Deep Learning-Based Road Damage Detection Using Improved YOLOv8: Model Performance and Implementation

Aulia Rahman[1], Hwa Jen Yap[2], Rusdha Muharar[3]
Department of Electrical and Computer Engineering, Universitas Syiah Kuala, Banda Aceh, Indonesia[1,3]
Faculty of Engineering-Department of Mechanical Engineering, Universiti Malaya, Kuala Lumpur, Malaysia[2]

***Abstract*—Road infrastructure monitoring plays a crucial role in ensuring safety and economic efficiency; however, conventional manual inspection methods are expensive and resource-intensive. This study presents the development and evaluation of a real-time road damage detection system using the improved YOLOv8 architecture. Building upon the strengths of previous YOLO models, YOLOv8 offers enhanced accuracy and inference speed, making it highly suitable for mobile deployment. The model was trained to identify six common road damage types in Indonesia: potholes, alligator cracks, transverse cracks, longitudinal cracks, edge cracks, and road joints. Utilizing a hybrid dataset of 2,946 images, combining locally collected data and the RDD2020 dataset, the system incorporates mosaic augmentation and optimized preprocessing to improve generalization. The optimized YOLOv8 model achieved a mean Average Precision (mAP@50) of 96.3%, an F1-score of 91%, and an overall accuracy of 91%, demonstrating superior detection and classification performance. The system was deployed as a user-friendly smartphone application, enabling automated, geo-tagged road condition surveys and offering road authorities a scalable, efficient, and practical tool for infrastructure monitoring.**

***Keywords*—Road damage detection; YOLOv8; deep learning; computer vision; mobile application**

I. Introduction

Road infrastructure is a crucial public asset that requires regular monitoring because of its significant role in fostering economic development [1], [2]. However, road surfaces inevitably deteriorate over time because of both natural and human influences. In Indonesia, motorists often encounter damaged roads resulting from extreme weather, heavy traffic loads, substandard construction quality, and insufficient maintenance. These conditions frequently lead to accidents, some of which are fatal, as drivers struggle to anticipate and navigate road defects. Damaged roads also disrupt transportation, increasing vehicle operating costs because of damage from potholes and other defects [3]. To mitigate accident risks and enhance transportation efficiency, monitoring road conditions is essential [4].

Currently, road damage surveys are conducted manually by physically marking damaged areas by inspectors. However, this approach is labor-intensive, time-consuming, and impractical for extensive road networks [5]. Continuous monitoring is challenging due to the vastness scope of road infrastructure. To address this, automated solutions utilizing computer vision and deep learning are increasingly being employed for road damage monitoring [6], [7], [8].

Previous studies have demonstrated the use of YOLO models for road damage detection, as seen in [9], [10], [11]. More recently, YOLOv5, using a 640×640 image size, achieved an F1 score of 66.51% with the RDD2022 dataset, detecting potholes, alligator cracks, transverse cracks, and longitudinal cracks [12].

Although previous studies have demonstrated the potential of YOLO-based approaches for road defect detection, a significant research gap remains in the development of models adapted to local road conditions and a wider variety of defect categories. Many publicly available datasets do not include classes such as edge cracks and road joints, despite these defects being common and important indicators of pavement condition in countries like Indonesia.

This study aims to fill the existing gap by developing and implementing a robust road damage detection system using improved YOLOv8. The primary goal was to create a model that not only accurately identifies common cracks and potholes but also detects two additional categories—edge cracks and road joints—using a novel dataset from Indonesian roads. The key scientific contributions of this study are as follows: 1) the creation of a comprehensive and locally-relevant dataset for road damage; 2) the development of a highly accurate YOLOv8 detection model, achieving an F1-score of 91% F1-score, which surpasses previous benchmarks through tailored data processing and augmentation; and 3) the successful integration of this model into an on-device smartphone application, offering a practical solution for automated infrastructure monitoring.

II. Related Works

Object detection is a branch of computer vision that focuses on locating and identifying objects within images or videos. The fundamental components of object detection consist of three main elements: bounding box, class label, and confidence score [13], [14]. Bounding boxes are utilized to pinpoint the positions of objects in an image. The class label specifies the type of object detected, whereas the confidence score reflects the certainty level that the detected object corresponds to the predicted class [15].

Recent advancements in deep learning have significantly enhanced the performance of object detection models. Two notable architectures, the Faster R-CNN [16] and YOLO (You Only Look Once) [17], exemplify key approaches to object detection. The Faster R-CNN employs a two-stage process:

generating region proposals followed by classification, whereas YOLO is a single-stage detector that predicts bounding boxes and class labels in one forward pass, making it ideal for real-time applications. Although two-stage detectors can achieve high accuracy, their slower inference speeds render them less suitable for on-the-fly detection from a moving vehicle. These methods greatly enhance the speed and accuracy of object detection tasks, offering powerful tools for various applications, including autonomous driving and surveillance. The next section discusses the building blocks of object detection used in this research: convolutional neural networks (CNNs) and YOLOv8.

A. Convolutional Neural Network

Convolutional neural networks (CNNs) are a type of artificial neural network that are frequently employed for processing image data [18], [19]. At the heart of CNNs is the convolution operation, which is essential to their operation and functions as a filter. These networks are composed of multiple layers that are specifically crafted to perform convolutions on image data. The training process of the model encompasses three primary stages: the convolutional layer, the pooling layer, and the fully connected layer. CNNs have become prominent because of their capacity to effectively capture spatial hierarchies within image data. Their architecture mirrors the structure of the human visual cortex, enabling them to focus on localized regions of an image through convolutional filters. As the network deepens, these filters progressively learn more intricate features, allowing CNNs to identify patterns, such as edges and textures, in the initial layers and more abstract patterns in the deeper layers [20]. Pooling layers reduce the dimensionality of the feature maps while preserving crucial information and minimizing computational costs. This makes CNNs highly effective for various image-based tasks, including classification, object detection, and segmentation [21].

