condition as severely damaged. These results highlighted the urgent need for maintenance and rehabilitation measures to improve road quality and reduce the risk of traffic accidents.

Application Usability Testing

To assess application usability and user experience, internal users were asked to complete a structured questionnaire evaluating four key aspects: learnability, flexibility, effectiveness, and user satisfaction. The results indicated a high level of usability, with an average score of 4.8875 out of 5 on a Likert scale. All 10 respondents reported that the user interface was intuitive and easy to understand, although several participants suggested improvements in response speed during real-time cloud-based operation. Overall, these findings suggested that the application provided strong usability and could be effectively adopted by a broad range of users.

Despite the promising results, several limitations of this study should be acknowledged. First, the accuracy of pothole dimension estimation relies on depth information derived from monocular images, which may be affected by camera calibration inaccuracies, variations in viewing angle, and environmental factors like changes in illumination and occlusions. Second, cloud-based deployment exhibited substantial latency due to limited computational resources and the absence of GPU acceleration, thereby restricting its suitability for real-time operation.

Table 10. Statistical Evaluation

Evaluation	type	Model	Mean $\pm$ standard deviation (pothole detection)
Real-time	Local computer	YOLOv8	0.947 $\pm$ 0.531
		Mask R-CNN	0.894 $\pm$ 0.078
	Website Application	YOLOv8	0.801 $\pm$ 0.0412
		Mask R-CNN	0.932 $\pm$ 0.031
Simulation	Local computer	YOLOv8	0.947 $\pm$ 0.012
		Mask R-CNN	0.894 $\pm$ 0.079
	Mobile Application	YOLOv8	0.947 $\pm$ 0.012
		Mask R-CNN	0.894 $\pm$ 0.079

Table 11. Types of road damage and calculation of SDI.

Type of SDI	Category	SDI Values
Total crack area	None	SDI1= 0
	< 10%	SDI1= 5
	10%–30%	SDI1= 20
	> 30%	SDI1= 40
Average crack width	None	SDI2 = SDI1
	< 1 mm	SDI2 = SDI1
	1–3 mm	SDI2 = SDI1
	> 3 mm	SDI2 = SDI1 * 2
Total number of potholes	None	SDI3 = SDI2
	< 10 per km	SDI3 = SDI2 + 15
	10–50 per km	SDI3 = SDI2 + 75
	> 50 per km	SDI3 = SDI2 + 225
Average depth of wheel rutting	None	SDI4 = SDI3
	< 1 cm	SDI4 = SDI3 + 2.5
	1–3 cm	SDI4 = SDI3 + 10
	> 3 cm	SDI4 = SDI3 + 2

Future studies will focus on improving depth estimation accuracy, investigating multiview or sensor-fusion strategies, and enhancing cloud-based inference via GPU-enabled servers or edge-cloud hybrid architectures. In addition, expanding the dataset to encompass a wider range of road conditions and weather scenarios is expected to further improve model robustness and generalizability.

CONCLUSION

A unified deep-learning-based framework was developed in this study for joint pothole detection and dimension estimation implemented within a mobile application environment. The YOLOv8 and Mask R-CNN algorithms achieved reliable performance in identifying and measuring road potholes, affording detection accuracies of 96.05% and 97.83%, respectively, confirming the reliability of the developed framework. These results highlighted the effectiveness of the proposed approach in addressing the limitations of conventional detection-only systems.

Beyond accuracy, the results emphasized the importance of computational architecture in real-time deployment. Application-level evaluation showed that the system operates effectively on both local computers and mobile platforms, although notable differences in processing time were observed. Real-time testing revealed significantly lower latency on local systems (120 ms) than cloud-based deployment (9.325 ms), suggesting that local processing is more suitable for time-critical applications. This finding underscored the need to balance accuracy and computational efficiency when designing deployable intelligent monitoring systems.

Furthermore, the successful implementation and positive usability evaluation indicated that the developed framework was technically robust as well as practically applicable. Usability testing involving 10 participants yielded an average score of 4.8875 out of 5 on a Likert scale, indicating that the application was intuitive and easy to operate. By providing both detection and quantitative dimension estimation, the system offered more informative outputs to support data-driven maintenance decision-making.

Overall, this study contributes to the advancement of automated road condition monitoring by bridging the knowledge gap between pothole detection and dimension estimation within a unified framework suitable for practical deployment.

