Road Damages Detection Using YOLOv8

Farhan Faturahman
Telkom University
Bandung, 40257 Indonesia
1301213188

Gevin Latifan Abduldjalil
Telkom University
Bandung, 40257 Indonesia
1301213541

Rizal Hidayat Putra
Telkom University
Bandung, 40257 Indonesia
1301213444

Abstract

Road damage, such as cracks, potholes, and surface deformations, significantly impacts transportation safety and road maintenance costs. This project addresses these challenges using YOLOv8, a state-of-the-art object detection framework, to detect and classify various types of road damage. By employing the preprocessed Road Damages Detection v7 dataset with isolated objects, our model demonstrates a precision of 84.4% and a recall of 88.7%, achieving a mean Average Precision (mAP50-95) score of 91.1%. The model was further integrated into a prototype web application, enabling users to upload isolated road damage images for detection. These results underscore the system's potential for real-world applications, improving road inspection efficiency and reducing long-term maintenance costs.

Keywords*: Road Damage Detection, YOLOv8, Object Detection, Deep Learning, Infrastructure Maintenance*

1. Introduction

Road infrastructure is a critical component of urban and rural mobility. Damage such as cracks, potholes, and surface deformations not only compromise safety but also escalate maintenance costs if left unaddressed. These issues contribute to traffic accidents, increased vehicle wear, and extended repair timelines. Manual inspections, the traditional approach, are resource-intensive, slow, and prone to human error. These limitations call for an automated system that is both accurate and efficient [1][2].

The rapid advancements in computer vision and deep learning have paved the way for automated damage detection systems. However, many existing solutions struggle with inconsistent datasets, high computational costs, and limited scalability. This project addresses these challenges by employing YOLOv8, a Convolutional Neural Network (CNN)-based object detection framework. By leveraging its real-time processing capabilities and the preprocessed Road Damages Detection v7 dataset, the project aims to develop a robust solution that can:

- Automate road damage detection with high precision and recall.
- Improve the efficiency of infrastructure inspections.
- Lay the groundwork for integrating automated solutions into road maintenance workflows.

YOLOv8 was chosen over alternative models such as Faster R-CNN and SSD due to its superior trade-off between speed and accuracy. While Faster R-CNN offers high precision, it suffers from slower inference times, making it less suitable for real-time applications. On the other hand, SSD, though faster, often underperforms in detecting smaller objects, which is critical in road damage detection. YOLOv8's lightweight architecture and efficient feature extraction make it the optimal choice for this project, particularly when paired with preprocessed datasets that mitigate noise and enhance detection accuracy. This decision aligns with findings from prior studies highlighting YOLO's capability to handle high-variance datasets effectively [3][4].

The preprocessed Road Damages Detection v7 dataset played a pivotal role in ensuring high detection performance by addressing class imbalances and noise. The dataset's isolated object preprocessing technique allowed YOLOv8 to focus on the damage features, improving learning efficiency and detection accuracy. Such preprocessing aligns with findings from [3][4], which highlight the importance of dataset quality in scalable and efficient detection systems.

This paper provides an in-depth analysis of the methodologies, experiments, and results, demonstrating the practical implications of the proposed system. The findings underscore the potential for deploying such technologies to enhance transportation safety and reduce long-term maintenance costs.

2. Related Work

Road damage detection has gained traction in recent years due to advancements in deep learning. Notable contributions include; Arya et al. [1] developed a global road damage detection system as part of the Global Road Damage Detection Challenge (GRDDC) 2020. The challenge utilized a large dataset comprising 26,336 road images from India, Japan, and the Czech Republic, with models evaluated on two test sets containing 2,631 and 2,664 images, respectively. The best-performing model employed a YOLO-based ensemble learning approach, achieving F1 scores of 0.67 and 0.66 on the two test sets. This study significantly advanced the development of improved road damage detection methods and established a benchmark for future evaluation.

