

Performance Evaluation of YOLO26 for Pothole Detection Under Diverse Environmental Conditions

Edelin Stephanie ¹, Puguh Hiskiawan ^{2*}

Data Science, Faculty of Technology and Design, Bunda Mulia University
s36230008@student.ubm.ac.id ¹, phiskiawan@bundamulia.ac.id ²

Article Info

Article history:

Received 2026-08-18

Revised 2026-09-11

Accepted 2026-10-16

Keyword:

Computer Vision,
Diverse Conditions,
Object Detection,
Pothole,
YOLO26.

ABSTRACT

Road potholes are a serious threat to road safety. They can damage vehicles and increase the risk of traffic accidents. However, factors like lighting and weather can impact how well computer vision systems detect potholes. To tackle this issue, this study compares four You Only Look Once (YOLO) 2026 (YOLO26) variants: YOLO26n, YOLO26s, YOLO26m, and YOLO26l, for detecting potholes in different environmental conditions. A dataset was independently collected in the Poris area of Tangerang, Indonesia, using a forward-facing camera to capture a driver's view. The video recordings were turned into annotated image frames in YOLO format, creating 368 images that included clear, rain, evening, and night conditions. Each model was trained for 200 epochs under the same experimental conditions. Their performance was evaluated using Precision, Recall, mean Average Precision at an Intersection over Union (IoU) threshold of 0.5 (mAP@0.5), and mean Average Precision across IoU thresholds from 0.5 to 0.95 (mAP@0.5:0.95). Computational efficiency was evaluated using parameter count, model size, training time, GPU memory usage, inference time, and frames per second (FPS) to assess the computational requirements and real-time detection capability of each YOLO26 variant. The results showed that YOLO26m achieving the highest Recall of 0.877 and mAP@0.5 of 0.934, while YOLO26l achieved the highest Precision of 0.989 and mAP@0.5:0.95 of 0.735. On the other hand, YOLO26n was the most computationally efficient option while still providing competitive detection performance. These results highlight the trade-off between detection accuracy and computational efficiency among YOLO26 variants. They also offer useful guidance for choosing the right model for real-time pothole detection in various conditions, which can help in the development of future driver assistance systems and pothole warning systems.



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

I. INTRODUCTION

Secure and dependable roadway infrastructure is crucial for enduring transportation. It enables public mobility, economic activity, and the smooth distribution of goods and services [1]. The degradation of road surfaces, particularly potholes, constitutes a prevalent kind of pavement impairment and poses a significant hazard to road users [2]. Potholes may inflict damage on vehicles, lead to abrupt loss of control, and result in traffic collisions, particularly when drivers fail to notice road defects in time to take evasive action

[3], [4]. The risk increases in bad weather and low-light conditions, where poor visibility makes it tougher for drivers to detect hazards [5], [6], [7]. For this reason, automatic pothole detection under varying environmental conditions has become an increasingly studied topic within the broader field of smart transportation and driver assistance systems (ADAS) research [8].

Recent developments in deep learning have significantly advanced pattern recognition across a wide range of data modalities. In the visual domain, deep learning has greatly enhanced computer vision applications such as smart

transportation, medical image analysis, and industrial inspection [9], [10], [11]. These improvements form the basis for computer vision-based pothole detection systems that can warn vehicles about potentially dangerous road conditions [12]. Among different object detection algorithms, the You Only Look Once (YOLO) family is well-liked for its balance between detection accuracy and speed, fitting for real-time applications [13], [14], [15], [16], [17], [18], [19], [20]. Successive YOLO models have consistently shown strong detection ability without losing speed, making them effective for use on devices with limited resources for pothole detection [6], [21], [22], [23]. YOLO26, the latest version in the YOLO series, includes architectural improvements such as a better backbone made up of Cross Stage Partial blocks with a kernel size of 2 (C3K2), a Spatial Pyramid Pooling Fast (SPPF) module for performing multi-scale contextual pooling, and a Cross-Stage Partial with Position-Sensitive Attention (C2PSA) attention mechanism which enhances feature discrimination without compromising computational efficiency [24]. Because of these improvements, YOLO26 provides a more solid basis for detecting small and irregularly shaped objects like potholes than earlier YOLO architectures, which is why it has been chosen as the main architecture examined in this study. Since edge devices used in smart transportation systems have different computational limitations, it is important to understand the trade-off between detection accuracy and computational efficiency among the various YOLO26 models if a suitable model is to be selected for real-time deployment.

Beyond the model design, the quality of the training data is essential for the practical performance of computer vision systems. Recent research has moved away from traditional close-up images of road damage to forward-facing images from a driver's viewpoint, as these better reflect the visual information available during real driving. Datasets derived from dashcams offer a more realistic basis for pothole detection by capturing the viewing angle, object size, and road context that drivers experience [25]. However, this perspective brings new challenges since the appearance of potholes varies significantly in different environmental conditions, such as rain, nighttime, and low light. To address this variability, several studies have broadened their evaluations to include adverse weather and lighting scenarios to boost model reliability [5], [6], [23].

Despite these improvements, several critical research gaps still exist. Current studies have shown that changes in lighting, rainfall, reflections, and visibility impact pothole detection accuracy significantly [5], [7], [23]. However, most datasets available to the public were collected outside of Indonesia and may not capture the visual traits of Indonesian roads, like pavement materials, road markings, surrounding environments, and local lighting. As a result, the applicability of existing models to Indonesian road conditions is not well studied. Additionally, model performance is often reported as

overall detection accuracy, without evaluating reliability across various conditions. Moreover, previous studies on pothole detection have mainly used earlier versions of YOLO, such as YOLOv9 [22] and other variants prior to 2025 [21], [23], these typically being tested on datasets gathered outside of Indonesia or relying on close-up images of potholes rather than from the point of view of a driver. Although new YOLO architectures continue to improve the balance between accuracy and computational efficiency, the effectiveness of the latest YOLO26 variants for pothole detection using a locally sourced driver's perspective dataset has not been extensively tested, and to date, there has been no investigation into the effectiveness of YOLO26 for detecting potholes. Thus, selecting the appropriate YOLO26 model for real-time applications remains unresolved.

To fill these gaps, this study assesses the performance of four YOLO26 variants: YOLO26n, YOLO26s, YOLO26m, and YOLO26l, for pothole detection under different environmental conditions, with consideration of their potential suitability for real-time applications. A local dataset was gathered in the Poris area of Tangerang, Indonesia, using a smartphone positioned at a forward-facing angle to simulate a driver's viewpoint. This dataset features pothole images captured in four environmental conditions: clear, rain, evening, and night. Each model was evaluated in each condition based on Precision, Recall, mean Average Precision at an Intersection over Union (IoU) threshold of 0.5 (mAP@0.5), and mean Average Precision across IoU thresholds from 0.5 to 0.95 (mAP@0.5:0.95), while computational efficiency was assessed through parameter count, model size, training time, GPU memory usage, inference speed, and frames per second (FPS), in order to examine the trade-off between detection accuracy and computational cost across model scales. In addition, to give some context regarding the relative performance of YOLO26, a supplementary comparison was carried out with recent detectors of similar size (YOLOv8n, YOLO11n, YOLO12n). Overall, by jointly examining detection accuracy, computational efficiency, and relative performance against comparable detectors, this study aims to provide practical, evidence-based guidance for selecting an appropriate YOLO26 variant for real-time pothole detection across varied environmental conditions.

II. METHODS

The proposed framework was developed to evaluate the performance of four YOLO26 variants for pothole detection under diverse environmental conditions. The overall workflow, illustrated in Figure 1, consists of dataset acquisition, dataset preprocessing, model training, and performance evaluation. Each stage of this workflow is described in detail in the following subsections.