Since AlexNet's success in the ImageNet competition in 2012, CNN architectures have undergone significant evolution. Models such as VGG [22], ResNet [23], and Inception [24] have enhanced both accuracy and computational efficiency. Furthermore, the adoption of transfer learning has broadened CNN applications, enabling models pretrained on extensive datasets to be fine-tuned for specific tasks. This approach boosts model performance, especially in situations where labeled data are scarce [25], [26]. The robustness and versatility of CNNs have made them indispensable not only in computer vision but also in other fields such as medical imaging, where they support disease diagnosis through the analysis of radiological data [27] and thermal images [28].

B. YOLOv8

YOLO, short for "You Only Look Once", has transformed object detection by approaching it as a single regression problem, thereby moving away from traditional sliding window techniques. By processing the entire image in one go, YOLO predicts bounding boxes and class probabilities in real-time, making it ideal for high-speed applications such as autonomous driving and surveillance [29]. Building on this foundation, YOLOv5 features a lightweight yet robust architecture that utilizes a CSPNet-based backbone for feature extraction and a path aggregation network (PANet) to enhance multi-scale

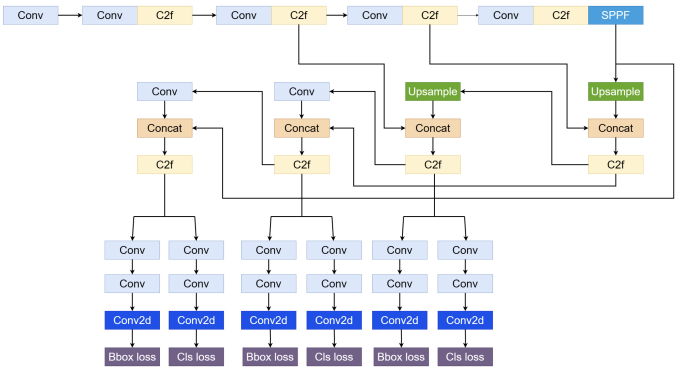


Fig. 1. Original YOLOv8 architecture

object detection. This combination of speed and accuracy has made YOLOv5 a favored choice for real-time object detection tasks across various domains. An application of YOLOv5 in real-time vehicle detection was conducted [30].

YOLOv8 is an advanced object detection architecture designed for real-time applications, which combines high inference speed with enhanced accuracy [31], [32]. It builds upon the YOLO series by utilizing a cross-stage partial network (CSPNet) backbone for efficient feature extraction, a neck component that refines features through additional convolutional layers, and a detection head that incorporates a feature pyramid network (FPN) to detect objects at multiple scales. This architecture enables precise detection and classification of objects, making YOLOv8 particularly suitable for tasks that require fast and accurate detection, such as road damage monitoring on mobile platforms.

This study introduces a road damage detection and classification system utilizing the YOLOv8 architecture, as shown in Fig. 1. This system was specifically designed to identify six common types of road damage found on Indonesian roads: potholes, alligator cracks, transverse cracks, longitudinal cracks, edge cracks, and road joints. The YOLOv8 architecture was selected for its exceptional balance between real-time inference speed and high detection accuracy, making it an ideal choice for deployment on a mobile platform for on-road monitoring. By leveraging the YOLOv8 model, this system aims to deliver accurate and efficient detection tailored to local road conditions.

C. Bidirectional Feature Pyramid Network (BiFPN)

The Bidirectional Feature Pyramid Network (BiFPN) represents a significant advancement in the design of the neck component for modern object detection architectures, particularly in improving efficient multi-scale feature fusion. Introduced as part of the EfficientDet family [33], BiFPN addresses the limitations of conventional Feature Pyramid Networks (FPN) that rely on static fusion strategies. It incorporates bidirectional connections (top-down and bottom-up) along with learnable weighting mechanisms, enabling the model to adaptively determine the relative importance of each input feature map. This approach effectively mitigates feature imbalance across scales and enhances detection performance, especially for small objects, while maintaining computational efficiency [34], [35].

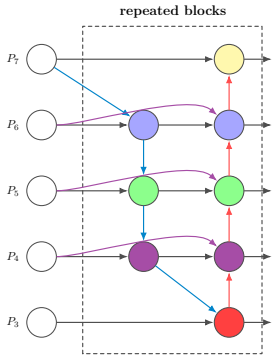


Fig. 2. BiFPN feature network design.

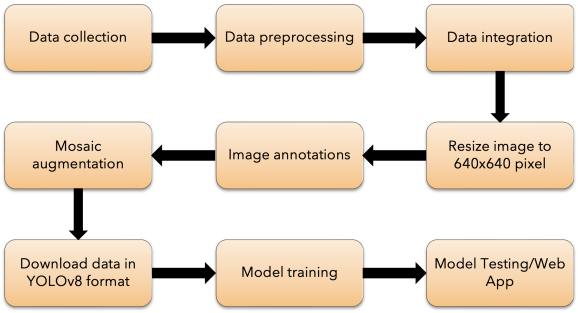


Fig. 3. Road damage detection and classification workflow.

