Applied Ensemble Technique for Road Damage Detection based on YOLOv8

Zhipeng Tang[1], Apirak Jirayusakun[2] and Rapeeporn Chamchong[1,*]

[1] Department of Computer Science, University of Mahasarakham, Maha Sarakham, Thailand
[2] Department of Computer Science, University of Ramkhamhaeng, Bangkok, Thailand

*Corresponding Email: rapeeporn.c@msu.ac.th

Received February 12, 2025, Revised June 12, 2025, Accepted July 4, 2025, Published December 30, 2025

Abstract. *The presence of road damage poses significant risks to pedestrians and traffic. Although deep learning-based object detection is widely used, the effectiveness of various detections shows considerable variation, and there remains substantial potential for enhancement. This paper proposes an ensemble technique using YOLOv8 for road damage detection. A comparison of object detection models, including YOLOv5, YOLOv8, Faster R-CNN, and SSD, is conducted to determine the best baseline. The study focuses on single-class detection to enhance accuracy in identifying specific types of road damage. Each model predicts class probabilities and bounding box locations. The predictions are then ensembled, with Non-Maximum Suppression applied to filter out overlapping detection boxes. The ensemble YOLOv8 model outperforms the standard one, especially in detecting alligator cracks and potholes, with detection accuracy improved by up to 3%. The method balances precision and recall effectively, suitable for complex road environments.*

Keywords: ensemble technique, object detection, road damage detection, deep learning,

1. Introduction

Roads have been fundamental to human civilization since ancient times, evolving from primitive paths to modern highways. Nowadays, roads are a critical component of transportation infrastructure, facilitating the movement of people, goods, and services. Modern roads are primarily constructed using asphalt and concrete materials, which, despite their durability, are susceptible to various forms of deterioration. These materials can degrade due to environmental conditions (temperature variations, rainfall), heavy traffic loads, aging, and extreme weather events, leading to various forms of damage, such as cracks, potholes, and surface deformation [1].

Road damage can have serious consequences for both drivers and pedestrians. Unrepaired potholes and cracks can cause vehicle damage, leading to costly repairs and an increased risk of accidents. Severe road degradation also makes travel more difficult and uncomfortable, especially for public transport, emergency vehicles, and vulnerable road users like cyclists and the elderly. Over time, unchecked road damage can even compromise the structural integrity of the entire transportation network, requiring major and expensive reconstruction efforts [2]. Road damage often presents a wide range of characteristics, which can vary in size, shape, and severity, making detection challenging, especially when minor damages, such as hairline cracks, are in the early stages and not yet fully visible [3].

Therefore, timely road damage detection and repair is crucial for maintaining a safe and efficient transportation system. Traditional methods of road inspection, such as visual surveying by human inspectors, are time-consuming, subjective, and unable to cover large road networks. The recent advancements in computer vision and deep learning techniques offer a promising alternative for automated, scalable, and objective road damage assessment [4], [5]. By analyzing road imagery captured by vehicle-mounted cameras or drones, deep neural networks can rapidly identify and locate various defects with high accuracy. This intelligent monitoring system helps agencies prioritize maintenance, optimize resources, and improve service delivery.

In this paper, we propose an applied ensemble technique for road damage detection using YOLOv8 model with the following main contributions:

- An extensive evaluation of various standard detectors is conducted to determine the most effective baseline model for detecting road damage.
- Single-class detection methods improve the model's capacity to precisely identify specific types of road damage by reducing interference among various classes.
- Ensemble techniques are applied to enhance overall detection accuracy, resulting in a more robust and accurate model.
- Comprehensive experiments are carried out to confirm the efficiency of the ensemble YOLOv8 method in detecting road damage.

2. Related Work

Object detection is a key task in computer vision that focuses on identifying and categorizing objects in images. Traditional image processing detection methods demonstrate efficiency in capturing low-level visual patterns; however, they encounter challenges when addressing complex scenes, fluctuating lighting conditions, and object deformations [6]. This difficulty arises from their reliance on hand-crafted features and strict matching rules, which may not adapt well to diverse visual contexts. Although these methods are relatively early, some techniques can still serve as effective image preprocessing steps to enhance overall performance. The emergence of deep learning has revolutionized this field, mainly through Convolutional Neural Networks

(CNNs), which serve as the backbone architecture in modern deep learning frameworks. CNNs have demonstrated exceptional capability in automatically learning hierarchical feature representations from raw image data, significantly improving detection accuracy compared to traditional computer vision methods [7]. Furthermore, various deep learning techniques have been adopted to enhance detection performance. These include transfer learning, which employs pre-trained models from extensive datasets; ensemble learning, which integrates multiple models to strengthen robustness; and attention mechanisms, which allow models to focus on pertinent regions within images.

Current deep learning-based object detection techniques can generally be divided into two main categories: one-stage detectors and two-stage detectors. [8]. One-stage detectors, including the You Only Look Once (YOLO) series, Single Shot Detector (SSD), MobileNet, and RetinaNet [9], are specifically developed to perform object class predictions and bounding box estimations in a single forward pass through the network. This methodology offers enhanced inference speeds and a streamlined network architecture, making these models well-suited for real-time applications. However, it is essential to note that this efficiency may result in a slight compromise in detection accuracy. In contrast, two-stage detectors, exemplified by the R-CNN series (which includes R-CNN, Fast R-CNN, Faster R-CNN, and Mask R-CNN) [10], operate by initially generating region proposals. Following this step, these proposals undergo classification and bounding box refinement. This method results in enhanced detection accuracy but requires greater computational resources and processing time.