In a complementary study, Arya et al. [2] explored a deep learning-based method for road damage detection using a heterogeneous dataset from multiple countries. This research evaluated the applicability of Japan's smartphone-based road monitoring model in other regions and proposed a large-scale dataset of 26,620 images collected from India, Japan, and the Czech Republic. By addressing the challenges of cost and accessibility, this work highlighted the potential of deep learning technology for road condition monitoring, despite differences in road conditions across regions.

Naddaf-Sh et al. [3] introduced a deep learning approach for evaluating pavement conditions to facilitate timely maintenance and mitigate further infrastructure damage. The dataset included diverse types of road cracks, such as longitudinal, transverse, and alligator cracks, captured using mobile devices. The proposed scalable models achieved F1 scores ranging from 52% to 56% with inference speeds of 10–178 images per second. Additionally, the study provided an error analysis to identify the limitations of the models when faced with data variability.

Angulo et al. [4] addressed the need for balanced and representative datasets by creating a large-scale road damage dataset with 18,034 images and 45,435 damage instances. The study evaluated several object detection methods, including MobileNet, RetinaNet, and traditional approaches like LBP-cascaded classifiers. Their findings demonstrated the utility of these models in real-world scenarios, highlighting the potential for mobile and embedded applications.

Rateke et al. [5] tackled the challenge of detecting roads with varying surface types, including heavily damaged and unpaved roads. Using a U-NET architecture with ResNet34 and transfer learning, the study demonstrated that cost-effective cameras could reliably identify road damage and other surface information critical for driving safety. The introduction of a new ground truth dataset further validated their approach in practical scenarios.

Arya et al. [6] extended their contributions by emphasizing the potential of smartphone-based models for road damage monitoring in developing countries. Their dataset of 26,620 images from multiple nations underscored the importance of heterogeneous data for improving model generalization. The study also offered recommendations for municipalities and agencies to adopt publicly available datasets and models for automated road damage detection and classification.

Although significant progress has been made, challenges remain in achieving high performance across diverse datasets. This project builds upon these studies by leveraging the preprocessed Road Damages Detection v7 dataset to address class imbalances and noise in training data. Integrating YOLOv8 further enhances detection efficiency and performance in real-world applications.

3. Data

Dataset Overview

The Road Damages Detection v7 dataset, sourced from Roboflow Universe [7], consists of 3,506 annotated images across seven distinct damage classes. Figure 1 illustrates samples from each damage class, organized in two rows for clarity.