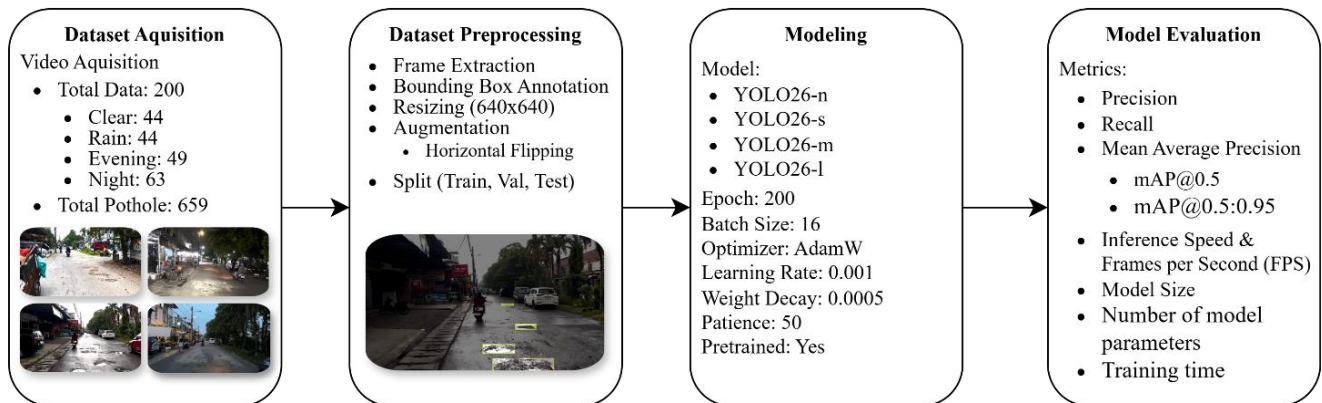


Figure 1. Proposed Method Workflow

A. Dataset Acquisition

A representative pothole dataset is fundamental for training and evaluating robust object detection models under real-world driving conditions. The data used in this study was independently collected in the Poris area of Tangerang, Indonesia, with an iPhone 15 being manually mounted in a horizontal (landscape) position and placed above the driver's head. The smartphone was employed to record forward-facing driving videos at a resolution of 1920×1080 pixels and at a frame rate of 30 frames per second. The data collection took place using a motorcycle, the smartphone being held by the passenger at a height of about 1.6 m above the road surface. The motorcycle was driven at a normal traffic speed, this being estimated at about 20–30 km/h, without maintaining a fixed speed. This arrangement was designed so that road scenes could be captured from a forward-facing point of view under natural driving conditions, taking into account variations in lighting, road surface, and traffic environment. The total length of the recordings was about 287 seconds.

Data acquisition occurred under four environmental conditions: clear, rain, evening, and night. In order to maintain consistent categorization of the environment throughout data collection, each driving scene was assigned to one of the four environmental conditions according to the lighting and weather conditions. Clear images were taken from about 09.00 AM to 12.00 PM, a time when natural sunlight produced bright illumination and allowed for clear visibility of the pavement with visible shadows. Rain condition was obtained during light rain or drizzle, with the road surface wet and the potholes partially filled with water. All the rain scenes in this dataset were recorded during the day, which meant that the wet-road reflections were caused by sunlight and not by rainfall at night. The evening images were captured between 05.00 PM and 07.00 PM, a period when the transition from daylight to dusk was taking place and the ambient lighting was slowly decreasing, with the sky showing warmer yellow and bluish-purple hues. Night images were taken after 08.00 PM, when natural daylight was no

longer present and road visibility mainly relied on street lighting and the headlights of the vehicles. These video recordings formed the raw data for the later frame extraction and dataset preprocessing stages.



Figure 2. Comparison of Pothole Image Perspectives: (a) Close-up and (b) Driver's Perspective

Difference between conventional close-up pothole images commonly used in previous studies and the driver's-perspective images adopted in this research are illustrated in Figure 2. Unlike close-up images, which focus mainly on the potholes themselves, the proposed approach captures the entire driving scene. This method offers a better match to visual information available to drivers during actual vehicle operation and is thus more effective for developing real-time pothole warning systems.

B. Dataset Preprocessing

The recorded videos were turned into image frames to construct the pothole detection dataset. Frames were extracted at an interval of one frame per second. This interval was selected to provide sufficient temporal spacing between consecutive frames while maintaining the progression of the road scene as it approached and passed potholes. This interval reduced excessive visual redundancy between consecutive frames while preserving variations in camera viewpoint and pothole appearance. Hence, the interval was regarded as sufficient to capture the spatial and visual differences of potholes while lowering the risk of excessive redundancy. The extracted frames underwent quality control to remove unusable images, such as severely blurred or overexposed

frames, while retaining challenging images to preserve real-world driving conditions.

The selected images were manually annotated using the Roboflow platform. Each pothole was annotated with a single object class (pothole) using bounding box annotations in YOLO26 format. A road defect was classified as a pothole when it showed a visible depression or cavity in the road surface, regardless of its specific shape or depth. After that, the annotations were reviewed to verify the class labels, the position of the bounding boxes, and the completeness of the potholes that had been identified. The final dataset included 200 images and a total of 659 pothole instances. Further details regarding the original dataset distribution can be found in Table 1.

TABLE I
DATASET DISTRIBUTION

Environmental Condition	Total Data	Total Pothole
Clear	44	147
Rain	44	141
Evening	49	155
Night	63	216
Total	200	659

There is some variation in the number of images depending on the environmental conditions, the Night condition having the highest proportion at 31.5%, then Evening at 24.5%, Clear at 22.0%, and Rain at 22.0%. The number of pothole instances generally matches the distribution of the images across the different conditions, with no single condition making up an unreasonably large part of the dataset. However, this variation was taken into account when interpreting the detection results, and the model's performance was therefore assessed separately for each environmental condition in order to avoid basing conclusions solely on the overall performance, especially in the cases where the number of samples is small.

After annotation, data augmentation was carried out to increase dataset diversity and improve the model's robustness under different environmental conditions. Albumentations library was used to apply horizontal flipping to the images. Augmentation was selectively applied according to the representation of each environmental condition, with greater augmentation applied to conditions with fewer original samples and limited augmentation applied to the night condition to avoid excessive duplication.



Figure 3. Augmented Strategy Applied to Dataset

This strategy was implemented to improve the balance of the dataset while preserving the characteristics of each environmental condition. Example of the applied

augmentation technique are illustrated in Figure 3. Dataset composition after augmentation is presented in Table 2.

TABLE II
DATASET COMPOSITION BEFORE AND AFTER AUGMENTATION

Environmental Condition	Original	Augmented	Final	Total Pothole
Clear	44	44	88	294
Rain	44	44	88	282
Evening	49	49	98	310
Night	63	31	94	323
Total	200	168	368	1209

The final dataset consisted of 368 annotated images containing a single object class, pothole, with a total of 1209 annotated pothole instances. The dataset was collected under four environmental conditions, comprising 88 clear images, 88 rain images, 98 evening images, and 94 night images after augmentation. As this study involves single-class object detection, the class distribution is balanced by design since all the annotations pertain to the pothole category and the rest of the pixels are considered background during training.

Even though the size of the dataset is fairly small when compared with that of large-scale public benchmarks, it was regarded as adequate for the purpose of this study since the aim had not been to create a new detection model, but rather to carry out a controlled comparative evaluation of the four YOLO26 variants under the same training conditions, dataset composition, and environmental conditions.

