

Enhancing Road Safety: A Deep Learning Approach to Detect and Classify Road Damage

Md. Mahadi Hasan¹ and Dr. Boshir Ahmed²

^{1,2}Department of Computer Science and Engineering

Rajshahi University of Engineering and Technology, Rajshahi-6204, Bangladesh.

Email- mahadihasan1604020@gmail.com, boshir78@gmail.com

Abstract—Detecting and classifying damaged regions of road images is a critical task in computer vision. It's particularly important for self-driving cars. We used two datasets named RDD-2020 and RDD-2022 to train our models. Several earlier approaches using this dataset were unable to reliably identify multiple types of damage within a single image. To improve this, we tried five different models for both RDD-2020 and RDD-2022 datasets. These models are composed of a feature extractor or backbone network and a detection head or detection network. We utilized ResNet-50, ResNet-101, and MobileNetv3 as feature extractors, and Faster RCNN (Region-based Convolution Neural Network), SSD (Single-Shot Detection), and YOLO (You Only Look Once) as detection heads for identifying damaged areas and classifying these damaged areas. The top-performing model was YOLOv8 with ResNet-50. Other models include YOLOv7 with ResNet-50, Faster RCNN with ResNet-50, Faster RCNN with ResNet-101, and SSD with MobileNetv3. We adjust our best model using different techniques to make it even better. We found that using a specific combination of optimizer and batch size resulted in the best performance for detecting different types of road damage. We achieved a better F1 score for YOLOv8 with ResNet-50 using a batch size of 4 and Adam optimizer using the same RDD-2020 and RDD-2022 datasets. We found an F1 score of 0.62 for RDD-2022 and 0.60 for RDD-2020.

Index Terms—Road damage detection, YOLOv8, Faster RCNN, ResNet-50.

I. INTRODUCTION

Road transportation is the primary means of public mobility in urban areas, but it is fraught with challenges, particularly regarding road safety. One of the leading causes of these accidents is poor road conditions, making the frequent inspection and maintenance of roads a critical task for protecting human life. However, traditional methods of manual road inspection are labor-intensive, time-consuming, and costly, necessitating the development of automated solutions to identify road damages efficiently. The rise of autonomous vehicles in modern cities has further amplified the need for automated road surface damage detection. Detecting and repairing damaged roads promptly is essential to maintaining safe and efficient transportation. Although current manual inspection processes are slow and prone to errors, recent advancements in computer vision and deep learning technologies offer promising automated solutions for accurately identifying road damage. In this study, we explored several state-of-the-art detection algorithms and feature extraction models to improve road damage detection. Using two datasets, RDD-2020 and RDD-2022, we combined

detection networks such as YOLO, Faster RCNN, and SSD with feature extractors like ResNet-50 and ResNet-101. Our experiments demonstrated that the combination of YOLOv8 with ResNet-50 produced the best results for detecting and classifying damaged regions. Road damage detection can be divided into three primary tasks: classification, detection, and segmentation. While classification determines the presence of damage, detection identifies its location, and segmentation accurately defines the damage boundaries. Detecting multiple overlapping damaged regions within a single image remains a complex challenge, but by employing advanced object detection techniques, we were able to improve the precision of damage identification and classification.

Our findings can be encapsulated as follows-

- We trained our model with both RDD-2020 and RDD-2022 datasets.
- Our model demonstrated improved F1 scores of 0.64 for D00, 0.58 for D10, 0.63 for D20, and 0.62 for D40 damage classes using RDD-2022 dataset compared to the previous approach [1]. We also achieved a F1-Score of 0.62 for D00, 0.58 for D10, 0.61 for D20, and 0.60 for D40 damage classes using RDD-2020 dataset which was better than [2].
- Our model is capable of identifying and classifying multiple damage regions that overlap within a single image.

II. RELATED WORK

Deep learning offers a promising approach for identifying road damage automatically. By analyzing images, these state-of-art algorithms can pinpoint the exact location, type, and extent of damage, facilitating efficient and accurate detection. M.M. Hasan et al. [2] proposed using different models for RDD-2020. They utilized Faster RCNN, SSD, and EfficientDet as detection networks, and employed ResNet-101, MobileNet v1, MobileNet v2, and EfficientNet as backbone feature extractors. The best F1 score was achieved using Faster RCNN with ResNet-101, with a batch size of 4 and the optimizer set as SGD with momentum. There are two main approaches to classification and detection: single-stage methods and two-stage methods

Single-Stage Detection Algorithm. Road damage detection and classification has been a recent research focus.