A. One-Stage detectors

Zhang et al. [11] proposed a CNN-based framework that learns features directly from raw image patches. Key innovations include random patch rotation, dropout for preventing overfitting, multi-view averaging, and ReLU activation for faster training. Meanwhile, Maeda et al. [12] integrated a Single Shot Detector (SSD) with MobileNet and Inception V2 backbones, creating a unified detection framework that eliminates separate preprocessing stages while maintaining computational efficiency. Aqsa et al. [13] evaluated MobileNet SSD models using a dataset of 18,756 images, finding MobileNet SSD V2 with optimized hyperparameters outperformed V1 in detecting various crack types.

Mandal et al. [14] integrated YOLO v2 with transfer learning from COCO dataset weights using a streamlined architecture that omitted the third convolutional and pooling layer from standard YOLO v2. Alfarrarjeh et al. [15] demonstrated YOLO's application in road damage detection using smartphone images by training on eight damage types. They addressed class imbalance with image augmentation and optimized non-maximum suppression for overlapping detections, achieving an F1 score of 0.62. Hassan et al. [16] improved YOLOv3-tiny by adding three additional convolutional layers to enhance feature extraction capabilities, achieving better accuracy and mean average precision while maintaining real-time UAV and vehicle deployment performance.

Camilleri and Gatt [17] evaluated YOLO and SSD architectures for pothole detection on embedded systems, finding that YOLOv3-SPP achieved the highest accuracy. At the same time, SSD-Lite MobileNet v2 showed better inference speed on mobile devices and Raspberry Pi. Angulo et al. [18] developed a RetinaNet-based system with a VGG19 backbone, trained on over 18,000 augmented images, achieving 0.91 mAP and 0.5s inference time on mobile devices. Jeong [19] developed a road damage detection system using YOLOv5x with test-time augmentation and model ensemble techniques, incorporating CSPNet backbone and Spatial Pyramid Pooling for handling various input image sizes.

B. Two-Stage detectors

Anand et al. [20] utilized a modified SqueezeNet architecture with an encoding layer to extract texture and spatial features. The method employs SegNet for road segmentation and edge detection to generate candidate regions. Bibi et al. [21] utilized pre-trained deep learning models, ResNet-18 and VGG-11, along with preprocessing steps, such as noise removal and data augmentation, for handling datasets of road anomalies. Chen et al. [22] utilized a Mask R-CNN framework with a DenseNet backbone, incorporating a feature pyramid network for multi-scale features, a region proposal network for generating candidate regions, and three heads for classification, bounding box regression, and mask generation. Pre-trained COCO weights were fine-tuned on a road damage dataset.

Fan et al. [23] integrated a Graph Attention Layer (GAL) into DeepLabv3+ to refine feature representations using graph-based relationships. The model combines a backbone for feature extraction, a GAL module for refinement, and an ASPP module for context aggregation, achieving superior pothole detection on RGB, disparity, and transformed disparity images. Liu et al. [24] utilized the Deeplabv3+ model to segment road areas and generate a road-interest map, which was then used as input for object detection. They utilized ResNeXt-101 as the backbone of Faster R-CNN, alongside Feature Pyramid Networks (FPN) and Deformable Convolution Networks (DCN), to improve feature extraction and adaptability to geometric transformations.

Pham et al. [25] used Detectron2's Faster R-CNN with ResNeXt-101-FPN as the backbone, tuning anchor box sizes and aspect ratios. The model achieved F1 scores of 51.0% and 51.4% on Test1 and Test2 datasets. Rateke et al. [26] applied region of interest extraction by retaining the lower half of each image to focus on road-relevant features. This approach removed irrelevant areas like the vehicle or background and ensured consistent preprocessing across datasets with similar perspectives.

C. YOLO

The YOLO algorithm trains the entire network using the whole image. It relies on a single neural network, taking images and their respective ground truth bounding boxes or segments as input during training. The results of this training are the bounding boxes and their associated labels for detecting objects within the image. Because the detection process utilizes a single network, it can be directly optimized for enhanced detection performance. This approach divides the image into a grid of cells. The features of each cell are utilized to identify objects with the centers of bounding boxes contained within that cell. The outcomes of the training are achieved in a single-step manner.

The evolution of the YOLO series, ranging from YOLOv1 to YOLOv11, has demonstrated ongoing enhancements focused on increasing speed and precision for real-time object detection applications [27]. The architecture of YOLOv8 is optimized for effective feature extraction and accurate object detection. As shown in Fig.1, it incorporates convolutional modules and residual blocks within its backbone, particularly using the C2f structure for its residual connections. The neck incorporates a

PAN-FPN network designed with the Cross Stage Partial (CSP) architecture to facilitate the fusion of multi-scale features by performing up-sampling and down-sampling operations multiple times. The Spatial Pyramid Pooling Framework (SPPF) module conducts pooling operations across various scales. It effectively combines feature maps from diverse scales to enhance the detection capability for objects of varying sizes. YOLOv8 has three distinct detection heads, each specifically designed to recognize objects of varying sizes: small, medium, and large. The model incorporates a decoupled head architecture, differentiating between classification and detection tasks and enhancing overall accuracy. It utilizes Binary Cross Entropy (BCE) for classification and employs a blend of Complete Intersection over Union (CIoU) and Distribution Focal Loss (DFL) for object localization. Additionally, it embraces an anchor-free method to enhance the identification of positive and negative samples, which increases processing speed while preserving high accuracy.

D. Ensemble Learning Technique

Ensemble learning is an advanced methodology that aims to enhance predictive accuracy by integrating multiple algorithms. This method utilizes the unique advantages of different models to develop a more reliable and precise predictive system. The process usually includes creating multiple weak models, known as base classifiers, and integrating their predictions using approaches like bagging, boosting, or stacking. [28]. The benefit of ensemble learning is its capacity to decrease errors by averaging the predictions of individual models, leading to enhanced accuracy and improved generalization compared to any single model.