The dataset was subsequently divided into training and validation sets using an 80:20 ratio within each environmental condition. In order to ensure reproducibility, the split was performed using a fixed random seed of 42. This stratified splitting procedure was applied to maintain the representation of each environmental condition in both subsets and to reduce potential bias caused by differences in sample distribution. The final dataset consisted of 295 training images and 73 validation images, with the distribution across environmental conditions presented in Table 3. The training set included 66, 73, 81, and 75 images from the Clear, Rain, Evening, and Night conditions, respectively, while the validation set included 22, 15, 17, and 19 images, respectively.

TABLE III
FINAL DATASET DISTRIBUTION

Environmental Condition	Train Data (80%)	Val Data (20%)
Clear	66	22
Rain	73	15
Evening	81	17
Night	75	19
Total	295	73

C. Modeling

The architecture of YOLO26 used in this study is shown in Figure 4. The network has three main components: backbone, neck, and detection head. The backbone progressively extracts visual features from the input image using successive convolutional and C3k2 blocks. These layers transform the input image into increasingly informative feature representations, allowing the network to capture relevant characteristics of potholes, such as their shape, texture, boundaries, and surrounding road patterns. This process generates multi-scale feature maps that capture both low-level spatial information and high-level semantic representations. The SPPF module then processes the extracted features to combine multi-scale contextual information. After that, the C2PSA module further refines the features. It merges convolutional processing with attention mechanisms to improve feature discrimination while keeping computational efficiency. The neck refines the features even further through upsampling and concatenation operations, enabling effective fusion of feature representations across various scales. The detection head predicts pothole locations and confidence scores at three feature scales, ensuring robust detection of potholes of different sizes. All YOLO26 variants share the same overall architecture, but YOLO26n, YOLO26s, YOLO26m, and YOLO26l differ in model scale, affecting parameter counts, costs, and detection capabilities [24].

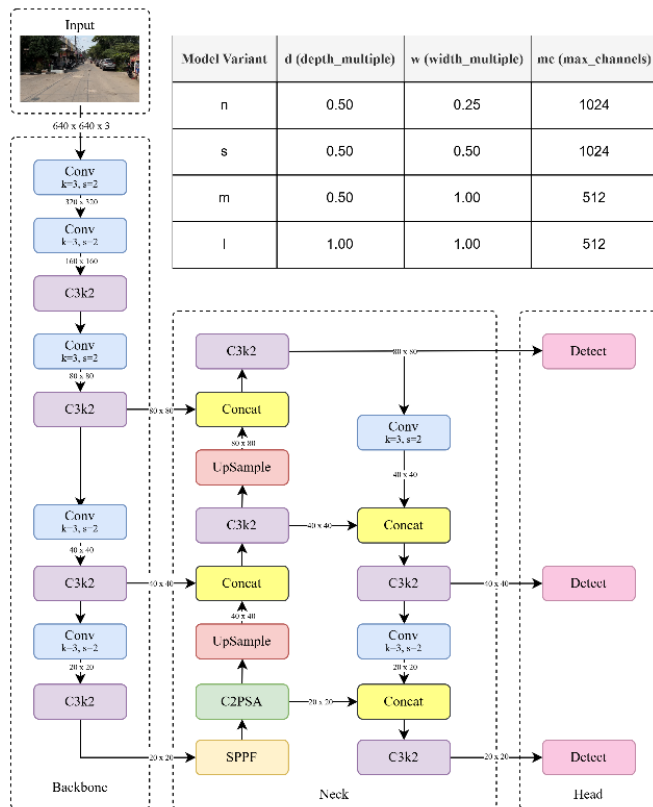


Figure 4. Architecture of YOLO26 Object Detection Network

Although all variants have the same architecture, they vary in depth and width, leading to different computational complexities and detection abilities. The lightweight YOLO26n is designed for deployment in resource-limited situations, while YOLO26l has greater representational capacity but requires more computational resources. Testing these variants under the same experimental conditions allows for a clear comparison of the balance between detection accuracy and computational efficiency for real-time pothole detection. The training process used a batch size of 16 for 200 epochs, with an input image size of 640×640 pixels. The AdamW optimizer was used with a learning rate of 0.001 and a weight decay of 0.0005. To avoid overfitting, early stopping was implemented with a patience value of 50 epochs. The use of pretrained weights provides an established initialization for feature learning. Automatic Mixed Precision (AMP) helps reduce GPU memory requirements and improve training efficiency, and a fixed random seed of 42 was applied to improve the reproducibility of the experimental results. Model training occurred on Google Colab with an NVIDIA Tesla T4 GPU. Table 4 provides the complete training configuration.

TABLE IV
TRAINING CONFIGURATION

Parameter	Value
Models	YOLO26n, YOLO26s, YOLO26m, YOLO26l
Epoch	200
Batch Size	16
Image Size	640 x 640
Optimizer	AdamW
Learning Rate	0.001
Weight Decay	0.0005
Patience	50
Pretrained	Yes
AMP	Enabled
Random Seed	42
Training Platform	Google Colab (NVIDIA Tesla T4 GPU)

The model's performance was analyzed separately under each environmental condition to assess the performance of the YOLO26 variants across varying lighting and weather conditions represented in the dataset and to identify differences in detection capability across these conditions. Evaluation used standard object detection metrics: Precision, Recall, mAP@0.5, and mAP@0.5:0.95 [22], [23], [25], [26]. Along with detection accuracy, computational efficiency was measured through model parameters, model size, training time, GPU Memory usage, inference speed, and frames per second (FPS), providing insights into the computational requirements of each model variant. These metrics collectively offer a comprehensive evaluation of detection capability and computational efficiency, enabling an assessment of the relative performance and suitability of the YOLO26 variants for real-time pothole detection under

varying environmental conditions. The confusion matrix was used to assess detection performance by providing a detailed representation of correct and incorrect predictions, and its interpretation is summarized in Table 5.

TABLE V
CONFUSION MATRIX INTERPRETATION

Description	Explanation
True Positive (TP)	Number of potholes correctly detected by the model.
True Negative (TN)	Number of non-pothole instances correctly classified as background.
False Positive (FP)	Number of non-pothole regions incorrectly detected as potholes.
False Negative (FN)	Number of potholes that were not detected by the model.

Precision and Recall provided a thorough assessment of pothole detection performance using the confusion matrix. Precision measures the proportion of correctly detected potholes among all predicted ones. A high Precision value indicates a low number of false positives, meaning most bounding boxes predicted as potholes correspond to actual potholes [27]. Precision is calculated as:

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

where TP and FP represent the number of true positive and false positive detections, respectively. Precision values range from 0 to 1, where a high value means a low number of incorrect pothole detections.

Recall measures the proportion of actual pothole instances that the model correctly detects. This metric shows the model's ability to identify potholes that exist in the ground-truth data and is especially crucial when minimizing missed detections, represented by false negatives [27]. Recall is calculated as:

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

in this case, TP and FN denote true positive and false negative detections, respectively. Higher Recall values indicate that the model can detect a larger share of the potholes noted in the ground-truth annotations.

Intersection over Union (IoU) assesses the overlap between a predicted bounding box and its actual ground-truth counterpart. IoU decides whether a predicted box aligns well with the ground-truth box and is therefore considered a correct detection at a specified threshold. IoU is calculated as:

$$IoU = \frac{B_{pred} \cap B_{gt}}{B_{pred} \cup B_{gt}} \quad (3)$$

where B_{pred} represents the predicted bounding box and B_{gt} represents the ground-truth bounding box. The intersection

represents the overlapping area between the two bounding boxes, whereas the union represents the total area covered by both bounding boxes. IoU values range from 0 to 1, where a higher value indicates greater overlap between the predicted and ground-truth bounding boxes. In this study, IoU thresholds of 0.50 to 0.95 were used to evaluate detection performance through mAP@0.5 and mAP@0.5:0.95.