From an architectural perspective, BiFPN operates by taking multiple feature maps from the backbone (e.g., levels P2–P5) and fusing them through nodes equipped with learnable, non-negative weights. These weights are activated using a ReLU function and normalized to form a convex combination, ensuring stable training while allowing the network to suppress less informative features. Additionally, BiFPN is designed with computational efficiency in mind by minimizing redundant connections and limiting the number of inputs per fusion node, thereby reducing FLOPs compared to traditional multi-scale fusion methods. The fusion process can also be iteratively repeated to progressively refine feature representations, which is particularly beneficial in scenarios with significant scale variation, such as UAV imagery or micro-defect detection [36] and surface flaw detection of a small automotive component [37]. This balance between efficiency and representational power has led to the widespread adoption and further development of BiFPN across various object detection frameworks. The BiFPN feature network design is shown in Fig. 2.

In this study, the original feature aggregation layers in the YOLOv8 neck were modified by replacing the conventional PAN-FPN structure with a BiFPN module, enabling more efficient fusion of low-level spatial features and high-level semantic features. This architectural modification improves the model's capability to detect road defects of varying sizes, particularly small and irregular crack patterns. Although the addition of BiFPN increases the number of feature fusion operations and introduces a slight computational overhead, the impact on inference speed remains manageable due to the lightweight design and optimized weighted connections of the BiFPN architecture.

D. Evaluation Metrics

Model performance is assessed using the Intersection over Union (IoU) score to measure bounding box accuracy and the mean Average Precision (mAP) to evaluate object detection accuracy [38]. The IoU score is determined by calculating the ratio of the overlap area between the predicted bounding box and the ground truth bounding box to the union area of these boxes. A higher IoU score indicates a closer alignment between the predicted and actual object locations.

The recognition accuracy was assessed using mAP, which integrates both precision and recall across various IoU thresholds, thereby offering a thorough evaluation of the model's performance under different conditions [39].

As illustrated in Fig. 3, this work comprises eight main steps: data collection, data cleaning, data integration, image resizing, image annotation, applying mosaic augmentation, downloading the dataset in the YOLOv8 PyTorch format, training the model, model testing, and application implementation.

E. Data Collection

The dataset for this study was independently gathered using a smartphone camera. To ensure high-quality images, the camera was stabilized by minimizing body movement and focusing on road damage under optimal lighting conditions. Data collection occurred along a national road in Aceh province in Indonesia, capturing various types of road damage. The dataset encompasses six categories of road damage: potholes, alligator cracks, transverse cracks, longitudinal cracks, edge cracks, and road joints. To balance the dataset, additional road images were sourced from the RDD2020 dataset, particularly from Japanese roads. The final dataset comprised 2,946 images, divided into 2,049 images for training, 600 for validation, and 294 for testing. The total number of images for each class is detailed in Table I.

TABLE I. ADDITIONAL LOCAL DATASET

Class	Amount
Pothole	451 images
Alligator crack	368 images
Longitudinal crack	327 images
Transverse crack	432 images
Edge crack	491 images
Road joints	491 images

F. Data Prepocessing

Dataset preprocessing, including data cleaning and data integration, is an essential step to ensure that the dataset used for analysis or model development is accurate, consistent, and complete. Its primary objective is to detect and correct errors, remove inconsistencies, and handle missing values. By improving the quality of the data, this process enhances the precision and reliability of the analysis or model, ultimately leading to more accurate and impactful results.

Following the completion of the data cleaning process, the subsequent step involves merging the independently collected data by the researchers with the RDD2020 dataset from Japan.

The combined dataset, which is crucial for the analysis, is presented in Table II.

TABLE II. RESULT OF MERGING DATASETS

Class	Independent dataset	RDD2020	Amount
Pothole	451 images	40 image	491 images
Alligator crack	368 images	123 images	491 images
Longitudinal crack	327 images	164 images	491 images
Transverse crack	432 images	59 images	491 images
Edge crack	491 images	-	491 images
Road joints	491 images	-	491 images
Total			2946 images

The subsequent step involves resizing all images to 640x640 pixels. This specific dimension was selected to offer a higher resolution, thereby enabling the model to capture more detailed features of objects, which is essential for enhancing detection accuracy, particularly for smaller objects. Moreover, employing larger image sizes during training allows the model to learn more intricate features, thereby improving its overall performance.

G. Image Annotation

The data annotation process is designed to deliver precise information regarding the location and category of objects within images by marking bounding boxes around them. During this stage, the dataset is imported into Roboflow, where the bounding box tool is employed to accurately outline the objects, as illustrated in Fig. 4.

Each object is encased in a polygon that outlines its shape, with class labels assigned based on the object type. The deep learning model then learns to identify objects by examining the patterns and features within these bounding boxes. Detailed and precise annotations greatly enhance the quality of the training data, thereby improving the model's capacity to detect and classify road damage. By ensuring that each object is accurately labeled and represented, the model is better prepared to generalize from the training data and perform effectively in real-world scenarios.

H. Data Augmentation

Data augmentation includes a variety of techniques designed to expand and enrich datasets for machine learning models, with mosaic augmentation being one of the most widely used approaches. The process begins with an input image, after which the mosaic technique applies several transformations, including flipping, rotation, shearing, brightness adjustment, contrast enhancement, and noise injection, before combining regions from multiple images into a single composite sample [40], [41]. The process of our mosaic augmentation procedure is illustrated in Fig. 5.

This technique is widely employed in training object detection models, particularly for computer vision tasks. Mosaic augmentation merges multiple images into a single composite training sample, thereby enabling the model to learn from a broader range of examples [42], [43], [44]. This approach was selected over simpler geometric augmentations, such as flips and rotations, to significantly boost model robustness. By combining four different images into a single training sample, mosaic augmentation exposes the model to a wider variety of object scales, backgrounds, and partial occlusions, thereby improving its ability to accurately detect small or partially hidden defects on complex road surfaces.