Ensemble learning has proven highly effective in detecting road damage. By harnessing the strengths of multiple models, this approach enhances both accuracy and robustness. Ding et al. [29] employed a hybrid approach that integrates one-stage detectors, such as the YOLO-series models, with two-stage detectors like Faster R-CNN. This innovative method has resulted in exceptional performance in various international road damage detection challenges. Doshi and Yilmaz [30] implemented a deep ensemble approach by utilizing multiple YOLOv4 models, significantly advancing the classification of road damage. Concurrently, Wang et al. [31] established a powerful ensemble method that integrates YOLOv5 with attention modules, achieving marked improvements in model adaptability and detection accuracy. Hegde et al. [32] proposed an ensemble of u-YOLO models that incorporates test-time augmentation (TTA) to enhance the accuracy of road damage detection. This approach, which integrates multiple models alongside augmentation strategies, significantly improves the robustness of the predictions. These studies underscore the effectiveness of ensemble learning in advancing road damage detection by utilizing a variety of model architectures and data augmentation techniques.

3. Materials and Methods

A. Overview

The approach for detecting road damage starts with collecting an extensive dataset on road damage, then moves on to data preprocessing employing mosaic augmentation [33] to guarantee that it is appropriate for model training. The dataset is split into training, validation, and test subsets. As shown in Fig.2, this approach involves a comparison of state-of-the-art detections based on deep learning, including YOLOv5, YOLOv8, Faster R-CNN, and SSD. These detections examine their performance across essential metrics such as precision, recall, and F1-score to determine the most efficient baseline models. To better understand each model's capabilities, single-class detection is conducted to evaluate their effectiveness in identifying specific types of road damage, highlighting their unique strengths and weaknesses. Ultimately, ensemble learning methods are employed to merge predictions from these models, improving overall detection accuracy and reliability by utilizing the distinct advantages of each approach. Through a comprehensive comparison and analysis of the general and ensemble models, we seek to identify strategies for enhancing detection efficiency.

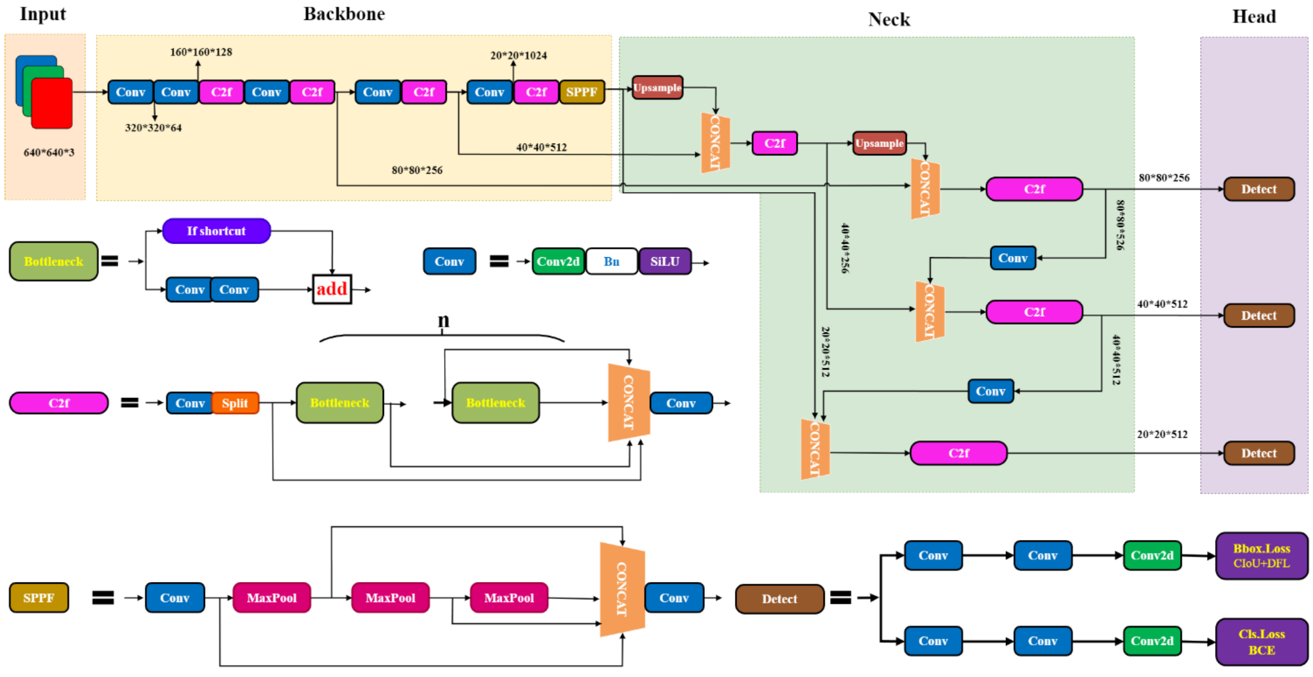

Figure 1. YOLOv8 network architecture diagram.

B. Dataset

The standard dataset was provided by the Crowdsensing-based Road Damage Detection Challenge (CRDDC'2022) [34]. The CRDDC'2022 challenge calls for researchers from around the world to suggest solutions for the automatic detection of road damage in various countries. The images were gathered from six different countries: India, Japan, the Czech Republic, Norway, the United States, and China. The data was gathered through multiple methods, such as smartphones, high-resolution cameras, and Google Street View. Japan's dataset covers urban and snowy regions. India's dataset includes local/state/national highways across metropolitan/rural areas. Norway's dataset includes expressways and county roads, capturing diverse weather conditions like snow and rain. China provides overhead drone images (China_D). The dataset contains 47,420 images. Japan has the largest share at 27.7% (13,133 images). In contrast, China provided the fewest images, totaling 4,878 (10.3%).