Average Precision (AP) summarizes the relationship between Precision and Recall for a specific object class, measuring the area beneath the Precision-Recall curve. AP is calculated as:

$$AP = \int_0^1 P(R) dR \quad (4)$$

here, $P(R)$ represents Precision based on Recall. A higher AP value indicates better detection performance across various confidence thresholds.

Mean Average Precision (mAP) serves as the main metric for overall object detection performance, calculated as the mean of AP values across all object classes [28]:

$$mAP = \frac{1}{n} \sum_{i=1}^n AP_i \quad (5)$$

where (n) denotes the number of object classes and AP_i represents the Average Precision of the i -th class. Since this study focuses solely on one object class, pothole, the resulting mAP is numerically equivalent to the AP of the pothole class.

Detection performance was assessed using two IoU criteria: mAP@0.5 and mAP@0.5:0.95. The mAP@0.5 metric represents the mean Average Precision at an IoU threshold of 0.50, considering a predicted bounding box correctly localized when its IoU with the corresponding ground-truth bounding box is 0.50 or higher. On the other hand, mAP@0.5:0.95 reflects the mean Average Precision averaged across IoU thresholds from 0.50 to 0.95, with steps of 0.05. Thus, mAP@0.5 offers a more lenient assessment of detection performance, while mAP@0.5:0.95 provides a stricter evaluation of both detection and bounding-box accuracy since higher IoU thresholds require closer alignment between predicted and ground-truth bounding boxes [28], [29].

III. RESULTS AND DISCUSSION

A. Overall Performance Evaluation

The training and validation loss curves for the four YOLO26 variants during the training process are shown in Figure 5. In the early epochs, all models consistently reduced their loss and then gradually converged. This indicates a stable optimization process without any significant divergence. As the model size increased, the loss generally decreased, with YOLO26l achieving the lowest loss during the later training epochs, while YOLO26n had the highest loss. Larger variants typically needed more epochs to converge due to their increased complexity, but all models showed stable performance before reaching the maximum training epochs.

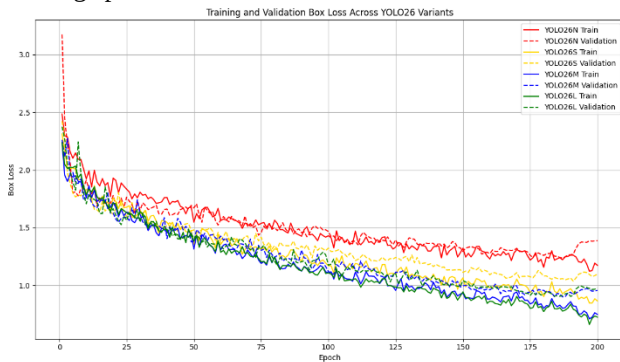


Figure 5. Training and Validation Loss Across YOLO26 Variants

Table 6 presents the overall detection performance of the four YOLO26 variants on the validation dataset. The models were assessed using Precision, Recall, mAP@0.5, and mAP@0.5:0.95 to evaluate both classification accuracy and localization ability. While all variants achieved high detection accuracy, their performance differed based on model complexity.

TABLE VI
PERFORMANCE COMPARISON OF YOLO26 VARIANTS

Model	Precision	Recall	mAP@0.5	mAP@0.5:0.95
YOLO26n	0.910	0.787	0.872	0.575
YOLO26s	0.932	0.849	0.916	0.687
YOLO26m	0.961	0.877	0.934	0.729
YOLO26l	0.989	0.871	0.933	0.735

As shown in Table 6, all of the YOLO26 variants obtained Precision and mAP@0.5 values exceeding 0.87, which shows that the dataset proposed can be learned efficiently no matter what the model size is. Detection performance improved as the model capacity increased, even though YOLO26m performed slightly better than YOLO26l in terms of Recall and mAP@0.5. YOLO26m achieved the highest Recall of 0.877 and the highest mAP@0.5 of 0.934, while YOLO26l had the highest Precision of 0.989 and an mAP@0.5:0.95 of 0.735. These results suggest that YOLO26m provided a slightly better overall detection at the 0.5 IoU threshold,

whereas YOLO26l demonstrated better precision and localization performance when stricter IoU thresholds were applied.

The various evaluation metrics show that model capacity influences the detection and localization features in different ways. YOLO26m obtained the highest Recall, which means that it was able to identify a slightly greater number of actual potholes, whereas YOLO26l achieved the highest Precision, indicating that it had fewer false-positive detections. The difference between their mAP@0.5 values was negligible and thus suggests that both models performed similarly when a moderate localization threshold was used. However, YOLO26l had a slightly higher mAP@0.5:0.95, showing a small advantage in bounding-box localization at stricter IoU thresholds. It therefore follows that YOLO26m is slightly better when it is important to minimize missed potholes, while YOLO26l is the preferable choice when both prediction precision and localization accuracy are of primary concern.

YOLO26n and YOLO26l showed steady improvement, particularly in the case of mAP@0.5:0.95, which rose from 0.575 to 0.735. Since mAP@0.5:0.95 assesses detection quality using more stringent IoU thresholds than mAP@0.5, which employs a more lenient localization threshold, this significant increase implies that the larger YOLO26 models achieve better accuracy in localizing the bounding boxes, not just in detecting potholes. The difference between mAP@0.5 and mAP@0.5:0.95 is to be expected since the latter calculates performance based on much stricter IoU thresholds ranging from 0.5 to 0.95. This gap shows that while the model is able to reliably detect potholes at an IoU of 0.5, it becomes more difficult to achieve highly accurate bounding-box alignment as the requirements for localization become more demanding. This finding aligns with previous studies showing that larger YOLO architectures usually extract richer semantic features and achieve better localization accuracy, although they come with higher computational demands [6], [21], [22], [23].

To assess the stability of the reported results and account for potential variation caused by random weight initialization, data shuffling, and stochastic augmentation, each YOLO26 variant was trained three times using different random seeds (42, 123, and 2024) under otherwise identical training configurations. The mean and standard deviation of Precision, Recall, mAP@0.5, and mAP@0.5:0.95 across the three runs are summarized in Table 7.

TABLE VII
PERFORMANCE STABILITY OF YOLO26 VARIANTS ACROSS REPEATED TRAINING RUNS

Model	Precision	Recall	mAP@0.5	mAP@0.5:0.95
	(mean±std)			
YOLO26n	0.92±0.010	0.76±0.028	0.85±0.018	0.58±0.006
YOLO26s	0.94±0.032	0.85±0.012	0.92±0.002	0.70±0.014
YOLO26m	0.96±0.005	0.89±0.011	0.94±0.005	0.74±0.008
YOLO26l	0.97±0.014	0.88±0.017	0.93±0.011	0.75±0.010

As presented in Table 7, all YOLO26 variants exhibited relatively low standard deviations across the three independent runs, with $mAP@0.5:0.95$ standard deviations ranging from 0.006 to 0.014. This indicates that the performance differences are largely not attributable to random initialization or training stochasticity, supporting the reliability of the overall comparison among YOLO26 variants. Among the four variants, YOLO26n showed the most notable variability in Recall (std = 0.028), which may be attributed to its smaller parameter count making it more sensitive to initialization and to the stochastic effects of augmentation. In contrast, YOLO26m exhibited the lowest variability across all four metrics (std ≤ 0.011), suggesting a more stable convergence behavior at this model scale. Overall, the consistency of results across repeated runs confirms that the trends observed, namely the improvement in localization accuracy with increasing model capacity, are stable and not the result of a single favorable training run.