D. Arya et al., 2020 [3] developed two SSD models, one using MobileNet v2 and the other using Inception v2, for road damage detection and classification. They trained these models on the RDD-2020 dataset. The models occasionally struggled to accurately detect multiple damage types within the same area. D. Arya et al., 2021 [4] The author of the previous study [3] improved the SSD model by using ResNet-50 and adjusting its parameters. This led to slightly better results than their earlier work for RDD-2020. Jiang et al. [5] used the state-of-the-art YOLO detection model along with two techniques: combining multiple models and creating more training data by data augmentation. Silva, Luís Augusto,et al., 2023 [6] introduced a new method for automatically detecting road damage using drone images and artificial intelligence. The authors propose using three different models, YOLOv4, YOLOv5, and YOLOv7, to classify and locate damaged regions of road in drone images. They trained and tested these models using a dataset called RDD-2022. The YOLOv5 model achieved 59.9% accuracy, the YOLOv5 model with some fine-tuning achieved 65.70% accuracy, and the YOLOv7 model achieved 73.20% accuracy. Yang et al., 2024 [7] presented a new method for detecting road damage using an improved version of YOLOv8. Seo et al., 2024 [8] collected 13,500 images taken with digital, infrared (IR), and MSX cameras. They used a model called DarkNet-56 with YOLOv8 to automatically classify these images. The model was most accurate with digital images (97.4%), followed by MSX images (91.1%), and then IR images (80.1%). They also developed a method to determine the direction of cracks using wavelet transforms. Zhang et al., 2024 [1] introduced a new, fast algorithm called FPDDN for detecting road damage. FPDDN combines deformable transformers, D2f blocks, and SFB modules to accurately detect damage of different sizes. Deformable transformers help the model adapt to different shapes of damage, improving accuracy for irregular defects like cracks. D2f blocks make the model smaller and faster, and SFB modules help preserve information about small-sized damage. This combination reduces information loss for small damage and improves the connection between different layers of the model. They tested the model on the RDD-2022 dataset to evaluate its speed and accuracy. The result was an F1-score of 0.601, outperforming other models. Li, Yue, et al., 2024 [9] presented a new and improved version of YOLOv8 called RDD-YOLO for detecting road damage. The model uses a technique called SimAM to help it focus on important details in the images, improving its ability to identify road damage. They enhance its ability by modifying the model's neck structure to reduce redundancy.

Two-Stage Detection Algorithm. uses the Region Proposal Network (RPN) to process the input image before the feature extractor. The RPN generates region proposals of varying sizes, which are then standardized by the ROI pooling layer. This layer converts these region proposals into the same size. These region proposals are finally taken by the Feature extractor. Single-stage detection algorithm skips the Region

Proposal Network stage. Li et al., 2023 [10] presented a new method for detecting road damage using a modified version of Faster-RCNN. The last layer of the model is changed to a larger receptive field convolution. The authors use several techniques to fine-tune the model. Kulambayev, B. et al., 2023 [11] presented a new method using a model called Mask R-CNN. The model is specifically designed to recognize and categorize damaged regions. The authors trained the model using a vast dataset of road images from different locations and conditions. While these methods have achieved promising results, there is still a need for faster detection algorithms to meet real-world requirements. To overcome the limitation of previous papers and to achieve better accuracy we used both single and two-stage detection algorithms.

III. DATASET DESCRIPTION

We have used two benchmark datasets, the first one is RDD-2020 [12] and the second one is RDD-2022 [13]. Both the datasets have 4 different classes as illustrated in Figure 1.

Longitudinal Cracks (D00). These cracks run parallel to the road's centerline, often appearing as long, straight lines.

Transverse Cracks (D10). These cracks run perpendicular to the road's centerline, typically appearing as short, straight lines crossing the road.

Alligator Cracks (D20). A network of interconnected cracks resembling alligator scales, often appearing as a series of small, intersecting cracks.

Pothole(D40). Depressions in the road surface are caused by wear and tear, varying in size and depth but typically having a rounded or oval shape.