I. Experimental Environment

The experimental environment consisted of a Windows 10 operating system, equipped with an NVIDIA GeForce RTX 3090 graphics card featuring 24 GB of GDDR6X memory, an Intel i7-10700 CPU, and 32 GB of RAM. Before initiating the training process, it is crucial to meticulously adjust the hyperparameters to ensure that the model learns effectively and yields optimal results. The key settings included the number of epochs, batch size, optimizer, momentum, and learning rate, all of which were customized to suit the data and model characteristics.

This research develops a deep learning model using the YOLOv8 architecture to detect six types of road defects: potholes, alligator cracks, transverse cracks, longitudinal cracks, edge cracks, and road joints. This section explains the process of building the baseline model and refining it to optimize the hyperparameters, ultimately creating a robust model for detecting and classifying these road damage types. The process involves fine-tuning settings such as the number of epochs, batch size, optimizer, momentum, and learning rate, all of which are adjusted to align with the characteristics of both the dataset and the model. Furthermore, robustness tests were conducted to evaluate the model's ability to generalize across

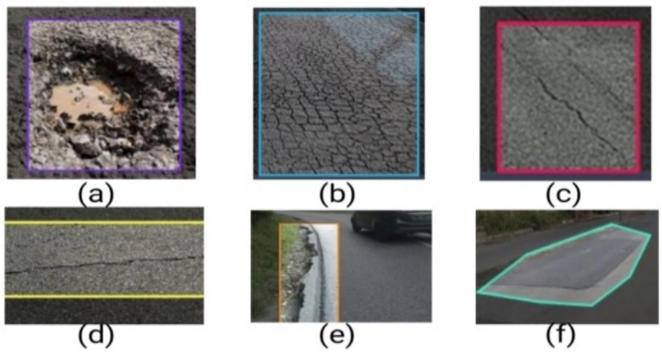


Fig. 4. Image annotation process: Creating a bounding box to enclose road damage.

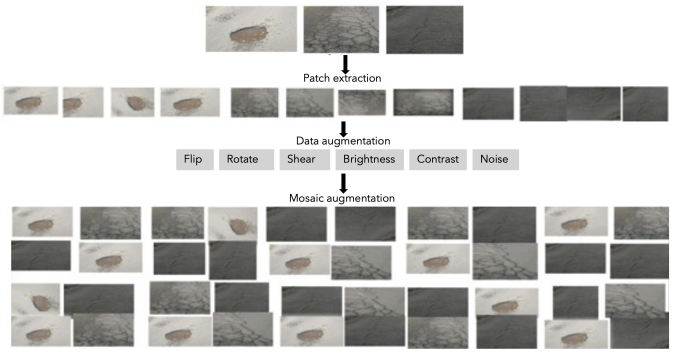


Fig. 5. Mosaic augmentation procedure.

different environmental conditions, such as varying lighting, road textures, and occlusions. The hyperparameters used in model training are shown in Table III.

TABLE III. HYPERPARAMETERS USED IN THE TRAINING PROCESS

Hyperparameter	YOLOv8
Epoch	200
Batch size	16
Optimizer	SGD
Learning rate	10^{-2}
Momentum	0.9
Image size	640x640

III. EXPERIMENTAL SETUP AND RESULTS

A. Training Results and Robustness Analysis

To evaluate the performance of the trained model, various metrics were examined, including loss curves, precision-recall metrics, confusion matrices, and mean Average Precision (mAP) scores. As shown in Fig. 6, the training loss curve starts at a high value but decreases significantly early in the training process. With an increasing number of epochs, the training loss continues to decline, indicating that the model is enhancing its predictions on the training data without significant overfitting. In contrast, the validation loss curve begins at a lower value but rises around epoch 75, likely due to variations in road damage patterns. However, with further training, the validation loss stabilizes and then decreases, suggesting that the model generalizes well to unseen data. The training of the model is illustrated in Fig. 6.

To further confirm the effectiveness of the model, we performed additional robustness tests by assessing its performance under various road surfaces and environmental conditions. Preliminary qualitative observations suggest that the model remains reasonably stable under moderate lighting variations and different road textures; however, comprehensive robustness testing under adverse environmental conditions remains future work.

Fig. 7 illustrates the precision and recall metric curves observed during training. Utilizing a learning rate of 10^{-2} alongside a mosaic augmentation strategy enabled the model to achieve an optimal balance between precision and recall. Both precision and recall stabilize around 0.9, demonstrating

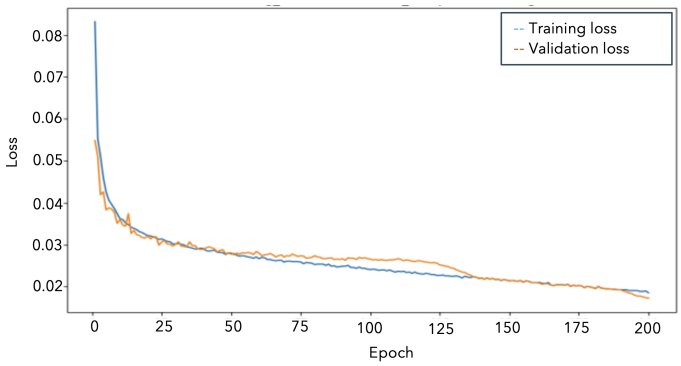


Fig. 6. Training and validation loss of the model.