The RDD2022 dataset is split into two sections: training and test sets. The train set contains road images and annotations in XML files formatted in PASCAL VOC. The test set is made without any annotations. The image sizes are 512×512, 600×600, 720×720, and 3650×2044. It comprises 38385 training images, 55007 labels, and four categories of damage: longitudinal crack, transverse crack, alligator crack, and pothole (designated as D00, D10, D20, and D40), as shown in Fig.3. Among all the labels, longitudinal cracks (D00) represent the largest percentage at 47% (26,016 labels), whereas potholes (D40) have the smallest percentage at 12% (6,544 labels). Transverse cracks (D10) and alligator cracks (D20) constitute 22% (11,830 labels) and 19% (10,617 labels) respectively.

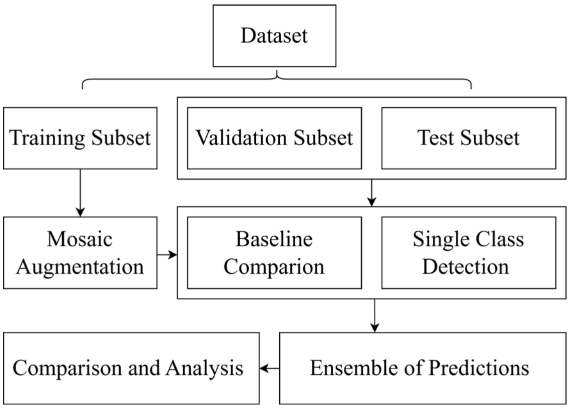

Figure 2. Overview of road damage detection methodology.

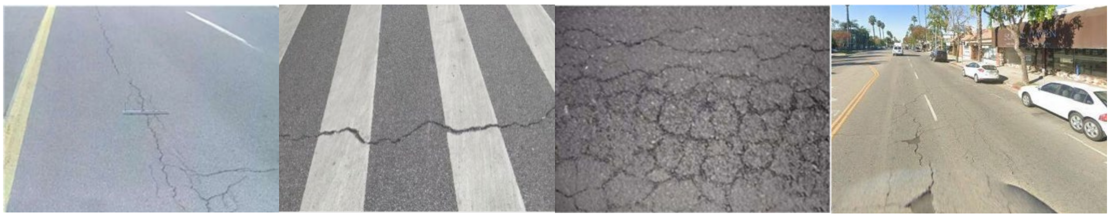

Figure 3. Four types of road damage for detection: top-left (longitudinal crack), top-right (transverse crack), bottom-left (alligator crack), and bottom-right (pothole).

C. Proposed Ensemble Method

In a typical multi-class detection scenario, the detector is tasked with distinguishing among multiple classes simultaneously. This adds a layer of complexity, as the features that differentiate one class may overlap with those that delineate another. However, the model can enhance its focus by directing the detection process toward a specific class or a selected subset of classes. This approach allows for optimization tailored to the distinct characteristics inherent in a particular class [35]. Consequently, the probability of interference between different classes is reduced, improving detection accuracy. Before applying this method, the best-performing model should be selected as the baseline. As illustrated in Fig.4, an ensemble approach is proposed, involving training multiple models specifically designed to detect distinct classes and effectively integrate their predictions. In this approach, each model independently generates probabilities and locations for each class within the test dataset. This output reflects the model's confidence level in accurately classifying an object into its respective categories. To obtain the final result, Non-Maximum Suppression (NMS) is used on the output of each model to eliminate redundant or overlapping detection boxes.

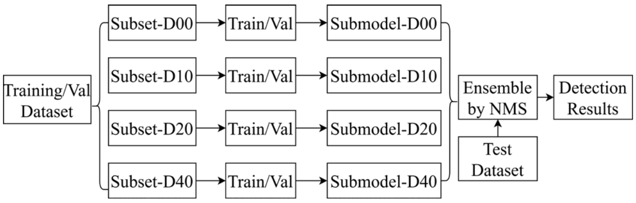

Figure 4. Ensemble method workflow for road damage detection.

When classification is performed using ensemble methods, it is common for the output to present multiple bounding boxes for the same object, each representing a distinct confidence score. NMS eliminates bounding boxes that overlap or are redundant and signify the same object. The specifics are as follows: each submodel identifies the damage category in an image by producing bounding box coordinates along with confidence scores for every detected object. The process should commence by organizing the predictions for each damage category across all models into a two-dimensional array. This array is arranged in descending order based on confidence scores for improved clarity and organization. The array, categorized and sorted, is organized as follows: [[c_1,c_2,…,c_i], [[x_{11},y_{11},x_{12},y_{12}], [x_{21},y_{21},x_{22},y_{22}],…,[x_{i1},y_{i1},x_{i2},y_{i2}]]]. In this context, c_i denotes the confidence score for every bounding box, while [x_{i1},y_{i1},x_{i2},y_{i2}] refers to the coordinates of the top-left and bottom-right corners of the bounding box. The NMS procedure begins by identifying the box with the highest confidence score, c_1, along with its associated bounding box [x_{11},y_{11},x_{12},y_{12}]. The Intersection over Union (IoU) with the chosen box is computed for every remaining bounding box. When the IoU exceeds a predefined threshold, usually established at 0.5, the corresponding bounding box is eliminated from the list since it is deemed the same object. The subsequent bounding box with the highest confidence is selected from the remaining list, followed by a repeated evaluation using the IoU metric. This procedure continues until all bounding boxes have either been selected or eliminated, ensuring that only the most confident and non-overlapping bounding boxes are retained.