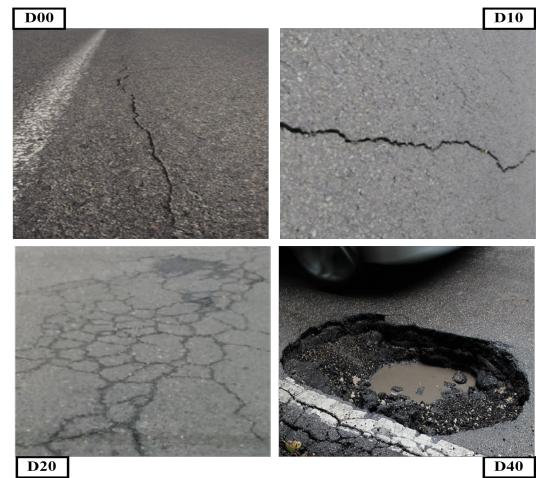


Fig. 1. Sample images illustrating the four damage classes.

A. RDD-2020

The RDD-2020 dataset comprises 26,336 road images sourced from India, Japan, and the Czech Republic. This dataset contains over 31,000 instances of road damage, annotated across four distinct damage categories. These images,

captured using cell phone cameras mounted on vehicles, offer a cost-effective solution for road monitoring initiatives undertaken by cities and road authorities. Table I provides a detailed breakdown of the data statistics for RDD-2020, categorized by damage type and country. Figure 2. presents a visual representation of the dataset's geographical distribution across the three countries.

TABLE I
STATISTICAL OVERVIEW OF THE RDD-2020 DATASET

	Japan	India	Czech Republic	Total
Total Images	13133	9892	3595	26620
D00	5006	1994	1242	8242
D10	4911	89	480	5480
D20	7846	2556	211	10613
D40	2798	3947	263	7008
Total Instances	20516	8586	2196	31343

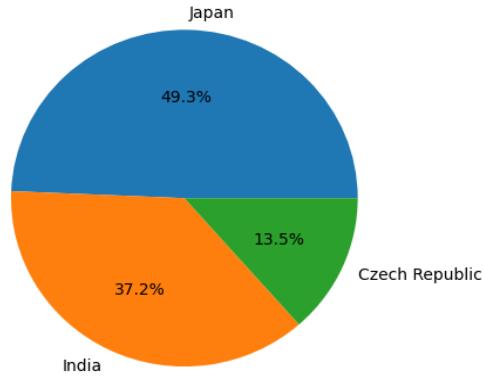


Fig. 2. Percentage distribution of images by country in the RDD-2020 dataset.

B. RDD-2022

The RDD-2022 dataset, an expansion of RDD-2020, encompasses a comprehensive collection of 47,420 road images from six countries: India, Japan, the Czech Republic, Norway, the United States, and China. This dataset includes over 55,000 instances of road damage, categorized into four distinct damage types, aligning with the RDD-2020 classification. The Japanese portion of the dataset features images captured from seven diverse cities, including both urban and cold regions. Indian images represent a mix of metropolitan and rural road conditions. The Czech Republic contributes images from various road types in Olomouc, Prague, and Bratislava. Norwegian data is sourced from expressways and county roads. The dataset further benefits from the inclusion of images from the United States, sourced from Google Street View, and China, encompassing aerial drone footage and

smartphone images captured under diverse lighting conditions. This broad geographical and environmental diversity enhances the dataset's suitability for training and evaluating robust road damage detection models. Table II presents the data statistics for RDD-2022 by damage category, highlighting the number of images and instances across four classes in six countries. Figure 3 provides a pie chart representation of the dataset distribution among these six countries.

TABLE II
STATISTICAL OVERVIEW OF THE RDD-2022 DATASET

	Japan	India	Czech Republic	Norway	USA	China	Total
Total Images	13133	9665	3538	10201	6005	4878	47420
D00	4049	1555	988	8570	6750	4104	26016
D10	3979	68	399	1730	3295	2359	11830
D20	6199	2021	161	468	834	934	10617
D40	2243	3187	197	461	135	321	6544
Total Instances	16470	6831	1745	11229	11014	7718	55007

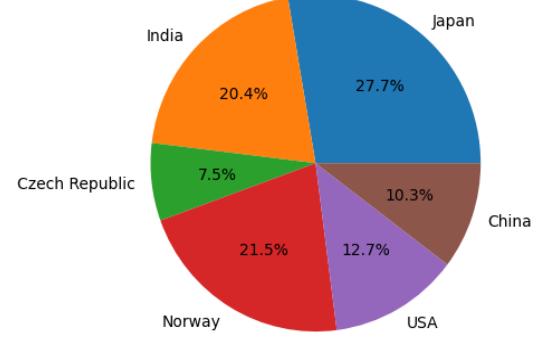


Fig. 3. Percentage distribution of images by Country in the RDD-2022 dataset.