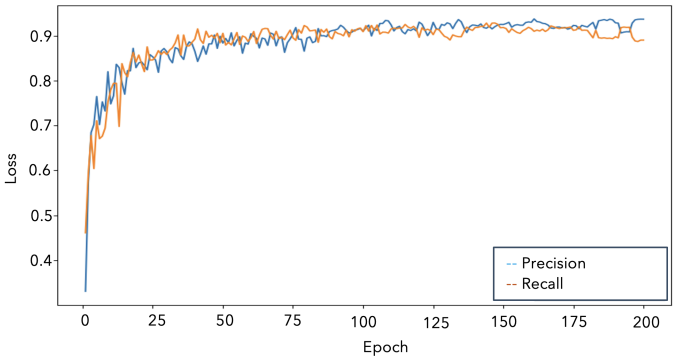


Fig. 7. Precision and recall of the model.

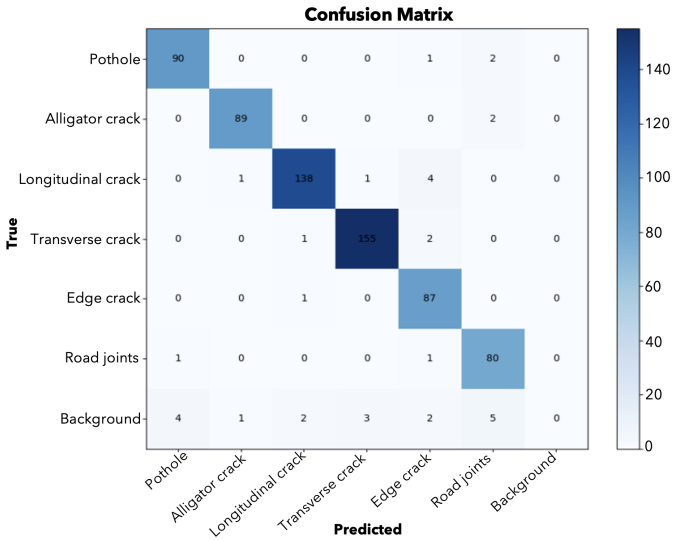


Fig. 8. Confusion matrix of model prediction.

the model's capability to accurately detect road damage while minimizing false positives.

The confusion matrix analysis, depicted in Fig. 8, illustrates the testing results using 294 enhanced images via mosaic techniques. The findings indicate that within the pothole class, 90 instances were accurately identified as potholes, while one was misclassified as an edge crack and two as road joints. In the alligator crack category, 89 were correctly recognized; however, two were misclassified as road joints. For the longitudinal crack class, 138 were correctly detected; one was misclassified as a transverse crack, and four transverse cracks were incorrectly identified as edge cracks. In the edge crack class, 87 instances were correctly classified; one was misclassified as a longitudinal crack. Regarding the road joint class, 80 predictions were accurate, while one was misclassified as a pothole and one as an edge crack. The confusion matrix underscores a significant improvement, as most predictions are correct, with only a few classification errors, demonstrating the model's robust performance.

Fig. 9 presents the F1-score results from the training process. The x-axis indicates the model's confidence level in its predictions, with higher confidence levels positioned on the right. The y-axis displays the F1 score, a metric that balances

precision and recall. The blue line (pothole) performs well, maintaining a high F1-score across most confidence levels. The orange line (transverse crack) begins with a high F1-score but drops significantly as confidence increases, likely owing to prediction errors, as the transverse crack class closely resembles the longitudinal crack class. The green line (longitudinal crack) demonstrates stable and consistent performance throughout. The red line (edge crack) also shows strong performance, consistently achieving a higher F1-score than the other classes.

The purple line, representing alligator cracks, demonstrates stable and robust performance. Similarly, the brown line, which denotes road joints, performs well, similar to the pothole class. The thick blue line illustrates the combined F1 score for all classes. At a confidence level of 0.556, this combined F1 score reaches 0.91, indicating excellent overall model performance.

Table IV presents the mAP scores for each class. At 0.5, the mAP value is outstanding, with the edge crack class achieving the highest score of 0.980, indicating the model's exceptional performance in detecting road damage with minimal error. This result indicates strong localization and classification performance at the IoU threshold of 0.5. with the ground truth box, showcasing high accuracy at this threshold. Furthermore, the mAP value at 0.5:0.95 also yields good results, with the edge crack class reaching a peak value of 0.698. Although performance slightly diminishes at higher IoU thresholds, the model continues to exhibit strong detection capabilities. The modest decline from mAP 0.5 to mAP 0.5:0.95 underscores the model's ability to manage tighter bounding box predictions, maintaining a balance between accuracy and precision.

TABLE IV. THE MAP VALUE OF EACH CLASS

Class	*Precision*	*Recall*	*mAP 0,5*	*mAP 0.5:0.95*
Pothole	0.916	0.941	0.976	0.676
Alligator crack	0.77	0.862	0.91	0.64
Transverse crack	0.702	0.683	0.708	0.589
Longitudinal crack	0.837	0.828	0.88	0.674
Edge crack	0.92	0.912	0.98	0.698
Road joints	0.734	0.805	0.921	0.692

We conducted the same experiment with the hyperparameters outlined in Table III to benchmark our proposed model against YOLOv3, YOLOv4, and YOLOv5. The performance of each model is presented in Table V, which demonstrates that YOLOv8 BiFPN exhibits superior performance.