The IoU is the ratio of the intersection area (the overlapping region) to the union area (the total area covered by both rectangles). For instance, given two overlapping bounding boxes, A and B, the overlapping region is represented as (A∩B), and the union area is defined as (A∪B). The formula for calculating IoU is as follows:

$$IoU = \frac{A \cap B}{A \cup B} \tag{1}$$

D. Evaluation Metrics

Evaluation metrics for road damage detection play a vital role in assessing how effectively models can identify and localize various types of road damage in images. The metrics are used to measure the accuracy and efficiency of detection. Precision, Recall, F1-score, and mean Average Precision (mAP) are commonly used evaluation metrics for road damage detection. As outlined in the formulas below, True Positives (TP) refer to instances where the model correctly detects road damage. Conversely, False Positives (FP) are instances of incorrect detections where the model identifies damage that is not present. False Negatives (FN) occur when the model fails to detect actual damage in the image.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

mAP is the most commonly employed metric for assessing object detection performance. It is computed by determining the average precision (AP) for each individual class and then averaging these values across all classes. AP is derived by plotting the precision-recall curve and calculating the area underneath this curve. Typically, mAP is calculated at various IoU thresholds (e.g., 0.5 and 0.75).

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{5}$$

4. Experiments and Result Analysis

A. Benchmark Comparison

In this section, we perform an extensive benchmark analysis of four object detection models on different scales: YOLOv5, YOLOv8, Faster R-CNN_Resnet101, and SSD_MobileNetV2. The dataset is subsequently divided into three subsets for training, validation, and test, maintaining a ratio of 8:1:1. During the preprocessing of the dataset, every input image is resized to dimensions of 640×640 pixels, and mosaic augmentation is implemented for the training images. The augmentation technique includes cropping and rotating sections of images, adjusting brightness, applying color transformations, and blending them with other images. This approach enhances the diversity of the training data, encouraging the model to learn object detection within varied contexts and enabling it to generalize more effectively to different scenarios.

Each model is trained using a learning rate of 0.001, a batch size of 16, and 100 epochs. To assess detection accuracy, an IoU threshold of 0.5 is utilized. For YOLO, we employ all five pre-trained models (n, s, m, l, x), with each model being trained separately. The YOLO models undergo pre-training using the COCO dataset comprising 80 categories. The key distinction between these models is their depth and width. In principle, a model characterized by substantial depth and width, referred to as "x," is expected to deliver the highest level of detection performance. Nonetheless, it also possesses the highest number of parameters and proceeds at the slowest speed. All experiments were carried out on Google Colab, utilizing an A100 equipped with 40GB of GPU memory and a system with 50GB of RAM. For the smaller models, each training round requires approximately two minutes, resulting in a total duration of about three hours for the comprehensive training and validation process. In contrast, the larger models demanded seven to nine minutes per round, ultimately totaling around 14 hours for the entire procedure. The findings from these experiments are detailed in Table 1.

The experimental results indicate that the YOLOv8 series demonstrates superior performance compared to YOLOv5 across key metrics. For efficiency, lightweight models like YOLOv5n and YOLOv8n stand out, making them ideal for applications where speed and minimal computational resource consumption are crucial. YOLOv8l achieves the highest mAP@0.5 of 0.636, indicating its exceptional capability in road damage detection. This performance maintains a strong balance between precision and recall, achieving highest F1 score of 0.62 . Overall, YOLOv8m outperforms precision, whereas YOLOv5l exhibits a higher recall rate. Although YOLOv5x performs well, with a mAP@0.5 of 0.612 and an F1 score of 0.617, it still falls short compared to the YOLOv8l models. The Faster R-CNN model lags comprehensively behind YOLO models in both detection accuracy and latency. Meanwhile, SSD exhibits the weakest results, indicating it is less suitable for this task. These findings highlight the effectiveness of YOLOv8l as a robust benchmark for road damage detection, consistently displaying strong accuracy while ensuring reasonable inference time and model size for portable device.

Table 1 Experimental results of baseline models comparison

Model	Precision	Recall	mAP@0.5	F1	Param. (M)	Latency (ms)
YOLOv5n	0.558	0.534	0.554	0.546	**2.5**	0.9
YOLOv5s	0.592	0.582	0.564	0.587	9.1	1.2
YOLOv5m	0.61	0.591	0.58	0.6	64	2.1
YOLOv5l	0.632	**0.595**	0.59	0.612	134.7	3.2
YOLOv5x	0.657	0.583	0.612	0.617	246.4	5.3
YOLOv8n	0.649	0.526	0.585	0.58	3.0	**0.8**
YOLOv8s	0.684	0.542	0.609	0.60	11.1	1.2
YOLOv8m	**0.689**	0.556	0.627	0.62	25.8	2.3
YOLOv8l	0.673	0.583	**0.636**	**0.62**	43.6	3.5
YOLOv8x	0.672	0.578	0.634	0.62	257.4	5.1
Faster R-CNN	0.652	0.588	0.61	0.599	48.6	7.3
SSD	0.613	0.53	0.57	0.568	14.9	5.9

Fig. 5 illustrates the precision-recall curve for the YOLOv8l model across four damage categories (D00, D10, D20, and D40), The mAP value for each category corresponds to the area under its respective precision-recall curve. The D20 category, which represents alligator crack, stands out with the highest mAP@0.5 of 0.715, showcasing significantly better performance than the other categories. This can be attributed to the distinct and repetitive visual patterns of alligator cracks, which are easier for the model to detect and classify accurately. In contrast, categories like D10 and D40 show lower mAP@0.5 values of 0.6 and 0.607, respectively. The elongated and less distinct nature of these cracks likely contributes to the model's relatively lower performance in detecting them. The overall mAP@0.5 for all classes is 0.636, indicating a strong overall balance between precision and recall but also highlighting the variability in detection performance across different damage categories.

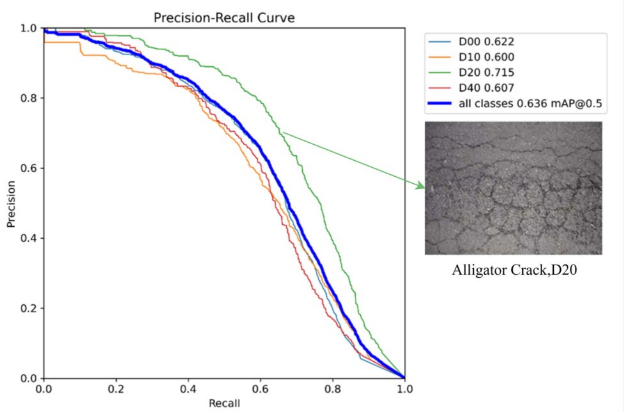

Figure 5. Precision-Recall curve for YOLOv8l.