IV. METHODOLOGY

This section outlines the proposed framework for identifying and categorizing road damage. The system processes input images, detects damaged regions with bounding boxes, and predicts a class and confidence score for each. State-of-the-art pre-trained transfer learning models, such as ResNet-50, ResNet-101, and MobileNetV3, serve as backbone networks to extract feature maps from preprocessed images. Detection networks like YOLO, Faster RCNN, and SSD are used to generate bounding boxes and classifications. Figure 4 illustrates the overall architecture, divided into three stages: preprocessing, feature extraction (backbone network), and detection network. Among the models tested, ResNet-50 achieved the highest F1 score. The output layer uses a sigmoid activation function, with ReLU for hidden layers. The bounding box regressor employs

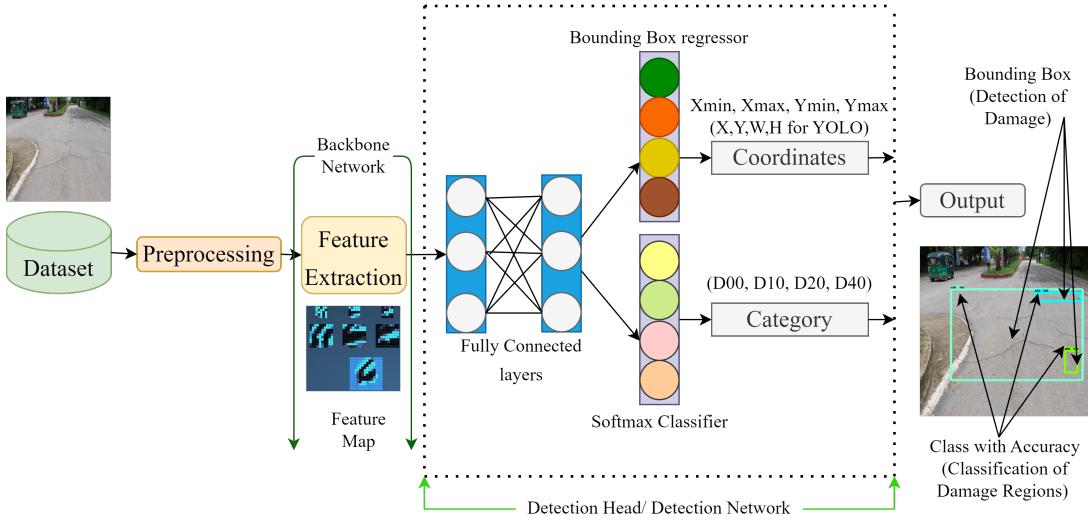


Fig. 4. Proposed Methodology of our Model.

Mean Squared Error (MSE) as the loss function, while Sparse Categorical Cross Entropy is used for classification. The final output includes bounding boxes, corresponding classes, and confidence scores. The three stages of the methodology are detailed below.

A. Data Preprocessing

To preprocess the RDD-2020 and RDD-2022 datasets, we resized the images into 448×448 to standardize their size and reduce computational costs, as different transfer learning models have varying input requirements. We converted the annotated XML files into CSV format and further processed them by converting them into TensorFlow Records (TFRecords) using UTF-8 encoding. This record contains the annotated information of the damaged type. To speed up training, we normalized the pixel values between 0 and 1 using the Min-Max normalization technique, as shown in equation 1.

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (1)$$

B. Feature Extraction

To extract spatial features, we experimented with various deep-learning architectures. We selected popular transfer learning models like -

ResNet-50 is a very effective backbone pre-trained transfer learning network. It has 50 layers that use convolutions. Deep neural networks with many layers can sometimes overfit and have too many parameters. They might also face the vanishing gradient problem. Residual networks solve this by adding skip connections. These connections can be identity blocks or convolutional blocks. Identity blocks are used for layers with equal input and output dimensions. Convolutional blocks are used when they are different.

ResNet-101 The main difference of ResNet-50 and ResNet-101 is that in 4th convolution block, ResNet-101 has 23 co-evolution layers, whereas ResNet-50 has 6 convolution layers.