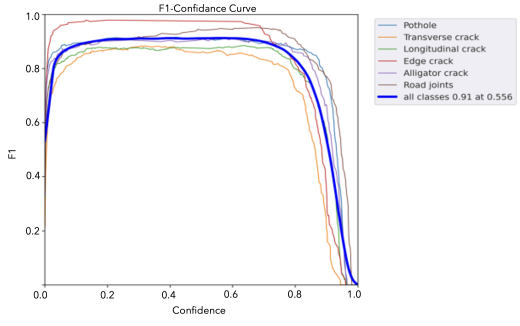
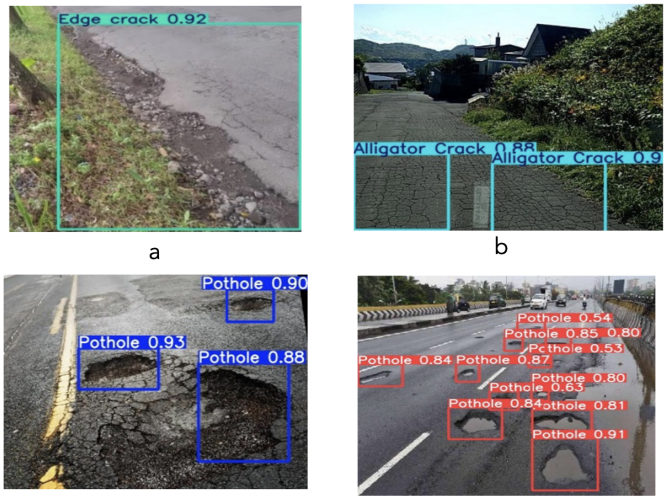


Fig. 9. F1-score of the model.



a b c d

Fig. 10. Road damage detection test: a) Edge crack, b) Alligator crack, c) Pothole, d) Multiple potholes.

TABLE V. PERFORMANCE COMPARISON AGAINST PREVIOUS YOLO ARCHITECTURES.

MODEL	Precision	Recall	mAP@0.5
YOLOv8 BiFPN	0.891	0.905	0.963
YOLOv8	0.873	0.893	0.955
YOLOv5	0.813	0.839	0.896
YOLOv4	0.793	0.818	0.88
YOLOV3	0.632	0.702	0.734

B. Detection Model Evaluation

Once the model for road damage detection and classification is trained, the subsequent step involves testing to evaluate its performance. This test determines how effectively the model can identify and categorize various types of road damage in previously unseen data. Additionally, it assesses the model's accuracy, precision, and recall across different predefined road damage classes. The results of the model testing are shown in Fig. 10.

C. Model Implementation on Smartphone

We developed a deep learning-based application utilizing the YOLOv8 architecture to detect and classify road defects into six primary categories. The trained model can accurately identify various defect types, including potholes, transverse cracks, longitudinal cracks, edge cracks, alligator cracks, and road joints. Owing to its real-time inference speed and high accuracy, this application enables road managers to perform automated inspections, thereby accelerating the damage detection process and enhancing the efficiency of road repair efforts.

The device was installed on the dashboard of a public passenger car. Our developed application captures images of the road surface approximately 10 m ahead, showcasing its capability to capture images while in motion without any data loss or duplication at an average speed of 40 km/h

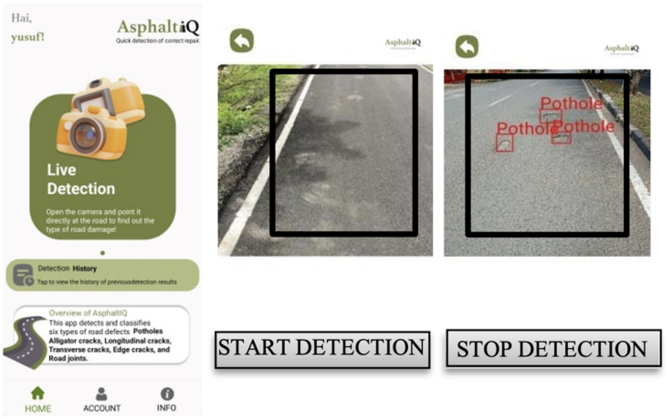


Fig. 11. On-device detection application on a smartphone.

(approximately 10 m/s), with one image captured per second. Furthermore, the application can accurately detect road defects within 1.5 s. A screenshot of the real-time detection application on a smartphone is shown in Fig. 11.

The operating screen of the smartphone application was mounted on the dashboard of a passenger car. The detection of road surface damage commenced when the "START DETECTION" button was activated. Images of the detected damage, along with their position information, were transmitted to the detection history in the database. Only images of confirmed damage were retained in the detection history.

The application features an image upload capability that enables users to submit photos of damaged roads. It automatically identifies and categorizes the type of damage depicted in an image. Powered by a trained YOLOv8 model, the application delivers fast and accurate detection results, simplifying the process for users to evaluate road conditions directly from uploaded images.

IV. Discussion

The performance of our YOLOv8 BiFPN model demonstrated a marked improvement over previous studies cited in the literature. For example, our model achieved an F1-score of 0.91, significantly surpassing the 50.39% reported by Opara [9] using YOLOv3 and the 66.51% using YOLOv5 on the RDD2022 dataset [12]. This significant improvement in performance can be attributed to several factors: the use of a tailored dataset that includes specific local road damage types (edge cracks and road joints), a larger and more balanced dataset, and the effective application of mosaic augmentation, which enhanced the model's ability to learn diverse features. The high mAP of 98% (at IoU=0.5) further confirms the model's accuracy in correctly detecting and classifying defects. The high mAP@50 should be interpreted as performance on the current test distribution rather than as universal generalization. The relatively small test set and visual similarity among locally collected road images may have contributed to the high score. Therefore, mAP@50:95 and future cross-region validation are reported/discussed to provide a more conservative assessment.

Despite these results, the study has several limitations. The performance of the model is highly dependent on environmental conditions, and its accuracy may decline in adverse weather, such as heavy rain or fog, or at night with insufficient street lighting. The training data were primarily collected from national roads in Indonesia, which may introduce a geographic bias. Consequently, the model may not perform as well on different types of road surfaces (e.g., concrete or unpaved roads) without additional training data. Furthermore, the tests were conducted at a moderate speed (30 km/h), and performance at significantly higher speeds has not yet been tested.