B. Ensemble of YOLOv8l's Single-Class Detection Results

In this experiment, the dataset was divided by category. YOLOv8l was used to train and validate four categories individually and ensemble using NMS. The ensemble model's inference speed has slowed down for combining multiple single-class predictions. The experimental results are shown in Table 2. The detection accuracy for each category improved significantly. Each category-specific submodel outperformed the general model in its respective categories, with improvements ranging from 1.1% to 3.7%. Compared to the general model, YOLOv8l ensemble model showed improvements of around 2-3% in detection accuracy for D20 and D40, while experiencing a slight reduction of about 1-2% in others. D00 and D10 exhibit high mutual misclassification due to their shared elongated crack features, differing only in direction. NMS further compounds errors by discarding valid detections from overlapping bounding boxes, leading to reduced precision for both classes. Performance variability is observed in ensemble methods due to their generalized nature. While combining models can enhance overall effectiveness, it may also diminish the unique advantages that individual models possess, particularly when addressing a diverse range of object types or damage categories.

Table 2 mAP@0.5 of single-class detection and ensemble model

Model	D00	D10	D20	D40	Over all
Submodel_D00	0.631	-	-	-	-
Submodel_D10	-	0.623	-	-	-
Submodel_D20	-	-	0.752	-	-
Submodel_D40	-	-	-	0.635	-
YOLOv8l_General	0.622	0.6	0.715	0.607	0.636
YOLOv8l_Ensemble	0.618	0.58	0.745	0.629	0.643

The evaluation metrics offer an additional understanding of the model's detection effectiveness. As illustrated in Fig.6, the training and validation losses for the box, class, and distribution focal loss typically decline as the number of epochs increases, exhibiting only minor fluctuations, which leads to a smooth curve. The validation losses exhibit comparable downward trends but tend to stabilize at slightly higher levels compared to the training losses. The general trend of the curves indicates that the model is advancing positively. The training loss graph shows a constant decline, while the validation curve hits its minimum around the midpoint of the epoch. Even though the validation loss begins to increase slightly toward the end, the change is minimal and can be viewed as insignificant concerning the overall trend. Performance metrics enhance progressively, with precision and mAP@0.5 nearing approximately 0.6, while mAP@0.5-0.95 levels at around 0.35, indicating effective detection performance across various overlap thresholds. The model achieves optimal performance within the 60-70 epochs, indicating that additional training beyond this point may lead to diminishing returns. This observation suggests a strategic approach to further training efforts.

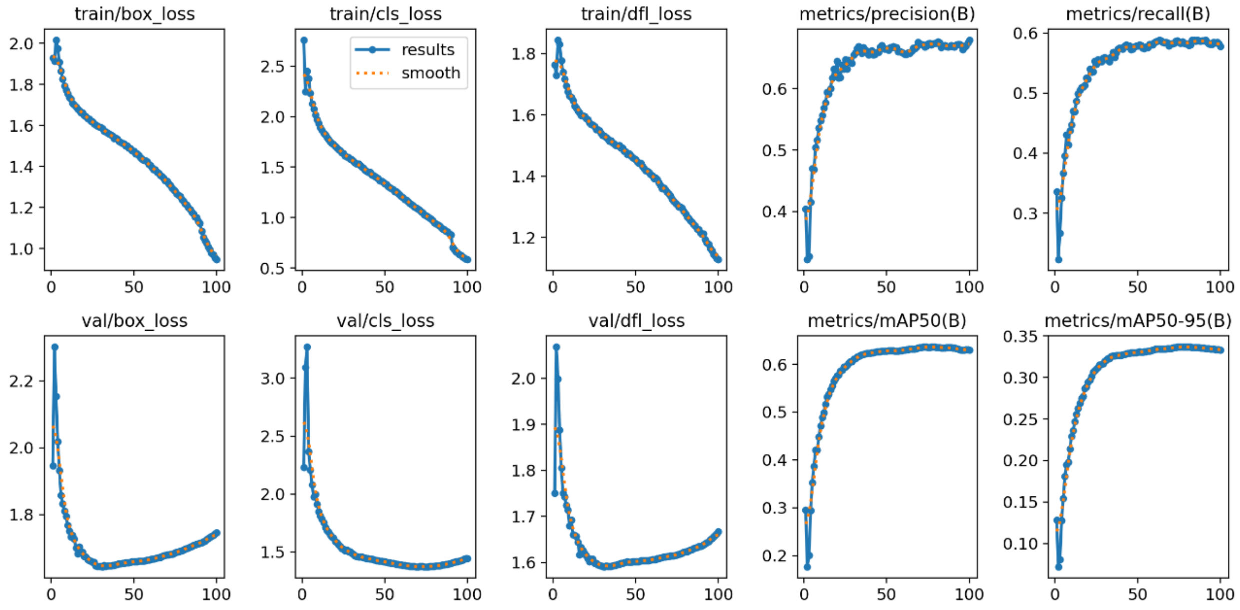

Figure 6. Performance curves of YOLOv8l_Ensemble over epochs.

Fig.7 illustrates the comparison between the predictions of the general model (left) and the ensemble model (right). It can be observed that both models demonstrate a high ability to detect road damage in complex environments, avoiding irrelevant objects such as manholes and repaired patches within the image. The general model is adept at identifying small cracks in the center of zebra crossings, often overlooked during visual inspections. Meanwhile, the ensemble model exhibits higher confidence scores for specific categories, such as D20 and D40, which is evident in the elevated confidence values on the right side of the image. However, it is significant to note that the ensemble model tends to miss certain types of road damage, like the D00 damage located in the top left corner, which the general model successfully detects but not the ensemble model.