Figure 6 demonstrates the convergence of Precision, Recall, $mAP@0.5$, and $mAP@0.5:0.95$ throughout the training process. All metrics rose quickly during the initial training epochs and gradually plateaued towards the end. This indicates that the models effectively learned the relevant features for pothole detection and reached stable performance. The larger YOLO26 variants generally achieved higher metric values than the smaller ones, with YOLO26l consistently showing the best performance overall, followed by YOLO26m and YOLO26s, while YOLO26n had the lowest values across the metrics evaluated. The variation between the models was especially noticeable in $mAP@0.5:0.95$, where the larger models maintained better performance under stricter IoU thresholds. Overall, the consistent convergence of the four metrics signals stable learning and effective optimization across all YOLO26 variants. Similar learning patterns have been reported in other deep learning applications, where stable loss convergence indicates effective optimization and reliable model training [30]

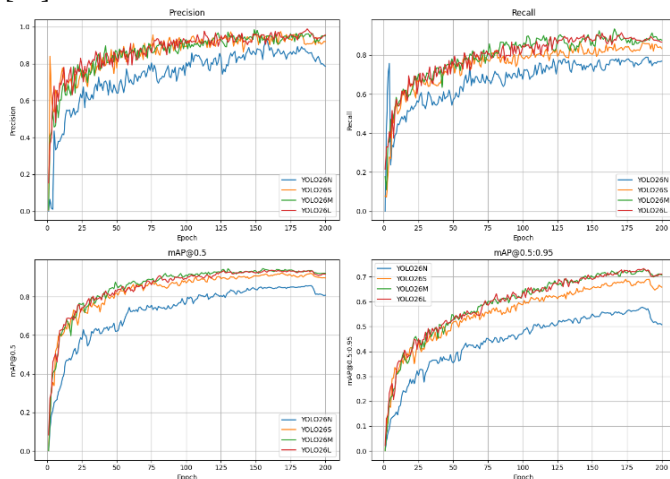


Figure 6. Evaluation Metric Curves of each YOLO26 Variant

B. Comparison with Baseline Object Detection Models

In order to find out whether the better performance was due to the YOLO26 architecture itself or rather to a general feature of more recent object detection models, YOLO26n was compared with three baseline models of the same size: YOLOv8n, YOLO11n, and YOLO12n. The models were trained and evaluated using the same dataset and training settings. The nano-scale version was chosen for this comparison to make sure that the evaluation was fair and matched in size.

TABLE VIII
COMPARISON WITH BASELINE OBJECT DETECTION MODELS

Model	Precision	Recall	$mAP@0.5$	$mAP@0.5:0.95$
YOLOv8n	0.904	0.703	0.869	0.557
YOLO11n	0.902	0.717	0.858	0.554
YOLO12n	0.895	0.781	0.847	0.564
YOLO26n	0.910	0.787	0.872	0.575

Table 8 shows that YOLO26n obtained the highest scores in all four metrics of the nano-scale models considered, having a Recall improvement of 0.6 to 8.4 percentage points and an $mAP@0.5:0.95$ improvement of 1.1 to 2.1 percentage points as compared to the baseline architectures. Even though the performance difference between YOLO26n and the more recent baseline models is relatively small, the fact that all the metrics showed consistent improvement suggests that the gains seen are indeed attributable, at least in part, to the architectural improvements specific to YOLO26. This result justifies the decision to select YOLO26 as the main architecture studied in this study.

C. Environmental Condition-Based Evaluation

To further examine model robustness, the detection performance of each YOLO26 variant was separately assessed under four environmental conditions: clear, evening, night, and rain. Condition-based evaluation is important because aggregate performance metrics may obscure differences in model behavior across specific environmental conditions, particularly when the number of images or pothole instances varies between conditions. By giving the results separately, it is possible to more clearly examine the detection ability of each model in different lighting and weather situations, at the same time lowering the possible effect of the sample distribution on the general evaluation. Even though the dataset has a fairly balanced representation across the four environmental conditions, this kind of separate evaluation offers a more detailed examination of model robustness and makes sure that good overall performance is not just due to better results in one particular condition. The results are presented in Table 9.

TABLE IX
PERFORMANCE UNDER DIFFERENT ENVIRONMENTAL CONDITIONS

Model	Condition	Precision	Recall	mAP@0.5	mAP@0.5:0.95
YOLO26n	Clear	0.90	0.78	0.87	0.58
	Evening	0.91	0.76	0.89	0.64
	Night	0.88	0.84	0.87	0.60
	Rain	0.95	0.69	0.78	0.48
YOLO26s	Clear	0.95 (+0.05)	0.83 (+0.05)	0.92 (+0.05)	0.70 (+0.12)
	Evening	0.89 (-0.02)	0.92 (+0.16)	0.94 (+0.05)	0.68 (+0.04)
	Night	0.98 (+0.10)	0.87 (+0.03)	0.91 (+0.04)	0.72 (+0.12)
	Rain	0.97 (+0.02)	0.80 (+0.06)	0.91 (+0.13)	0.65 (+0.17)
YOLO26m	Clear	0.93 (-0.02)	0.89 (+0.04)	0.96 (+0.04)	0.76 (+0.05)
	Evening	0.96 (+0.07)	0.88 (-0.04)	0.94	0.73 (+0.06)
	Night	0.98	0.89 (+0.02)	0.88 (-0.03)	0.72
	Rain	0.97	0.91 (+0.11)	0.94 (+0.03)	0.68 (+0.03)
YOLO26l	Clear	0.96 (+0.03)	0.87 (-0.02)	0.93 (+0.03)	0.75 (-0.01)
	Evening	0.98 (+0.02)	0.92 (+0.04)	0.95 (+0.01)	0.77 (+0.04)
	Night	0.95 (-0.03)	0.87 (-0.02)	0.90 (+0.02)	0.73 (+0.01)
	Rain	0.95 (-0.02)	0.93 (+0.02)	0.95 (+0.01)	0.69 (+0.01)

All models performed well under clear conditions, with mAP@0.5 values ranging from 0.87 to 0.96. YOLO26m reached the highest mAP@0.5 of 0.96, while YOLO26l had the highest Precision of 0.96. Recall ranged from 0.78 to 0.89 across the variants. Figure 7 shows that the radar profiles across all models are relatively similar, indicating consistent performance under clear conditions, with only moderate differences as model size increased. Good lighting and clear pavement appearance facilitated pothole feature extraction across all variants.

Performance under evening conditions showed greater variation among the YOLO26 variants. YOLO26n had the lowest Recall of 0.76 and mAP@0.5:0.95 of 0.64, while the larger variants reached higher values, with YOLO26l achieving 0.77 for mAP@0.5:0.95. Figure 7 highlights the relatively lower performance of YOLO26n in evening conditions, while the larger variants showed broader profiles. Reduced light and changes in natural light may have impacted the smaller YOLO26n variant more significantly, while the larger variants maintained more stable performance.

Performance under night conditions was similar across all YOLO26 variants. Precision ranged from 0.88 to 0.98, while Recall stayed between 0.84 and 0.87. The mAP@0.5:0.95 values were close, ranging from 0.60 for YOLO26n to 0.73 for YOLO26l. Figure 7 presents overlapping radar profiles, indicating only a slight performance improvement with increasing model size. Reduced lighting, headlight glare, and limited pavement texture may have made it more challenging to distinguish pothole edges [7].

Rain conditions posed the greatest challenge for precise localization across all YOLO26 variants, as reflected by the lowest mAP@0.5:0.95 values. Yet YOLO26m and YOLO26l still managed to show relatively good performance, with mAP@0.5 values of 0.94 and 0.95 and mAP@0.5:0.95 values of 0.68 and 0.69 respectively. Their Recall values also stayed high at 0.91 and 0.93, which means that the majority of

potholes could still be detected even though visibility was reduced and there were visual disturbances caused by the rain. Figure 7 shows the increase in radar performance from YOLO26n to YOLO26l, while the overall performance was more limited compared to other conditions. Rainwater, surface reflections, and reduced pavement contrast may have hindered model performance.