The rest of the convolution layers are similar to ResNet-50. ResNet-101 also overcomes the vanishing gradient issue by incorporating skip connections

MobileNet-v3 has 78 layers that use depth-wise convolutions. Each layer has a depth of 5. This means that 5 depth-wise convolution layers are grouped together and then used like a regular point-wise convolution layer.

C. Detection Network

We have experimented with YOLOv8, Faster RCNN, and SSD as detection networks. YOLOv8 provides the best accuracy.

Faster RCNN is a two-stage detector characterized by the inclusion of a region proposal network that proposes the regions where objects might be present. The ROI pooling layer reshapes feature maps and regions to a uniform size. Faster RCNN, a detection head, was used, and its fully connected network was learned from the extracted feature maps. The bounding box regressor component within the detection head predicted the four coordinate values (X_{\max} , X_{\min} , Y_{\max} , Y_{\min}) defining the bounding box, while the softmax classification component assigned probabilities to each class.

SSD is a single-stage detection model. It consists of six convolutional layers. SSD predicts 8732 bounding boxes for each object. To choose the best ones, SSD uses Non-Max Suppression to remove duplicates. SSD ranks each box by confidence and selects the top 200 for each image. During training, SSD compares these predicted boxes to the actual object locations (ground truth). It uses the Intersection over Union (IoU) metric to measure the overlap between predicted and ground truth bounding boxes.

YOLO is another single-stage detector that takes an input image, resizes to 448×448 , and divides into 7×7 grid cells. Each grid cell is designed to predict a maximum of B bounding boxes, with a corresponding confidence score assigned to each

box. Out of B, only one with the highest confidence score is kept. In our experiment, we considered B=2. If the center of a damaged region falls within a grid cell, that cell is tasked with detecting and localizing that damage using a bounding box. For each bounding box, a five-element target vector (X, Y, W, H, C) is calculated. X and Y represent the coordinates of the bounding box's top-left corner, while W and H denote its width and height and C represents the confidence score. For faster training YOLO normalized X, Y, W, and H by using the equations shown in Figure 5.

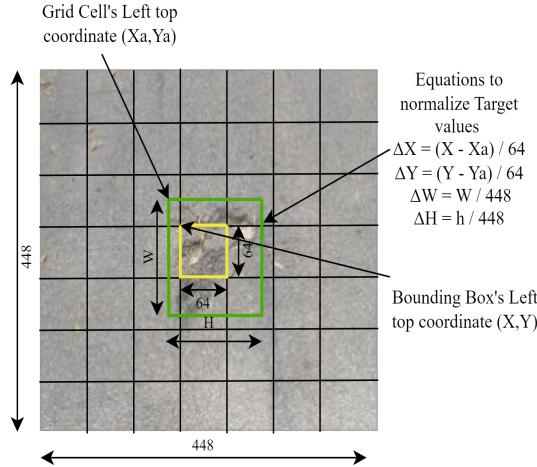


Fig. 5. Normalization scheme for target vectors (X, Y, W, H) in YOLO.

The Confidence score (C) depends on how likely an object is in the box and how well the box matches the actual object's location. The confidence score is the product of these two factors: the probability of an object being present and the overlap between the predicted and actual boxes. The equation of confidence score is shown in 2.

$$\text{confidence score (C)} = p(\text{object}) \cdot \text{IoU}_{\text{pred}}^{\text{truth}} \quad (2)$$

Here, $p(\text{object})$ represents the probability of an object being present, ranging from 0 to 1, with 0 indicating no object. $\text{IoU}_{\text{pred}}^{\text{truth}}$ denotes the intersection-over-union between the predicted and ground truth bounding boxes. By calculating a particular class confidence score, we select the bounding box of that cell which is responsible for detecting the damage region. Our feature extractor ResNet-50 generates $7 \times 7 \times 1024$ to 50176 feature vectors which are passed through two fully connected layers. The loss function is evaluated, and the model's parameters are updated iteratively using the back-propagation algorithm. While training, we used 8th version of YOLO.

V. EXPERIMENTAL RESULT

To evaluate the performance of different object detection models in identifying road damage, we conducted experiments using two datasets: RDD-2022 and RDD-2020. The models

were evaluated on three key metrics: Precision, Recall, and F1-Score, which are commonly used to measure the effectiveness of detection algorithms in terms of accuracy and balance between false positives and false negatives. Table III provides a comprehensive comparison of the performance of YOLOv8, YOLOv7, Faster RCNN, and SSD models, combined with various backbone networks, including ResNet-50, ResNet-101, and MobileNetv3, across both datasets.