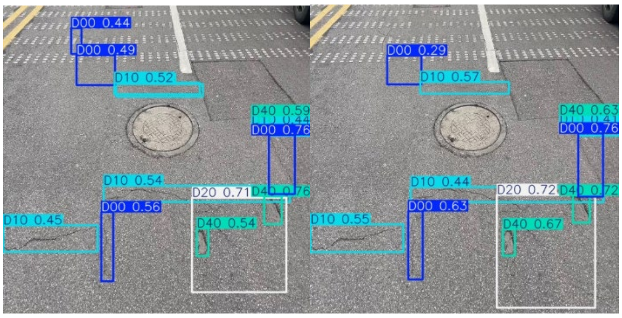

Figure 7. Comparison of general(left) and ensemble(right) predictions.

5. Conclusion

This paper introduces an enhanced method for detecting road damage utilizing the ensembled YOLOv8 model, improving accuracy and robustness. Through a comprehensive comparison of various models and single-class detection approaches, we have illustrated that category-specific submodels perform better than general detection models within their respective damage classifications. The ensemble approach elevates overall performance, especially when tackling challenging categories like alligator cracks and potholes. The notable gains fully validate the approach's effectiveness, reaching as much as 3% for those categories. However, this powerful ensemble model has drawbacks; it sometimes falters, overlooking smaller damage types, such as minor cracks that could easily go unnoticed. The accuracy in detecting the two varieties of elongated cracks is relatively low, leading to only a slight enhancement in overall performance. The ensembled YOLOv8 approach demonstrates significant promise for practical applications in road damage detection. It offers an effective solution for identifying and classifying road damage across a range of diverse and complex environments. This capability positions it as a valuable tool in ensuring road safety and maintenance efficiency.

For future research, the focus should primarily be on enhancing the detection performance of elongated cracks to improve overall efficacy further. Additionally, exploring various deep learning ensemble approaches and investigating the practical application of these methods in real-time scenarios is important.

Acknowledgments

This research project was financially supported by Mahasarakham University.

References

[1] E. H. Manurung, K. Sawito, A. Satoto, and N. Tuanany, “Analysis of the Causes of Road Damage,” civilla, vol. 7, no. 1, p. 87, Apr. 2022, doi: 10.30736/cvl.v7i1.793.

[2] T. Tsubota, C. Fernando, T. Yoshii, and H. Shirayanagi, “Effect of Road Pavement Types and Ages on Traffic Accident Risks,” Transportation Research Procedia, vol. 34, pp. 211–218, 2018, doi: 10.1016/j.trpro.2018.11.034.

[3] J. Wu, Y. Zhang, and X. Zhao, “A Review of Image-Based Pavement Crack Detection Algorithms,” in 2021 40th Chinese Control Conference (CCC), Shanghai, China: IEEE, Jul. 2021, pp. 7300–7306. doi: 10.23919/CCC52363.2021.9549966.

[4] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, “Object Detection in 20 Years: A Survey,” Proceedings of the IEEE, vol. 111, no. 3, pp. 257–276, 2023, doi: 10.1109/JPROC.2023.3238524.

[5] X. Wu, D. Sahoo, and S. C. H. Hoi, “Recent Advances in Deep Learning for Object Detection,” Neurocomputing, vol. 396, pp. 39–64, Jul. 5, 2020, doi: 10.1016/j.neucom.2020.01.085.

[6] G. Shen, “Road crack detection based on video image processing,” in 2016 3rd International Conference on Systems and Informatics (ICSAI), Shanghai, China: IEEE, Nov. 2016, pp. 912–917. doi: 10.1109/ICSAI.2016.7811081.

[7] N. Ma et al., “Computer vision for road imaging and pothole detection: a state-of-the-art review of systems and algorithms,” Transp Safety Env, vol. 4, no. 4, p. tdac026, Nov. 2022, doi: 10.1093/tse/tdac026.

[8] M. Carranza-García, J. Torres-Mateo, P. Lara-Benítez, and J. García-Gutiérrez, “On the Performance of One-Stage and Two-Stage Object Detectors in Autonomous Vehicles Using Camera Data,” Remote Sensing, vol. 13, no. 1, Art. no. 1, Jan. 2021, doi: 10.3390/rs13010089.

[9] A. A. Mustapha and M. S. Yoosuf, “Exploring the efficacy and comparative analysis of one-stage object detectors for computer vision: a review,” Multimed Tools Appl, vol. 83, no. 20, pp. 59143–59168, Jun. 2024, doi: 10.1007/s11042-023-17751-2.

[10] S. K. Pal, A. Pramanik, J. Maiti, and P. Mitra, “Deep learning in multi-object detection and tracking: state of the art,” Appl Intell, vol. 51, no. 9, pp. 6400–6429, Sep. 2021, doi: 10.1007/s10489-021-02293-7.

[11] L. Zhang, F. Yang, Y. Daniel Zhang, and Y. J. Zhu, “Road crack detection using deep convolutional neural network,” in 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA: IEEE, Sep. 2016, pp. 3708–3712. doi: 10.1109/ICIP.2016.7533052.

[12] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, “Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Images: Road damage detection and classification,” Comput-aided Civ Inf, vol. 33, no. 12, pp. 1127–1141, Dec. 2018, doi: 10.1111/mice.12387.

[13] A. C. Aqsa, H. Mahmudah, and R. W. Sudibyo, “Detection and Classification of Road Damage Using CNN with Hyperparameter Optimization,” in 2022 6th International Conference on Informatics and Computational Sciences (ICICoS), Semarang, Indonesia: IEEE, Sep. 2022, pp. 101–104. doi: 10.1109/ICICoS56336.2022.9930607.