Because the local dataset was collected from a national road in Indonesia, the model may learn region-specific pavement textures, lane markings, lighting conditions, and defect appearances. Therefore, the current model should be regarded as optimized for the studied Local road context, and additional cross-region validation is required before wider deployment.

The proposed system demonstrated promising performance under daytime conditions; however, its robustness under heavy rain, fog, and nighttime illumination has not yet been comprehensively validated. Therefore, claims of real-world deployment suitability are limited to conditions similar to those represented in the test data.

The practical implications of this research are substantial. The successful implementation of the model into a smartphone application provides road management authorities with a practical and cost-effective tool for road infrastructure monitoring. This system can replace slow and labor-intensive manual inspections with rapid automated surveys. This enables authorities to quickly identify critical areas, prioritize repairs based on the severity and type of damage, and allocate maintenance budgets more efficiently.

V. Conclusion and Future Work

We successfully developed two additional dataset classes, namely, edge cracks and road joints, and evaluated the performance of the YOLOv8 architecture on these new classes. For the edge crack class, using a learning rate of 10^{-2}, the mAP value at 0.5 reached 0.963, demonstrating high prediction accuracy. A well-performing model was developed using a modified YOLOv8 architecture, achieving an mAP 0.5 score of 96.3% and an F1 score of 91% with mosaic augmentation preprocessing, indicating excellent performance in detecting road damage.

An application was successfully developed using the YOLOv8 architecture to detect and classify road damage into six categories: potholes, transverse cracks, longitudinal cracks, edge cracks, alligator cracks, and road joints. This application enables users to upload images of road damage for on-device detection and classification, offering a practical tool for efficient road maintenance and inspection.

Although the model demonstrated high accuracy under controlled daytime conditions, future work is necessary to assess its performance in more challenging environments. Future studies may include testing the system under nighttime conditions, where limited illumination and shadows could affect detection accuracy. In addition, the model should be evaluated on different road surface materials, such as concrete roads and unpaved roads, to ensure broader applicability across

diverse infrastructure conditions. Expanding the dataset to include these scenarios is expected to improve the adaptability and reliability of the YOLOv8-based detection system for real-world deployment.

REFERENCES

[1] Y. Song, P. Wu, Q. Li, Y. Liu, and L. Karunaratne, "Hybrid Nonlinear and Machine Learning Methods for Analyzing Factors Influencing the Performance of Large-Scale Transport Infrastructure," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12 287–12 300, Aug. 2022.

[2] Y. Song, P. Wu, D. Gilmore, and Q. Li, "A Spatial Heterogeneity-Based Segmentation Model for Analyzing Road Deterioration Network Data in Multi-Scale Infrastructure Systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 7073–7083, Jun. 2020.

[3] M. Emu, F. B. Kamal, S. Choudhury, and Q. A. Rahman, "Fatality Prediction for Motor Vehicle Collisions: Mining Big Data Using Deep Learning and Ensemble Methods," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 3, pp. 199–209, Jan. 2022.

[4] H. Xin, Y. Ye, X. Na, H. Hu, G. Wang, C. Wu, and S. Hu, "Sustainable Road Pothole Detection: A Crowdsourcing Based Multi-Sensors Fusion Approach," *Sustainability*, vol. 15, no. 8, p. 6610, Apr. 2023.

[5] J. Wang and T. Ueda, "A review study on unmanned aerial vehicle and mobile robot technologies on damage inspection of reinforced concrete structures," *Structural Concrete*, vol. 24, no. 1, pp. 536–562, Jan. 2023.

[6] A.-P. Botezatu, A. Burlacu, and C. Orhei, "A Review of Deep Learning Advancements in Road Analysis for Autonomous Driving," *Applied Sciences*, vol. 14, no. 11, p. 4705, May 2024.

[7] F. Wan, C. Sun, H. He, G. Lei, L. Xu, and T. Xiao, "YOLO-LRDD: a lightweight method for road damage detection based on improved YOLOv5s," *EURASIP Journal on Advances in Signal Processing*, vol. 2022, no. 1, Oct. 2022.

[8] S.-S. Park, V.-T. Tran, and D.-E. Lee, "Application of Various YOLO Models for Computer Vision-Based Real-Time Pothole Detection," *Applied Sciences*, vol. 11, no. 23, p. 11229, Nov. 2021.

[9] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Images," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 12, pp. 1127–1141, Jun. 2018.

[10] L. A. Silva, V. R. Q. Leithardt, V. F. L. Batista, G. Villarrubia González, and J. F. De Paz Santana, "Automated Road Damage Detection Using UAV Images and Deep Learning Techniques," *IEEE Access*, vol. 11, pp. 62 918–62 931, Jan. 2023.

[11] Y. Li, C. Yin, Y. Lei, J. Zhang, and Y. Yan, "RDD-YOLO: Road Damage Detection Algorithm Based on Improved You Only Look Once Version 8," *Applied Sciences*, vol. 14, no. 8, p. 3360, Apr. 2024.

[12] A. A. Sami, S. Sakib, K. Deb, and I. H. Sarker, "Improved YOLOv5-Based Real-Time Road Pavement Damage Detection in Road Infrastructure Management," *Algorithms*, vol. 16, no. 9, p. 452, Sep. 2023.

[13] Y. Safyari, M. Mahdianpari, and H. Shiri, "A Review of Vision-Based Pothole Detection Methods Using Computer Vision and Machine Learning," *Sensors (Basel, Switzerland)*, vol. 24, no. 17, p. 5652, Aug. 2024.

[14] M. L. Pawelczyk and M. Wojtyra, "Real World Object Detection Dataset for Quadcopter Unmanned Aerial Vehicle Detection," *IEEE Access*, vol. 8, pp. 174 394–174 409, Jan. 2020.