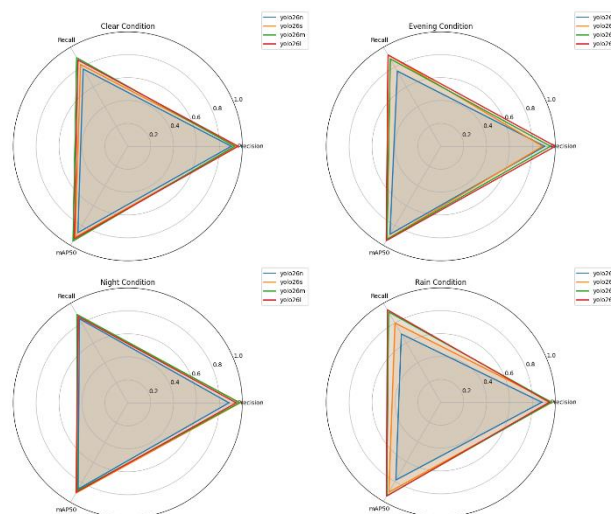


Figure 7. Radar Chart Comparison Across Environmental Conditions

In all four conditions YOLO26n showed the greatest decrease in performance when compared to clear conditions, especially during the evening and at night, while YOLO26m and YOLO26l kept their Precision, Recall, and mAP@0.5:0.95 more or less unchanged. This shows that the robustness benefit of the larger models is not just due to their higher overall scores but also results from a smaller difference in performance between good and bad conditions, which confirms their suitability for use in situations with variable lighting and weather.

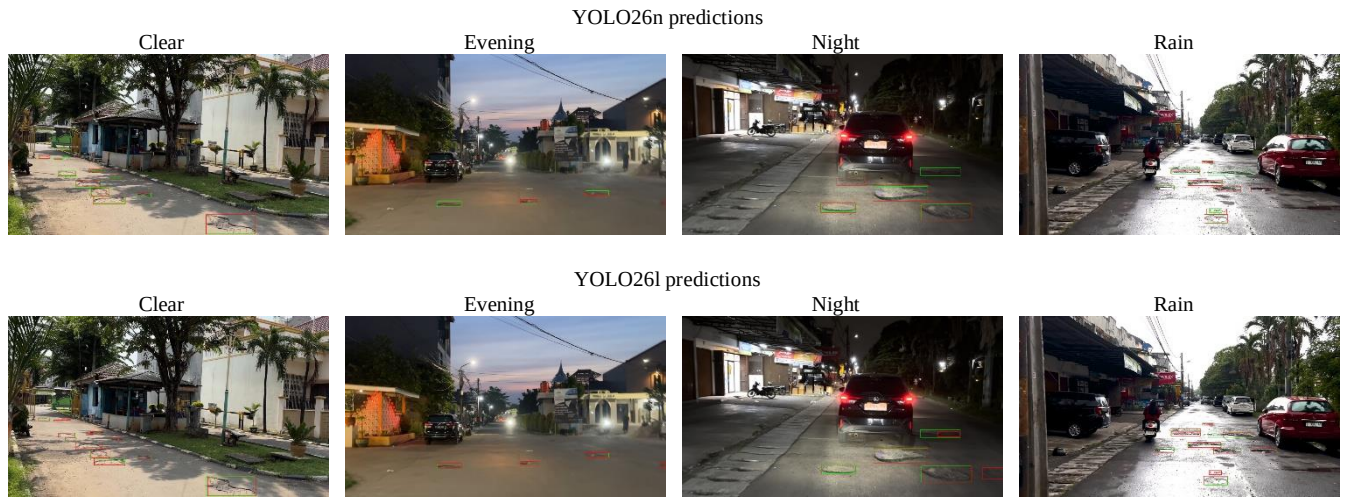


Figure 8. Ground truth vs Predicted Bounding Box: Visual Comparison of Pothole Detection under Four Environmental Conditions

The qualitative detection results for the four environmental conditions are given in Figure 8. The ground-truth and the bounding boxes predicted by YOLO26n and YOLO26l for pothole detection are compared under clear, rainy, evening, and night conditions, with the ground-truth boxes shown in green and the predicted ones in red. Overall, both YOLO26 variants successfully detected potholes across different conditions. Detection was generally more consistent under clear and evening conditions, while night and rain conditions showed reduced localization accuracy due to limited lighting and less visible pavement texture.

D. Qualitative Error Analysis

In order to look more closely at the other detection errors, a qualitative analysis was carried out on some of the false-positive (FP) and false-negative (FN) predictions shown in Figure 9 and Figure 10. In the figures in question, the green bounding boxes show the ground-truth annotations, and the red bounding boxes are the ones predicted by the model. False-negative (FN) cases are those in which pothole instances are present in the ground-truth annotations but have not been detected by the model, while false-positive (FP) cases are cases where the model wrongly identifies a non-pothole area as a pothole.



Figure 9. Representative False-Negative Cases in Pothole Detection

False-negative cases were most frequently seen in situations where the potholes had imprecise visual boundaries

because of surface noise, for example from scattered stones, together with shadow interference. Potholes which were relatively far from the camera were also more likely to be overlooked since their visual appearance became smaller and less distinguishable from the surrounding road surface. These results show that potholes having ambiguous boundaries, low visual contrast, or a small apparent size still pose a challenge for the detector to recognize accurately.



Figure 10. Representative False-Positive Cases in Pothole Detection

False-positive detections were mainly linked to objects that had visual features similar to potholes; for instance, dry leaves on the road surface were at times wrongly identified as potholes and in one case dark-brown motorcycle handlebars were also detected as potholes. Moreover, overlapping or duplicated bounding boxes were found in a number of the predictions, leading to false-positive detections. These mistakes indicate that the model might still confuse potholes with irregular patterns on the road surface or with dark-colored objects, especially in cases where their appearance mimics the visual texture and color distribution of potholes.

E. Computational Efficiency Evaluation

While detection accuracy matters for finding potholes, computational efficiency is equally important in determining whether a model can be deployed effectively in real-time driver assistance systems. A model with high detection accuracy may still be impractical for real-time applications if it requires excessive computational resources or produces low

inference speeds. Therefore, evaluating computational efficiency is necessary to assess the trade-off between detection performance and resource requirements. Table 10 compares the computational characteristics of each YOLO26 variant.

TABLE X
COMPUTATIONAL EFFICIENCY COMPARISON OF YOLO26 VARIANTS

Model	Parameter (M)	Model Size (MB)	Train Time (h)	Memory (GB)	Inference (ms)	FPS
YOLO26n	2.38	5.13	0.517	2.58	16.4	60.8
YOLO26s	9.47	19.37	0.578	5.21	16.9	59.2
YOLO26m	20.35	41.98	1.264	9.07	20.7	48.2
YOLO26l	24.75	50.52	1.601	11.40	21.2	47.1

As model complexity increased from YOLO26n to YOLO26l, the computational requirements generally increased. The number of parameters increased from 2.38 M to 24.75 M, accompanied by an increase in model size from 5.13 MB to 50.52 MB, training time from 0.517 to 1.601 hours, and GPU memory usage from 2.58 to 11.40 GB.