TABLE III
MODEL PERFORMANCE COMPARISON ON RDD-2022 AND RDD-2020 DATASETS

Dataset	Model Name	Backbone Network	Precision	Recall	F1-Score
RDD-2022	YOLOv8	ResNet-50	0.58	0.67	0.62
	YOLOv7	ResNet-50	0.57	0.62	0.59
	Faster RCNN	ResNet-50	0.53	0.57	0.55
	SSD	MobileNetv3	0.51	0.49	0.5
RDD-2020	YOLOv8 [1]	DarkNet-56	0.56	0.65	0.60
	YOLOv8	ResNet-50	0.55	0.65	0.60
	YOLOv7	ResNet-50	0.57	0.61	0.59
	Faster RCNN	ResNet-50	0.59	0.49	0.54
	Faster RCNN [2]	ResNet-101	0.58	0.39	0.47
	SSD	MobileNetv3	0.50	0.53	0.52

Performance on RDD-2022. For the RDD-2022 dataset, YOLOv8 with ResNet-50 as the backbone achieved the best performance, with a Precision of 0.58, Recall of 0.67, and an F1-Score of 0.62. YOLOv7, also using ResNet-50, followed closely with a Precision of 0.57 and Recall of 0.62, yielding an F1-Score of 0.59. Faster RCNN with ResNet-50 showed a slightly lower performance, with an F1-Score of 0.55, and SSD with MobileNetv3 recorded the lowest performance among all the models, with an F1-Score of 0.5. Our model achieved a better F1-Score than the previous model [1]. We fine-tuned our model with different optimizers and batch sizes and found the best result using Adam with a batch size of 4.

Performance on RDD-2020. Similarly, for the RDD-2020 dataset, YOLOv8 with ResNet-50 achieved the best overall results, with a Precision of 0.55, Recall of 0.65, and an F1-Score of 0.60. YOLOv7, again with ResNet-50, showed comparable performance, with an F1-Score of 0.59. Faster RCNN using ResNet-50 and ResNet-101 produced F1-Scores of 0.54 and 0.47 [2]. Lastly, SSD with MobileNetv3 yielded an F1-Score of 0.52. we also fine-tuned our models for RDD-2020 dataset and found a better F1 score than the previous model [2].

A. Analysis of Results

Figure 6 shows the confusion matrix that illustrates the performance of YOLOv8 with ResNet-50 on the RDD-2022 test data. Our model correctly identified 3486 samples as D00 but misclassified 1714 as D10, D20, or D40. Additionally, it accurately classified 1585 samples as D10 and 1422 as

D20. Finally, 878 samples were correctly classified as D40. From the experimental results, it is evident that YOLOv8

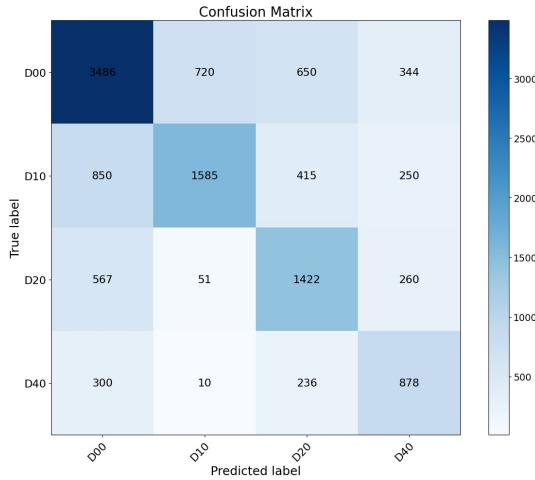


Fig. 6. YOLOv8-ResNet-50 confusion matrix (RDD-2022)

outperforms the other detection models across both datasets, achieving the highest precision, recall, and F1-Score values. YOLOv7 also performs well but slightly trails behind YOLOv8. While Faster RCNN demonstrates competitive performance, it falls short compared to YOLO-based models. SSD with MobileNetv3 shows the weakest performance across both datasets, indicating that more lightweight models like SSD may not be as effective in this particular task of road damage detection. Finally, it is clear that YOLOv8 with ResNet-50 consistently delivers the best balance between precision and recall across both RDD-2022 and RDD-2020 datasets, making it the most suitable model for road damage detection in our experiments. Figure 7 illustrates the visualization of the damaged areas detected by our model. These results highlight the importance of both the detection network and the choice of backbone for improving road damage detection accuracy.