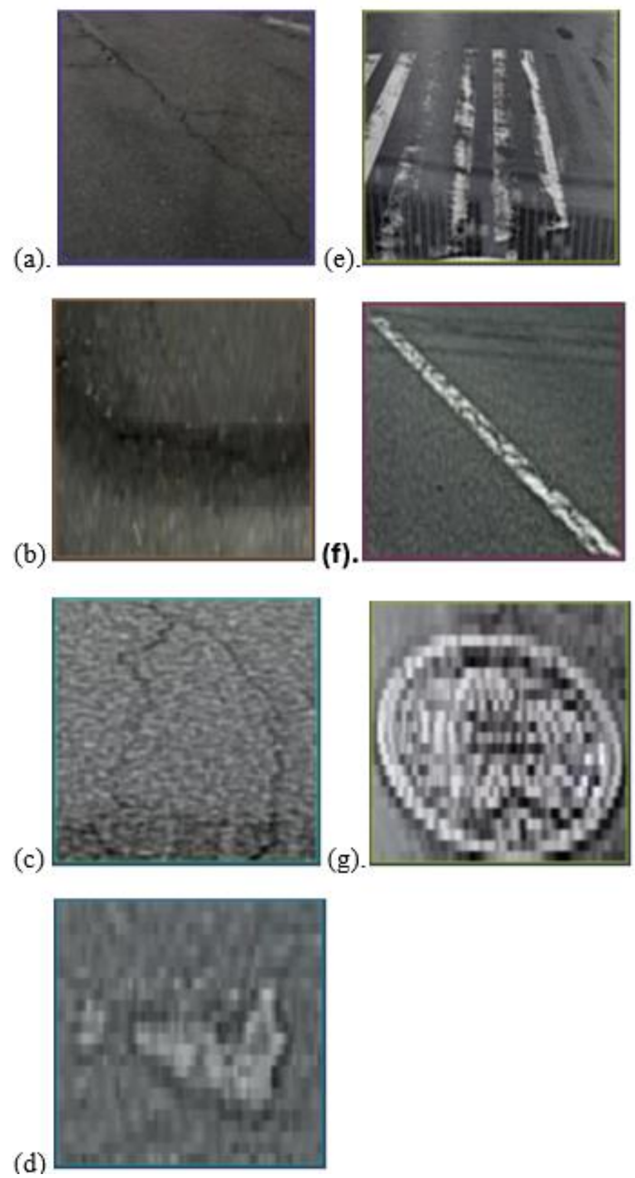

Figure 1 Dataset Samples

(a)	(b)	(c)
Longitudinal Crack (D00)	Pothole (D40)	Alligator Crack (D20)

(d)	(e)	(f)	(g)
Transverse Crack (D10)	Cross Walk Blur (D44)	White Line Blur (D43)	Manhole Cover (D50)

The dataset consists of seven distinct damage classes, each representing a specific type of road defect. **D00 (Longitudinal Crack)** refers to linear cracks that run along the road surface, while **D10 (Transverse Crack)** describes horizontal cracks that typically cross the road. **D20 (Alligator Crack)** comprises interconnected cracks that resemble the skin of an alligator. **D40 (Pothole)** represents depressions or holes in the road surface. **D43 (White Line Blur)** captures instances of blurred road markings, whereas **D44 (Cross Walk Blur)** highlights faded pedestrian crossings. Lastly, **D50 (Manhole Cover)** identifies circular or rectangular covers embedded in the road.

To enhance the dataset's suitability for the YOLOv8 model, comprehensive preprocessing steps were applied. These include normalization to standardize image dimensions and object isolation to remove irrelevant background noise. These preprocessing techniques ensure robust feature extraction and facilitate efficient learning for the detection and classification of road damages.

Preprocessing Techniques

To enhance the dataset's usability for YOLOv8, several preprocessing techniques were applied. One of the key steps was object isolation, where each damage instance was carefully isolated using bounding boxes to effectively remove irrelevant background noise. Additionally, the image dimensions were standardized to 640x640 pixels through normalization, ensuring compatibility with the model's architecture. These preprocessing techniques played a significant role in improving feature extraction, reducing model training time, and enhancing detection accuracy. Furthermore, the alignment of these preprocessing methods with recommendations from previous studies [3][4] underscores their importance in achieving scalable and robust performance for road damage detection tasks.role in achieving scalable and robust performance.

Evaluation Metrics

The evaluation of the model's performance relied on several key metrics. Precision, defined as the proportion of true positives among predicted positives, measures the accuracy of the model's positive predictions. Recall, representing the proportion of true positives among actual positives, evaluates the model's ability to identify all relevant instances of road damage. The F1-score, calculated as the harmonic mean of precision and recall, provides a balanced assessment of the model's performance, particularly when dealing with imbalanced datasets. Additionally, the Mean Average Precision (mAP50-95) was used as a comprehensive metric to evaluate performance across multiple Intersection over Union (IoU) thresholds, offering a more nuanced understanding of the model's detection capabilities across various levels of overlap. These metrics collectively provided a robust framework for assessing the effectiveness of the YOLOv8 model in detecting and classifying road damages.

4. Methods

YOLOv8 Architecture

The YOLOv8 architecture is structured into three primary components: the Backbone, Neck, and Head. The Backbone is responsible for extracting hierarchical features from input images, capturing essential visual information at various scales. These features are then passed to the Neck, which aggregates them to enhance the model's ability to detect objects of varying sizes effectively. Finally, the Head generates the final predictions, including bounding boxes and class probabilities for detected objects. This architecture is designed for efficient feature extraction and object detection, ensuring high performance across a range of object detection tasks.

