Improving Road Safety through Deep Learning-based Approaches for Road Damage Detection and Classification

Md. Mahadi Hasan
Dept. of CSE
Rajshahi University of Engineering
and Technology, Rajshahi-6204, Bangladesh

Boshir Ahmed
Dept. of CSE
Rajshahi University of Engineering
and Technology, Rajshahi-6204, Bangladesh

ABSTRACT

Transportation heavily depends on roads, which require proper maintenance for safe and efficient travel. Traditional manual inspection methods are time consuming, labor-intensive and pose safety risks. To address these challenges, Deep learning models are presented for road damage detection using two benchmark datasets: RDD-2020 and RDD-2022. This study compares five models integrating feature extractors such as ResNet-50, ResNet-101, and MobileNetv3 with detection frameworks like Faster R-CNN, SSD, and YOLO. Among them, YOLOv10 achieved the best performance. Fine-tuning with the Adam optimizer and a batch size of four improved its F1 scores to 0.67 for RDD-2022 and 0.63 for RDD-2020. Additionally, the CrackBD-2024 dataset was developed, consisting of 2,038 images with 5,060 instances from Bangladeshi roads, to enhance the generalization of the models. This work contributes to advancing road damage detection, improving monitoring, and facilitating better maintenance planning.

General Terms

Computer Vision, Image Processing

Keywords

Road Damage Detection, Faster RCNN, YOLO, SSD, ResNet-50, ResNet-101

1. INTRODUCTION

Efficient and safe transportation depends on the proper maintenance of road infrastructure. Traditional methods for detecting road damage, such as visual inspections for cracks and potholes, are time-consuming, labor-intensive, and prone to human error. Additionally, these inspections pose significant safety risks for personnel. In Bangladesh, where urban roads are a vital component of public transportation, road accidents remain a major concern. A 2024 report by the Road Safety Foundation recorded 6,927 road accidents, resulting in 7,294 fatalities and 12,019 injuries. To improve road safety, regular inspections and timely maintenance are essential.

Advancements in artificial intelligence, particularly deep learning, provide a promising solution for automating road damage detection through image analysis. This technology significantly enhances accuracy, speeds up the process, and reduces costs compared to conventional methods. Automated detection also improves road maintenance planning and resource allocation, ultimately contributing to safer and more efficient transportation systems.

This study explores advanced detection algorithms and feature extraction models to enhance the accuracy of road damage detection. Using the RDD-2020 and RDD-2022 datasets, Detection networks such as YOLO, Faster R-CNN, and SSD were integrated with feature extractors like ResNet-50, ResNet-101, and MobileNet v3. Among these architectures, YOLOv10 demonstrated the highest accuracy in identifying and classifying damaged regions.

Road damage detection involves three primary tasks: classification, detection, and segmentation. Classification determines whether damage is present, detection identifies its location, and segmentation defines its precise boundaries. A key challenge in this domain is accurately identifying multiple overlapping damaged areas within a single image. Advanced object detection techniques improved the precision of damage identification and classification, effectively addressing these challenges. The findings of this study can be encapsulated as follows-

- Introduced a custom dataset named “CrackBD-2024”, featuring samples collected from roads in Bangladesh.
- Utilized the RDD-2020 and RDD-2022 datasets for extensive research and analysis.
- Developed an improved YOLO-based model, YOLO v10 that achieved an F1-Score of 0.67 on the RDD-2022 dataset and 0.63 on the RDD-2020 dataset, outperforming previous approaches [1] [2].
- Evaluated the models using the CrackBD-2024 dataset, highlighting their effectiveness in detecting road damage in local conditions.