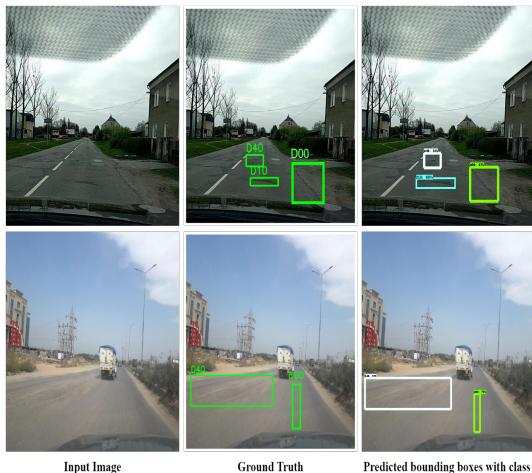


Fig. 7. Input image with ground truth and predicted bounding boxes.

VI. CONCLUSION

In this research work, we have developed an automated road damage detection system using YOLOv8 with ResNet-50, achieving high accuracy in detecting and classifying road damage on the RDD-2020 and RDD-2022 datasets. Our model consistently outperformed others, particularly in identifying overlapping damage regions within a single image, with superior Precision, Recall, and F1-score metrics. This approach offers a more efficient alternative to manual inspections, enhancing road safety and supporting the demands of autonomous vehicles. Future work will focus on expanding the model to handle more complex damage types and improving its real-time performance for continuous road monitoring.

REFERENCES

- [1] Y. Zhang and C. Liu, "Real-time pavement damage detection with damage shape adaptation," *IEEE Transactions on Intelligent Transportation Systems*, 2024.
- [2] M. M. Hasan, S. Sakib, and K. Deb, "Road damage detection and classification using deep neural network," in *2022 4th International Conference on Electrical, Computer Telecommunication Engineering (ICECTE)*, 2022, pp. 1–6.
- [3] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, A. Mraz, T. Kashiyama, and Y. Sekimoto, "Transfer learning-based road damage detection for multiple countries," *arXiv preprint arXiv:2008.13101*, 2020.
- [4] D. Arya, H. Maeda, S. Ghosh, D. Toshniwal, A. Mraz, T. Kashiyama, and Y. Sekimoto, "Deep learning-based road damage detection and classification for multiple countries," *Automation in Construction*, vol. 132, p. 103935, 12 2021.
- [5] Y. JIANG, "Road damage detection and classification using deep neural networks," 2024.
- [6] L. A. Silva, V. R. Q. Leithardt, V. F. L. Batista, G. V. González, and J. F. D. P. Santana, "Automated road damage detection using uav images and deep learning techniques," *IEEE Access*, vol. 11, pp. 62918–62931, 2023.
- [7] J. Yang and Y. Li, "Research on road damage detection based on improved yolov8," *Academic Journal of Science and Technology*, vol. 9, no. 3, pp. 162–167, 2024.
- [8] H. Seo, Y. Shi, and L. Fu, "Automatic damage detection of pavement through darknet analysis of digital, infrared, and multi-spectral dynamic imaging images," *Sensors*, vol. 24, no. 2, p. 464, 2024.
- [9] 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, 2024.
- [10] S. Li and Y. Huang, "Damage detection algorithm based on faster-rcnn," in *2023 5th International Conference on Electronics and Communication, Network and Computer Technology (ECNCT)*. IEEE, 2023, pp. 177–180.
- [11] B. Kulambayev, M. Nurlybek, G. Astaubayeva, G. Tleuberdiyeva, S. Zholdasbayev, and A. Tolep, "Real-time road surface damage detection framework based on mask r-cnn model," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 9, 2023.
- [12] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, and Y. Sekimoto, "Rdd2020: An annotated image dataset for automatic road damage detection using deep learning," *Data in brief*, vol. 36, p. 107133, 2021.
- [13] D. Arya, H. Maeda, S. Ghosh, D. Toshniwal, and Y. Sekimoto, "Rdd2022: A multi-national image dataset for automatic road damage detection," 09 2022.