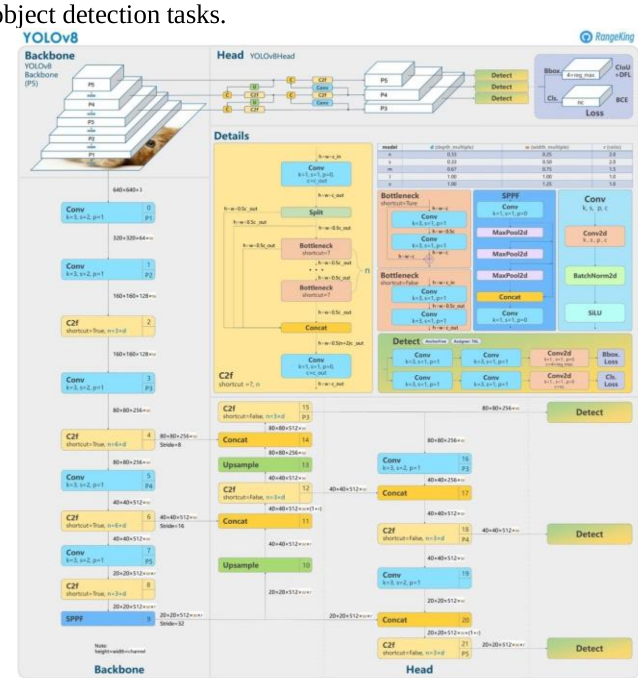

Figure 2 YOLOv8 Architecture Diagram

In this project, the YOLOv8 Nano variant was utilized due to its optimized resource efficiency without compromising performance. This lightweight configuration makes it particularly suitable for real-time applications where computational resources may be limited. The YOLOv8 architecture's innovative design facilitates seamless processing of complex datasets, contributing to its ability to detect and classify road damages with high accuracy. Figure 1 illustrates the structure of the YOLOv8 architecture, showcasing its components and their roles in the detection process.

The architecture's robustness and scalability are further supported by the advancements in convolutional layers, feature aggregation blocks, and detection heads, making YOLOv8 a state-of-the-art choice for object detection tasks. This setup ensures efficient training and real-time inference, which is essential for applications like road damage detection.

Training Configuration

The training of the YOLOv8 Nano model was configured with hyperparameters optimized to balance performance and computational efficiency. Table outlines the key hyperparameters used in this project:

Hyperparameter	Value	Description
Pre-trained Weights	COCO Dataset	Initialized with pre-trained weights to accelerate convergence.
Batch Size	16	Number of images processed per training iteration, optimizing memory usage.
Epochs	50	Number of iterations to ensure comprehensive learning.
Optimizer	AdamW	Utilized with automated learning rate adjustments for consistent updates.
IoU Threshold	0.7	Used to refine bounding box predictions.
Early Stopping	100	Triggered if no significant improvement occurred within the defined epochs.

These configurations were carefully chosen to align with the best practices for scalable model training. For example, the use of pre-trained weights from the COCO dataset accelerated the training process by leveraging prior knowledge. The batch size of 16 ensured an efficient balance between memory usage and training speed. AdamW was selected as the optimizer to provide consistent updates to the model's parameters, while the IoU threshold helped refine the accuracy of bounding box predictions. The implementation of early stopping prevented overfitting by halting training when performance improvements plateaued.

Insights from prior studies, such as those outlined in [3], significantly influenced the fine-tuning of these parameters, ensuring the training process was both efficient and effective.

5. Experiments

Training, Validation, And Testing Results

Metrics	**Training**	**Validation**	**Testing**
Precision	**84.4%**	**84.4%**	**89.1%**
Recall	**88.7%**	**88.7%**	**86.7%**
mAP50	**92.6%**	**92.6%**	**94.4%**
mAP50-95	**91.1%**	**91.1%**	**92.9%**
Duration	**8 hours**	**N/A**	**N/A**

The training phase showed consistent improvement across epochs. The high mAP values indicate effective feature learning, which can be attributed to the isolated object preprocessing and the robust architecture of YOLOv8 Nano.

Validation results mirrored the training phase, demonstrating minimal overfitting. The alignment between training and validation performance is indicative of the model’s generalization capabilities.