[14] V. Mandal, L. Uong, and Y. Adu-Gyamfi, “Automated Road Crack Detection Using Deep Convolutional Neural Networks,” in 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA: IEEE, Dec. 2018, pp. 5212–5215. doi: 10.1109/BigData.2018.8622327.

[15] A. Alfarrarjeh, D. Trivedi, S. H. Kim, and C. Shahabi, “A Deep Learning Approach for Road Damage Detection from Smartphone Images,” in 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA: IEEE, Dec. 2018, pp. 5201–5204. doi: 10.1109/BigData.2018.8621899.

[16] S. A. Hassan, S. H. Han, and S. Y. Shin, “Real-time Road Cracks Detection based on Improved Deep Convolutional Neural Network,” in 2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), London, ON, Canada: IEEE, Aug. 2020, pp. 1–4. doi: 10.1109/CCECE47787.2020.9255771.

[17] N. Camilleri and T. Gatt, “Detecting road potholes using computer vision techniques,” in 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania: IEEE, Sep. 2020, pp. 343–350. doi: 10.1109/ICCP51029.2020.9266138.

[18] A. Angulo, J. A. Vega-Fernández, L. M. Aguilar-Lobo, S. Natraj, and G. Ochoa-Ruiz, “Road Damage Detection Acquisition System Based on Deep Neural Networks for Physical Asset Management,” in Advances in Soft Computing, vol. 11835, L. Martínez-Villaseñor, I. Batyrshin, and A. Marín-Hernández, Eds., in Lecture Notes in Computer Science, vol. 11835. , Cham: Springer International Publishing, 2019, pp. 3–14. doi: 10.1007/978-3-030-33749-0_1.

[19] D. Jeong, “Road Damage Detection Using YOLO with Smartphone Images,” in 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA: IEEE, Dec. 2020, pp. 5559–5562. doi: 10.1109/BigData50022.2020.9377847.

[20] S. Anand, S. Gupta, V. Darbari, and S. Kohli, “Crack-pot: Autonomous Road Crack and Pothole Detection,” in 2018 Digital Image Computing: Techniques and Applications (DICTA), Canberra, Australia: IEEE, Dec. 2018, pp. 1–6. doi: 10.1109/DICTA.2018.8615819.

[21] R. Bibi et al., “Edge AI-Based Automated Detection and Classification of Road Anomalies in VANET Using Deep Learning,” Computational Intelligence and Neuroscience, vol. 2021, pp. 1–16, Sep. 2021, doi: 10.1155/2021/6262194.

[22] Q. Chen, X. Gan, W. Huang, J. Feng, and H. Shim, “Road Damage Detection and Classification Using Mask R-CNN with DenseNet Backbone,” Computers, Materials & Continua, vol. 65, no. 3, pp. 2201–2215, 2020, doi: 10.32604/cmc.2020.011191.

[23] R. Fan, H. Wang, Y. Wang, M. Liu, and I. Pitas, “Graph Attention Layer Evolves Semantic Segmentation for Road Pothole Detection: A Benchmark and Algorithms,” Ieee T Image Process, vol. 30, pp. 8144–8154, 2021, doi: 10.1109/TIP.2021.3112316.

[24] Y. Liu, X. Zhang, B. Zhang, and Z. Chen, “Deep Network For Road Damage Detection,” in 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA: IEEE, Dec. 2020, pp. 5572–5576. doi: 10.1109/BigData50022.2020.9377991.

[25] V. Pham, C. Pham, and T. Dang, “Road Damage Detection and Classification with Detectron2 and Faster R-CNN,” in 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA: IEEE, Dec. 2020, pp. 5592–5601. doi: 10.1109/BigData50022.2020.9378027.

[26] T. Rateke, K. A. Justen, and A. Von Wangenheim, “Road Surface Classification with Images Captured From Low-cost Camera - Road Traversing Knowledge (RTK) Dataset,” RITA, vol. 26, no. 3, pp. 50–64, Nov. 2019, doi: 10.22456/2175-2745.91522.

[27] 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, Art. no. 7, Jul. 2023, doi: 10.3390/machines11070677.

[28] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, “A survey on ensemble learning,” Frontiers of Computer Science, vol. 14, no. 2, pp. 241–258, Apr. 2020, doi: 10.1007/s11704-019-8208-z.

[29] W. Ding et al., “An Ensemble of One-Stage and Two-Stage Detectors Approach for Road Damage Detection,” in 2022 IEEE International Conference on Big Data (Big Data), Dec. 2022, pp. 6395–6400. doi: 10.1109/BigData55660.2022.10021000.

[30] K. Doshi and Y. Yilmaz, “Road Damage Detection using Deep Ensemble Learning,” in 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA: IEEE, Dec. 2020, pp. 5540–5544. doi: 10.1109/BigData50022.2020.9377774.

[31] S. Wang et al., “An Ensemble Learning Approach with Multi-depth Attention Mechanism for Road Damage Detection,” in 2022 IEEE International Conference on Big Data (Big Data), Dec. 2022, pp. 6439–6444. doi: 10.1109/BigData55660.2022.10021018.

[32] V. Hegde, D. Trivedi, A. Alfarrarjeh, A. Deepak, S. Ho Kim, and C. Shahabi, “Yet Another Deep Learning Approach for Road Damage Detection using Ensemble Learning,” in 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA: IEEE, Dec. 2020, pp. 5553–5558.

[33] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 6626-6635.

[34] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, and Y. Sekimoto, “Rdd2022: A multi-national image dataset for automatic road damage detection,’’ Geoscience Data Journal, vol. 11, no. 4, pp. 846–862, 2024.

[35] Sanjeewani, P., Verma, B., “Single class detection-based deep learning approach for identification of road safety attributes,” Neural Comput & Applic 33, 9691–9702. Feb. 2021.