Nevertheless, in this study, system performance was primarily evaluated under limited operating conditions, and the effects of diverse environmental conditions were not comprehensively investigated. Future research will focus on improving the model's robustness under varying environmental conditions, including changes in lighting, weather, and road textures. In addition, further research will explore lightweight model optimization for edge and mobile deployment, as well as extending the framework to support multi-class road damage detection and more accurate 3D geometric estimation.

ACKNOWLEDGMENT

This work is funded by the Indonesia Directorate of Research and Community Service, Directorate General of Research and Development, Ministry of Higher Education, Science, and Technology, based on Research Contract No. 109/C3/DT.05.00/PL/2025.

DATA AVAILABILITY

The data that support the findings of this study are available on request from the corresponding author.

REFERENCES

- [1] Badan Pusat Statistik Sumatera Selatan, "Panjang Jalan Menurut Kondisi Jalan (km) [Road Length by Road Condition (km)], 2020-2022." Accessed: Nov. 15, 2023. [Online]. Available: <https://sumsel.bps.go.id/indicator/17/547/1/panjang-jalan-menurut-kondisi-jalan.html>.
- [2] B. Buřcko, E. Lieskovská, K. Záborská, and M. Záborský, "Computer Vision Based Pothole Detection under Challenging Conditions," *Sensors*, vol. 22, no. 22, pp. 1–18, 2022, doi: 10.3390/s22228878.
- [3] Y. K. Yik, N. E. Alias, Y. Yusof, and S. Isaak, "A real-time pothole detection based on deep learning approach," *Journal of Physics: Conference Series*, vol. 1828, no. 1, 2021, doi: 10.1088/1742-6596/1828/1/012001.
- [4] M. Chong and J. Ying, "Real-time pothole detection system on vehicle using improved YOLOv5 in Malaysia," *IECON 2022 – 48th Annu. Conf. IEEE Ind. Electron. Soc.*, no. December, pp. 1–5, 2020, doi: 10.1109/IECON49645.2022.9968423.
- [5] N. K. Rout, G. Dutta, V. Sinha, A. Dey, S. Mukherjee, and G. Gupta, "Improved Pothole Detection Using YOLOv7 and ESRGAN," arXiv preprint arXiv:2401.08588, 2023.