The testing phase highlighted the model’s robustness in unseen data, particularly with high precision and mAP scores. The slightly reduced recall compared to training and validation suggests some missed detections, which may be due to challenging examples in the test set, such as faded road markings or heavily blurred features.

Analysis of Results

The training phase, completed in 8 hours, demonstrated consistent improvements in performance metrics. Validation metrics mirrored the training phase, highlighting minimal overfitting and robust generalization capabilities. The testing phase revealed strong performance on unseen data, as evidenced by an mAP50-95 of 92.9%. Slight variations in precision and recall were observed across datasets, attributable to the following factors:

- **Dataset Diversity:** The test set included more complex examples, challenging the model’s detection capabilities.
- **Class Imbalance:** Certain classes, such as D40 (Small Cracks), had fewer examples, impacting performance.

- **Preprocessing Impact:** Object isolation significantly reduced noise, but some subtle features in the test set may have been overlooked.

These results highlight the model's strengths in detecting prominent damage types like transverse cracks and blurred markings while revealing opportunities for future improvement in underrepresented classes.

Prototype Web Application

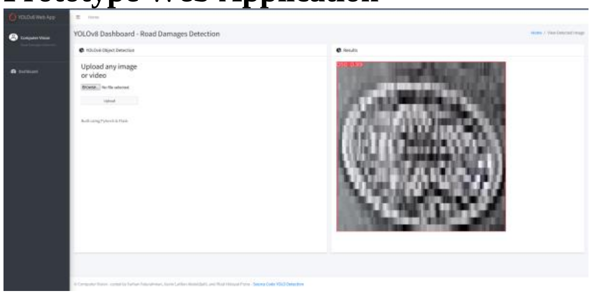

Figure 3 Prototype Web App UI

The trained YOLOv8 model was successfully integrated into a web application prototype. The application allows users to upload images of isolated road damage, with the system providing detection and classification results in real-time. This implementation demonstrates the model's practical applicability and scalability.

Visualizations

The visualizations below provide a detailed representation of the model's performance and the progression of training and validation processes:

1. **Training Loss Curve:** This graph illustrates the reduction in loss over the training epochs, highlighting the model's learning efficiency.
2. **Precision-Recall Curve:** Depicts the trade-off between precision and recall at various confidence thresholds, indicating the model's overall balance between accuracy and sensitivity.
3. **F1 Score Curve:** Shows the F1 scores across different thresholds, representing the harmonic mean of precision and recall.
4. **Confusion Matrix:** Provides insights into the classification performance for each damage class, identifying areas of strength and misclassification.

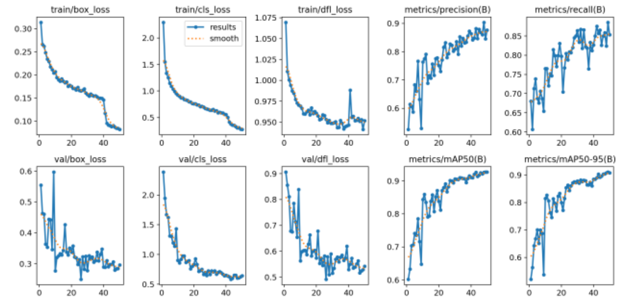

Figure 4 Training and Validation Results

The training loss curve illustrates the consistent reduction in loss over the course of 50 epochs, indicating effective learning by the YOLOv8 Nano model. The smooth curve highlights the stability of the training process, with minimal fluctuations that could indicate instability or overfitting.