Inference latency also tended to increase for the larger models, with YOLO26n achieving the fastest inference at 16.4 ms (60.8 FPS), only minor variations observed between YOLO26n and YOLO26s, making YOLO26n a great choice for real-time use on devices with limited resources. In contrast, YOLO26l processed images at 47.1 FPS, which remains sufficient for real driving applications while achieving the best detection accuracy. These results illustrate the trade-off between detection performance and computational efficiency. YOLO26l required roughly 10.4 times more parameters than YOLO26n and took over three times longer to train, yet only improved mAP@0.5:0.95 by 0.16 points from 0.575 to 0.735. This corresponds to a substantially higher accuracy-per-parameter ratio for YOLO26n, confirming that its competitiveness reflects a measurable efficiency advantage rather than a general impression. Meanwhile, YOLO26m achieved competitive detection performance with a much lower computational cost than YOLO26l, suggesting a better balance between accuracy and efficiency for real-time pothole detection.

Overall, the results demonstrate a trade-off between model capacity, computational resource requirements, and detection performance. Therefore, the right YOLO26 variant should be chosen based on the intended usage platform. YOLO26n is better for lightweight edge devices that need fast inference speed, while YOLO26l is suited for applications where detection accuracy is the priority. Among the models tested, YOLO26m strikes a good balance between detection performance and computational efficiency, making it a strong candidate for future intelligent pothole warning systems.

Experimental results indicate that boosting the model's capacity from YOLO26n to YOLO26l consistently improves

localization performance. The larger models, with deeper feature extraction networks and greater representational ability, can better capture complex pavement textures and subtle pothole edges. This improvement is reflected in the higher mAP@0.5:0.95 values, which indicate that more complex models improve localization precision more than just object recognition. Previous studies on YOLO-based pothole detection have shown similar findings, where larger network architectures led to better detection accuracy despite higher computational costs [21].

Environmental conditions pose a significant challenge for vision-based pothole detection since they directly impact the visible features available for extraction. Reduced illumination under evening and night affects detection performance, especially for the smaller YOLO26n variant during evening conditions. However, the performance of all variants becomes fairly similar at night. Rain is the biggest challenge for precise localization, resulting in the lowest mAP@0.5:0.95 values across all variants, although larger models still show improvement. These findings agree with earlier studies indicating that poor lighting is a key factor limiting pothole detection performance [5], [7].

This study used a locally collected driver's perspective dataset representing road conditions in the Poris area of Tangerang. Many earlier studies relied on close-up pothole images or datasets collected in different environments. A driver's perspective view provides a more accurate reflection of the visual information available when driving. Thus, it offers a realistic benchmark for assessing vision-based pothole early-warning systems. While the dataset does not cover all Indonesian road conditions, it is an important step toward developing computer vision systems using locally sourced driving scenes.

Despite the promising results, this approach has several limitations. To begin with, the dataset was gathered in a single geographic area, which limits environmental variety and thus reduces the generality of the findings. Although the dataset covers clear, rain, evening, and night conditions, the environmental variations under each condition were not systematically controlled or classified. For instance, the images obtained include light rain, wet road surfaces, water puddles in potholes, and shadows in clear conditions; yet the different levels of rain intensity, the glare from vehicle headlights, and the various camera viewing angles were not specifically examined as separate scenarios relating to robustness. The dataset was extended by using horizontal flipping, which produced 168 augmented images from the original 200. Although this increased the number of training samples, the augmented images did not add any new road scenes or environmental conditions and so the overall visual diversity of the dataset remains limited. A possible drawback of this study is that the dataset was derived from continuous driving videos. Even though frames were taken at 1 FPS to decrease the temporal similarity between successive images, future research should use sequence-level or video-level dataset splitting and include an independent test set obtained

from previously unseen driving sequences in order to further reduce any possible data leakage and to give a more rigorous assessment of the model's generalization. Another limitation relates to the evaluation of real-time performance. The inference speed and frames per second were measured on the testing device and so may not reflect the models' performance on resource-constrained edge or in-vehicle devices. Testing on platforms such as embedded computing devices or smartphones was not within the scope of this study.

The dataset should therefore be extended in future research by including independent images that cover a wider variety of weather conditions, road surfaces, illumination levels, viewing angles, and geographic locations in Indonesia. This kind of data would allow for a more systematic evaluation of robustness in difficult situations and lead to a better assessment of the model's generalization capabilities. It is also necessary in future research to carry out hardware-specific benchmarking and field testing in order to evaluate the system's real-time performance under real driving conditions.

IV. CONCLUSION

The study assessed the performance of four variants of YOLO26 in the context of real-time pothole detection, using a dataset that had been collected locally from the driver's point of view under four different conditions: clear, rain, evening, and night. The results indicate that greater model complexity generally results in better detection performance, especially in terms of localization accuracy, at the expense of increased computational cost. YOLO26m obtained the highest Recall (0.877) and mAP@0.5 (0.934), whereas YOLO26l achieved the highest Precision (0.989) and mAP@0.5:0.95 (0.735). Yet YOLO26n offered the fastest inference speed, needed the least GPU memory, and had the lowest computational cost, which makes it suitable for use on resource-constrained edge devices. YOLO26m should be chosen when it is important to have high recall and overall detection performance, as it provides the highest Recall and mAP@0.5 while maintaining a speed of 48.2 FPS. YOLO26l is more appropriate for applications in which precision and maximum localization accuracy are prioritized since it achieves the highest Precision and mAP@0.5:0.95, even though it has higher computational requirements and a lower inference speed of 47.1 FPS. The findings thus offer practical guidelines for selecting a model depending on the requirements of the intended deployment scenario.

The environmental assessment shows that lighting has a major influence on the performance of pothole detection, night-time conditions always being more difficult than those at clear or evening times. The dataset gathered by local drivers offers a realistic standard for vision-based pothole detection when vehicles are actually being driven. Yet the lack of variety in geographic and environmental terms limits the ability of the results to be applied generally. The findings constitute an offline evaluation which can form the foundation for further work on intelligent driving assistance and pothole early-warning systems. Still, the results on their

own do not prove that the systems are ready for use in real vehicles. Future research must test the models on the target edge or within the vehicle's hardware and carry out field tests under real driving conditions in order to better evaluate their practical usefulness.