- [6] A. Lincy, G. Dhanarajan, S. K. S, and B. Gobinath, "Road Pothole Detection System," In *ITM Web of Conferences*. EDP Sciences vol. 01008, pp. 1–12, 2023, doi: 10.1051/itmconf/20235301008.
- [7] Y. Zanevych, V. Yovbak, O. Basystiuk, N. Shakhovska, S. Fedushko, and S. Argyroudis, "Evaluation of Pothole Detection Performance Using Deep Learning Models Under Low-Light Conditions," *Sustainability*, vol. 16, no. 24, 2024, doi: 10.3390/su162410964.
- [8] B. Mohan Prakash and K. C. Sriharipriya, "Enhanced pothole detection system using YOLOX algorithm," *Autonomous Intelligence Systems*, vol. 2, no. 1, 2022, doi: 10.1007/s43684-022-00037-z.
- [9] R. Mirajkar, A. Yenikar, S. Nawalkar, R. Kaul, A. Rokade, and K. Rothe, "Enhanced Pothole Detection in Road Condition Assessment Using YOLOv8," *2024 IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE)*, 2024, pp. 429–433, doi: 10.1109/ICWITE59797.2024.10502437.
- [10] S. D. Achirei, I.A. Opariuc, O. Zvoristeanu, Otilia and S. Caraiman, V.I Manta, "Pothole Detection for Visually Impaired Assistance." In *2021 IEEE 17th International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2021, pp. 409–415, doi: 10.1109/iccp53602.2021.9733610.
- [11] W. Ye, W. Jiang, Z. Tong, D. Yuan, and J. Xiao, "Convolutional neural network for pothole detection in asphalt pavement," *Road Material Pavement Design*, vol. 0, no. 0, pp. 1–17, 2019, doi: 10.1080/14680629.2019.1615533.
- [12] E. Salcedo, M. Jaber, and J.R. Carrión, "A Novel Road Maintenance Prioritisation System Based on Computer Vision and Crowdsourced Reporting," *Journal of Sensor and Actuator Networks*, vol. 11, no. 1, 15, 2022, doi: 10.3390/jsan11010015.
- [13] R. Alfan, R. Fahrizal, T. P. Utomo, T. Firmansyah, and F. Muhammad, "Implementation of wavelet method and backpropagation neural network on road crack detection based on image processing," *Sinergi*, vol. 28, no. 03, pp. 489–496, 2024. doi: 10.22441/sinergi.2024.3.005
- [14] K. R. Ahmed, "Smart pothole detection using deep learning based on dilated convolution," *Sensors*, vol. 21, no. 24, 2021, doi: 10.3390/s21248406.
- [15] R. Bharat, A. M. Ikotun, A. E. Ezugwu, L. Abualigah, M. Shehab, and R. A. Zitar, "A real-time automatic pothole detection system using convolution neural networks," *Applied and Computational Engineering*, vol. 6, no. 1, pp. 750–757, 2023, doi: 10.54254/2755-2721/6/20230948.
- [16] O. A. Egaji, G. Evans, M. G. Griffiths, and G. Islas, "Real-time machine learning-based approach for pothole detection," *Expert Systems With Applications*, vol. 184, p. 115562, 2021, doi: 10.1016/j.eswa.2021.115562.
- [17] F. Ozoglu and T. Gökgöz, "Detection of road potholes by applying convolutional neural network method based on road vibration data," *Sensors*, vol. 23, no. 22, 2023, doi: org/10.3390/s23229023.
- [18] A. Rahman, R. Mustafa, and M. S. Hossain, "Real-time pothole detection and localization using convolutional neural network," in *Proceedings of the International Conference on Big Data, IoT, and Machine Learning: BIM*, 2021, pp. 579–592, doi: 10.1109/ies53407.2021.9594038.
- [19] G. Kim and S. Kim, "A Road Defect Detection System Using Smartphones," *Sensors*, vol. 24, no. 7, 2024, doi: 10.3390/s24072099
- [20] A. K. Pandey, V. Palade, R. Iqbal, T. Maniak, C. Karyotis, and S. Akuma, "Convolution neural networks for pothole detection of critical road infrastructure," *Computers and Electrical Engineering*, vol. 99, pp. 1–20, 2022, doi: 10.1016/j.compeleceng.2022.107725.
- [21] M. Khan, M. A. Raza, G. Abbas, S. Othmen, A. Yousef, and T. A. Jumani, "Pothole detection for autonomous vehicles using deep learning: a robust and efficient solution," *Frontiers in Built Environment*, vol. 9, no. January, pp. 1–17, 2023, doi: 10.3389/fbuil.2023.1323792.
- [22] S. R. Kuthyar et al., "An Intelligent Pothole Detection and Alerting System using Mobile Sensors and Deep Learning," *2021 IEEE 18th India Council International Conference (INDICON)*, India, 2021, doi: 10.1109/INDICON52576.2021.9691661.
- [23] L. V. Fortin and O. E. Llantos, "Performance Analysis of YOLO versions for Real-time Pothole Detection," *Procedia Computer Science*, vol. 257, pp. 77–84, 2025, doi: 10.1016/j.procs.2025.03.013.
- [24] E. Ranyal, A. Sadhu, and K. Jain, "AI assisted pothole detection and depth estimation," *2023 International Conference on Machine Intelligence for GeoAnalytics and Remote Sensing (MIGARS)*, 2023, vol. 1, pp. 1–4, doi:

- 10.1109/MIGARS57353.2023.10064547.
- [25] R. Rubin, C. Jacob, S. Sundar, G. Stoian, D. Danciulescu, and J. Hemanth, "Pothole Detection and Assessment on Highways Using Enhanced YOLO Algorithm With Attention Mechanisms," *Advances in Civil Engineering*, vol. 2025, no. 1, 2025, doi: 10.1155/adce/7911336
- [26] A.K. Bhatt, H. Raj, V.B. Sharma, S. Biswas, A.Singh, R. Silori, and M. Pandey, "Advancements in pothole detection techniques : a comprehensive review and comparative analysis," *Discover Artificial Intelligence*, vol. 5, no. 1, pp. 255, 2025, doi: 10.1007/s44163-025-00297-7.
- [27] B. Y. Suprpto, S. Dwijayanti, D. M. A. Hafiz, F. A. Ardandy, and J. Jonathan, "Designing an Autonomous Vehicle Using Sensor Fusion Based on Path Planning and Deep Learning Algorithms," *SAIEE Africa Research Journal*, vol. 115, no. 3, pp. 86–98, 2024, doi: 10.23919/saiee.2024.10551314.