[15] Q. Tao, H. Yang, and J. Cai, "Exploiting Web Images for Weakly Supervised Object Detection," *IEEE Transactions on Multimedia*, vol. 21, no. 5, pp. 1135–1146, May 2019.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2016.

[17] M. Hussain, "YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection," *Machines*, vol. 11, no. 7, p. 677, Jun. 2023.

[18] R. Khanam, M. Hussain, R. Hill, and P. Allen, "A Comprehensive Review of Convolutional Neural Networks for Defect Detection in Industrial Applications," *IEEE Access*, vol. 12, pp. 94 250–94 295, Jan. 2024.

[19] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, Feb. 2016.

[20] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, "A review of convolutional neural networks in computer vision," *Artificial Intelligence Review*, vol. 57, no. 4, Mar. 2024.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[22] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Sep. 2014.

[23] K. He, S. Ren, X. Zhang, and J. Sun, "Deep Residual Learning for Image Recognition," Dec. 2015.

[24] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, Feb. 2017.

[25] G. Wang, W. Li, M. A. Zuluaga, R. Pratt, P. A. Patel, M. Aertsen, T. Doel, A. L. David, J. Deprest, S. Ourselin, and T. Vercauteren, "Interactive Medical Image Segmentation Using Deep Learning With Image-Specific Fine Tuning." *IEEE Transactions on Medical Imaging*, vol. 37, no. 7, pp. 1562–1573, Jan. 2018.

[26] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang, "Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?" *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1299–1312, Mar. 2016.

[27] S. Takahashi, Y. Sakaguchi, N. Kouno, K. Takasawa, K. Ishizu, Y. Akagi, R. Aoyama, N. Teraya, A. Bolatkan, N. Shinkai, H. Machino, K. Kobayashi, K. Asada, M. Komatsu, S. Kaneko, M. Sugiyama, and R. Hamamoto, "Comparison of Vision Transformers and Convolutional Neural Networks in Medical Image Analysis: A Systematic Review," *Journal of Medical Systems*, vol. 48, no. 1, Sep. 2024.

[28] R. Roslidar, M. J. Alhamdi, A. Rahman, K. Saddami, F. Arnia, M. Syukri, and K. Munadi, "Self-supervised bi-pipeline learning approach for high interpretation of breast thermal images," *IEEE Access*, vol. 12, pp. 103 433–103 449, 2024.

[29] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection." institute of electrical electronics engineers, Jun. 2016, pp. 779–788.

[30] Q. Cheng, H. Wang, B. Zhu, Y. Shi, and B. Xie, "A Real-Time UAV Target Detection Algorithm Based on Edge Computing," *Drones*, vol. 7, no. 2, p. 95, Jan. 2023.

[31] Z. J. Khow, Y.-F. Tan, H. A. Karim, and H. A. A. Rashid, "Improved YOLOv8 Model for a Comprehensive Approach to Object Detection and Distance Estimation," *IEEE Access*, vol. 12, pp. 63 754–63 767, Jan. 2024.

[32] X. Wang, H. Gao, Z. Jia, and Z. Li, "BL-YOLOv8: An Improved Road Defect Detection Model Based on YOLOv8." *Sensors*, vol. 23, no. 20, p. 8361, Oct. 2023.

[33] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[34] H. Park and J. Paik, "Pyramid attention upsampling module for object detection," *Ieee Access*, 2022.

[35] J. Shang, J. Wang, S. Liu, C. Wang, and B. Zheng, "Small target detection algorithm for uav aerial photography based on improved yolov5s," *Electronics*, 2023.

[36] L. Han, F. Li, H. Yu, K. Xia, Q. Xin, and X. Zou, "Birpn-yolovx: A weighted bidirectional recursive feature pyramid algorithm for lung nodule detection," *Journal of X-Ray Science and Technology*, vol. 31, no. 2, pp. 301–317, 2023, pMID: 36617767. [Online]. Available: https://doi.org/10.3233/XST-221310

[37] W. Li, F. Heltha, A. Rahman, and M. I. Solihin, "Surface flaw detection of small automotive component based on improved faster r-cnn and yolo algorithms," *The International Journal of Advanced Manufacturing Technology*, 2026. [Online]. Available: https://doi.org/10.1007/s00170-026-18208-0

[38] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, pp. 12 993–13 000, Apr. 2020.

[39] H. Peng and S. Yu, "A Systematic IoU-Related Method: Beyond Simplified Regression for Better Localization," *IEEE Transactions on Image Processing*, vol. 30, pp. 5032–5044, Jan. 2021.

[40] T.-H. Cheung and D.-Y. Yeung, "A Survey of Automated Data Augmentation for Image Classification: Learning to Compose, Mix, and Generate." *IEEE transactions on neural networks and learning systems*, vol. 35, no. 10, pp. 13 185–13 205, Oct. 2024.

[41] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features." institute of electrical electronics engineers, Oct. 2019, pp. 6022–6031.

[42] T. Kumar, R. Brennan, A. Mileo, and M. Bendechache, "Image Data Augmentation Approaches: A Comprehensive Survey and Future Directions," *IEEE Access*, vol. 12, pp. 187 536–187 571, Jan. 2024.

[43] N. Dvornik, J. Mairal, and C. Schmid, "On the Importance of Visual Context for Data Augmentation in Scene Understanding." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 6, pp. 2014–2028, Jan. 2020.

[44] R. Walambe, A. Marathe, and K. Kotecha, "Multiscale Object Detection from Drone Imagery Using Ensemble Transfer Learning," *Drones*, vol. 5, no. 3, p. 66, Jul. 2021.