REFERENCES

- [1] D.-I. Dumitrascu, "Influence of Road Infrastructure Design over the Traffic Accidents: A Simulated Case Study," *Infrastructures (Basel)*, vol. 9, no. 9, p. 154, Sep. 2024, doi: 10.3390/infrastructures9090154.
- [2] A. Abed, M. Rahman, N. Thom, D. Hargreaves, L. Li, and G. Airey, "Analysis and Prediction of Pothole Formation Rate Using Spatial Density Measurements and Pavement Condition Indicators," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2677, no. 11, pp. 651–664, Nov. 2023, doi: 10.1177/03611981231166684.
- [3] S. K. Satti, G. N. V. Rajareddy, K. Mishra, and A. H. Gandomi, "Potholes and traffic signs detection by classifier with vision transformers," *Sci. Rep.*, vol. 14, no. 1, p. 2215, Jan. 2024, doi: 10.1038/s41598-024-52426-4.
- [4] M. Ling *et al.*, "Nighttime Pothole Detection: A Benchmark," *Electronics (Basel)*, vol. 13, no. 19, p. 3790, Sep. 2024, doi: 10.3390/electronics13193790.
- [5] M. Jakubec, E. Lieskovska, B. Bucko, and K. Zabolovska, "Pothole detection in adverse weather: leveraging synthetic images and attention-based object detection methods," *Multimed. Tools Appl.*, vol. 83, no. 39, pp. 86955–86982, 2024, doi: 10.1007/s11042-024-19723-6.
- [6] M. Jakubec, E. Lieskovská, B. Bučko, and K. Zabolovská, "Comparison of CNN-Based Models for Pothole Detection in Real-World Adverse Conditions: Overview and Evaluation," *Applied Sciences (Switzerland)*, vol. 13, no. 9, 2023, doi: 10.3390/app13095810.
- [7] A. Abu-raddaha, Z. A. El-Shair, and S. Rawashdeh, "Leveraging Perspective Transformation for Enhanced Pothole Detection in Autonomous Vehicles," *J. Imaging*, vol. 10, no. 9, 2024, doi: 10.3390/jimaging10090227.
- [8] M. Khanmohamadi and M. Guerrieri, "Advanced Sensor Technologies in CAVs for Traditional and Smart Road Condition Monitoring: A Review," *Sustainability*, vol. 16, no. 19, p. 8336, Sep. 2024, doi: 10.3390/su16198336.
- [9] P. Hiskiawan, T. P. Wijayanti, H. Heryanto, S. Everlin, S. A. Yasodhara, and D. Alexander, "Mel-Frequency Cepstral Coefficients and Neural Networks for Indonesian Traditional Music Recognition," in *2025 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, Dec. 2025, pp. 1707–1712. doi: 10.1109/ICIMCIS68501.2025.11327436.
- [10] P. Hiskiawan, Y. M. Geasela, H. Heryanto, E. Stephanie, M. Ardianti, and F. A. Sukarno, "Trustworthy Data Science Framework for Non-Invasive Nutritional Screening Using Computer Vision," in *2025 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, Dec. 2025, pp. 1743–1748. doi: 10.1109/ICIMCIS68501.2025.11327268.
- [11] P. Hiskiawan, M. K. Sari, R. E. Siregar, N. A. Valencia, and T. P. Wijayanti, "A Deep Learning Data Fusion Approach for Prostate MRI Zonal Segmentation Using Dual-Channel U-Net with T2 and ADC Images," in *2026 International Conference on Current Research in Artificial Intelligence and Data Science (ICCRAIDS)*, Apr. 2026, pp. 1–7. doi: 10.1109/ICCRAIDS67816.2026.11519671.
- [12] Y. Safyari, M. Mahdianpari, and H. Shiri, "A Review of Vision-Based Pothole Detection Methods Using Computer Vision and Machine Learning," *Sensors*, vol. 24, no. 17, p. 5652, Aug. 2024, doi: 10.3390/s24175652.

- [13] A. Vijayakumar and S. Vairavasundaram, "YOLO-based Object Detection Models: A Review and its Applications," *Multimed. Tools Appl.*, vol. 83, no. 35, pp. 83535–83574, Mar. 2024, doi: 10.1007/s11042-024-18872-y.
- [14] R. Sapkota *et al.*, "YOLO advances to its genesis: a decadal and comprehensive review of the You Only Look Once (YOLO) series," *Artif. Intell. Rev.*, vol. 58, no. 9, p. 274, Jun. 2025, doi: 10.1007/s10462-025-11253-3.
- [15] M. L. Ali and Z. Zhang, "The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection," *Computers*, vol. 13, no. 12, p. 336, Dec. 2024, doi: 10.3390/computers13120336.
- [16] R. C. Subianto, M. Naufal, and F. Alzami, "Improving YOLO12 Performance Using Efficient Channel Attention For Ship Object Detection," *Journal of Applied Informatics and Computing*, vol. 10, no. 3, pp. 2710–2722, Jun. 2026, doi: 10.30871/jaic.v10i3.13067.
- [17] N. K. Hamzidah *et al.*, "AI-YOLO Based Smart Laboratory Security for Automated Face Recognition and Suspicious Activity Detection," *Journal of Applied Informatics and Computing*, vol. 10, no. 1, pp. 440–450, Feb. 2026, doi: 10.30871/jaic.v10i1.11936.
- [18] R. F. Puspita, M. Naufal, and F. Al Zami, "Improving YOLO Performance with Advanced Data Augmentation for Soccer Object Detection," *Journal of Applied Informatics and Computing*, vol. 9, no. 6, pp. 3601–3611, Dec. 2025, doi: 10.30871/jaic.v9i6.11256.
- [19] L. K. Ramadhani and B. N. Widyaningrum, "Implementation of YOLO v11 for Image-Based Litter Detection and Classification in Environmental Management Efforts," *Journal of Applied Informatics and Computing*, vol. 9, no. 3, pp. 617–624, Jun. 2025, doi: 10.30871/jaic.v9i3.9213.
- [20] N. Mery Handayani and Y. Azhar, "Implementation of YOLOv11 for Detection and Identification of Strawberry Ripeness Stages," *Journal of Applied Informatics and Computing*, vol. 10, no. 2, pp. 1210–1219, Apr. 2026, doi: 10.30871/jaic.v10i2.12283.
- [21] M. H. Asad, S. Khaliq, M. H. Yousaf, M. O. Ullah, and A. Ahmad, "Pothole Detection Using Deep Learning: A Real-Time and AI-on-the-Edge Perspective," *Advances in Civil Engineering*, vol. 2022, 2022, doi: 10.1155/2022/9221211.
- [22] S. Parvin, F. Munsy, M. T. Rahat, A. Nahar, K. Nur, and D. Ghose, "Robust multi-weather pothole detection: An enhanced YOLOv9 trained on the MWPD dataset," *Results in Engineering*, vol. 28, p. 107817, Dec. 2025, doi: 10.1016/j.rineng.2025.107817.
- [23] S. S. Reddy, I. U. Khan, and others, "Weather and monsoon resilient pothole detection: A YOLO based real-time application for diverse road conditions," *SGS-Engineering & Sciences*, vol. 1, no. 1, 2025.
- [24] M. Tórnero-Soria, A.-J. Sánchez-Salmerón, and E. Vendrell Vidal, "Toward a Deeper Understanding of YOLO26: Block-Level Architectural Analysis and Ablation Studies," *Applied Sciences*, vol. 16, no. 13, p. 6758, Jul. 2026, doi: 10.3390/app16136758.
- [25] B. Bučko, E. Lieskovská, K. Záborská, and M. Záborský, "Computer Vision Based Pothole Detection under Challenging Conditions," *Sensors*, vol. 22, no. 22, p. 8878, Nov. 2022, doi: 10.3390/s22228878.
- [26] A. K. Alsharafi and M. N. Osman Zahid, "Investigation on a Vision-Based Approach For Smart Pothole Detection Using Deep learning Based on Fast CNN," *MEKATRONIKA*, vol. 5, no. 2, pp. 87–99, Dec. 2023, doi: 10.15282/mekatronika.v5i2.9813.
- [27] R. Kaur and S. Singh, "A comprehensive review of object detection with deep learning," *Digit. Signal Process.*, vol. 132, p. 103812, Jan. 2023, doi: 10.1016/j.dsp.2022.103812.
- [28] T. Kiobya, J. Zhou, B. Maiseli, and M. Khan, "Hybrid Intersection Over Union Loss for a Robust Small Object Detection in Low-Light Conditions," *IEEE Access*, vol. 13, pp. 12321–12331, 2025, doi: 10.1109/ACCESS.2025.3530089.
- [29] M. M. Alam, "Towards Precise Localization in Object Detection: Stable Matching Strategies and Advanced Bounding Box Regression Loss Functions in Deep Neural Networks," University of New South Wales (Australia), 2025. doi: <https://doi.org/10.26190/unsworks/31743>.
- [30] P. Hiskiawan, C.-C. Chen, and Z.-K. Ye, "Processing of electrical resistivity tomography data using convolutional neural network in ERT-NET architectures," *Arabian Journal of Geosciences*, vol. 16, no. 10, pp. 1–14, 2023, doi: 10.1007/s12517-023-11690-w.