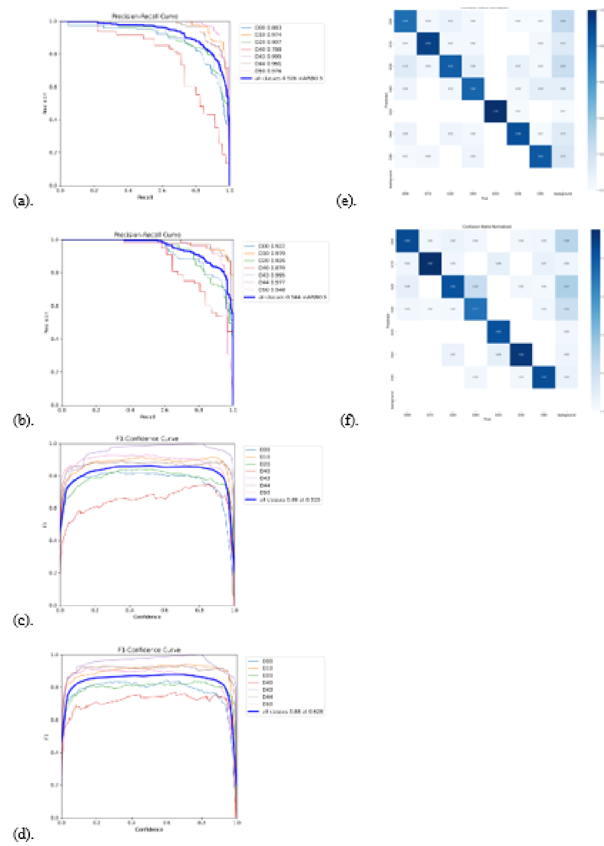

Figure 5 Curve Graph and Confusion Matrix

Figure a: Precision-Recall Curve for Training and Validation

This graph demonstrates the trade-off between precision and recall across different confidence thresholds during the training and validation phases. The near-identical curves for both phases confirm the model's capacity to generalize well without overfitting.

Figure b: Precision-Recall Curve for Testing

The precision-recall curve for the test set reveals slight dips compared to the training and validation curves. This reflects the challenges posed by unseen examples in the test data, including faded markings or subtle damage features.

Figure c: F1 Score Curve for Training and Validation

The F1 score curve for training and validation displays consistent growth as the confidence threshold increases, underscoring the model's balanced optimization for precision and recall during these phases.

Figure d: F1 Score Curve for Testing

In the testing phase, the F1 score curve shows stable but slightly lower performance at lower confidence thresholds. This behavior indicates a reduction in sensitivity, likely due to the diversity and complexity of the test set.

Figure e: Confusion Matrix for Training and Validation

The confusion matrix for training and validation sets shows high diagonal values, demonstrating strong performance across all classes. Minor misclassifications are concentrated in underrepresented classes such as D40 (Potholes).

Figure f: Confusion Matrix for Testing

For the test set, the confusion matrix highlights excellent detection rates for well-represented classes such as D10 (Transverse Cracks) and D43 (White Line Blur). However, underrepresented classes like D40 and D50 (Manhole Covers) exhibit lower detection rates, suggesting the need for further dataset balancing or augmentation.

6. Conclusion

This project successfully applied the YOLOv8 object detection framework for real-time road damage detection, leveraging the preprocessed Road Damages Detection v7 dataset. Through object isolation and normalization, the system demonstrated robust feature extraction, achieving an impressive mAP50-95 of 91.1%. The experimental results highlighted the model's effectiveness in detecting various damage types, particularly transverse cracks and blurred road markings, with minimal overfitting and high generalization capability.

Future work should address the challenges of class imbalance by incorporating advanced augmentation techniques and class-weighted loss functions to enhance performance for underrepresented classes, such as small cracks and manhole covers. Additionally, expanding the dataset to include more diverse road conditions will further validate the model’s robustness. Exploring the system's scalability to other infrastructure-related applications, such as bridge or railway damage detection, could significantly extend its utility and impact.

References

[1] Arya, D., et al. *Global road damage detection: State-of-the-art solutions*. IEEE, 2020.

[2] Arya, D., et al. *Deep learning-based road damage detection*. Automation in Construction, 2021.

[3] Naddaf-Sh, S., et al. *An efficient and scalable deep learning approach for road damage detection*. IEEE, 2020.

[4] Angulo, A., et al. *Road damage detection acquisition system*. Springer, 2019.

[5] Rateke, T., et al. *Road surface detection and differentiation considering surface damages*. Autonomous Robots, 2021.

[6] Arya, D., et al. *RDD2020: Annotated image dataset for road damage detection*. Data in Brief, 2021.

[7] Tam, E. *Road Damages Detection Dataset*. Roboflow Universe, 2023.