2. RELATED WORK

Object detection in computer vision has made significant advancements in recent years. Two-stage detectors, such as Fast R-CNN, Faster R-CNN, and Mask R-CNN, along with single-stage detectors like SSD and YOLO, have played a crucial role in this progress. Faster R-CNN, with its Region Proposal Network (RPN), provides high accuracy but is computationally intensive. In contrast, SSD offers a faster approach by directly predicting bounding boxes and class probabilities within an image. The YOLO family, particularly YOLOv8 and YOLOv10, is widely used for its real-time performance and high accuracy. In road damage detection, researchers have explored various techniques to assess and classify road surface defects accurately. Faster R-CNN, SSD, and YOLO-based models have been effectively applied to this task. However, balancing accuracy and inference speed remains a significant challenge.

Single Stage Detection. Dipankar Dutta et al. [3] emphasize the importance of cost-effective road maintenance strategies to enhance driving safety. Traditional road damage detection methods are expensive, leading to the exploration of smartphone-based imaging and machine learning techniques, particularly YOLO algorithms (YOLOv3–YOLOv7). The research demonstrates improved accuracy and efficiency in road damage detection by incorporating data augmentation methods, including GANs, and optimizing YOLOv6 through hyperparameter tuning. Notably, the optimized YOLOv6 model maintains high accuracy despite significant size reduction, making it a practical solution for resource-constrained applications. Nik Ahmad Farihin Mohd Zulkifli et al. [4] aim to improve road safety in autonomous vehicles by enhancing road damage detection. The researchers utilized the YOLOv8 model and optimized its hyperparameters using the Salp Swarm Algorithm (SSA), achieving a notable 3.5% enhancement in accuracy. The model was trained on the RDD2022 dataset, which includes data from multiple countries, such as the Czech Republic, India, and China. The findings demonstrate that SSA-based hyperparameter optimization can significantly enhance the performance of YOLOv8 in detecting road damage, ultimately contributing to safer autonomous driving systems. Devesh Mothilall et al. [5] utilized the Indian subset of the Road Damage Dataset (RDD) 2022, which contains diverse street images. After data processing and labeling, a YOLOv5s model was trained, validated, and tested on 1,959 images, achieving an F1 score of 41% for road damage detection. The study recommends using the broader Global RDD 2022 dataset in future research to develop more robust and accurate models. Yiwen Jiang et al. [6] present an optimized YOLOv8 model for road damage detection. Tested on the RDD2022 dataset, the model achieved an mAP50 of 62.5%, mAP50-95 of 36.4%, and an F1 score of 69.6%, outperforming the baseline by 2.5%, 5.2%, and 2.8%, respectively. These improvements enhance detection accuracy and efficiency, reducing resource requirements while supporting effective road maintenance and safety assessments. Vaishnavee V. Rathod et al. [7] introduces RDD-YOLO, an enhanced YOLOv8-based algorithm for road damage detection. The model improves feature extraction by integrating the Simple Attention Mechanism (SimAM), which helps focus computational resources on critical image areas. Additionally, the neck structure is optimized using GhostConv to reduce computational complexity, while nearest-neighbor upsampling is replaced with bilinear interpolation to enhance visual detail retention. These enhancements address challenges in detecting pavement defects such as cracks, potholes, and rutting. Tian-Yi Jiang et al. [8] introduce YOLOv5s-Road, a novel model incorporating advancements such as the MHSA mechanism, a 1D convolution block, and the ASFF module. Additionally, the VariFocal loss function effectively addresses class imbalance. YOLOv5s-Road surpasses five leading models in performance on the RDD2022 dataset.

Two-stage detection. Qihan He et al. [9] propose a novel LSF-RDD model for road damage detection on a custom dataset. Faster R-CNN demonstrated superior performance, achieving the highest overall mean average precision (mAP) of 88.49%, with an exceptional mAP of 96.62% for focused damage classes. While exhibiting slightly lower overall performance (86.47% mAP), EfficientDet D1 proved to be a strong contender, achieving 95.12% mAP for focused classes while demonstrating lower computational requirements. These findings highlight the potential of EfficientDet D1 for developing accurate and computationally efficient RDD systems. Vaishnavee V. Rathod et al. [10] introduce a novel deep-learning framework for automated road crack detection. The approach is based on a hybrid feature extraction architecture that combines a Pyramid Vision Transformer with a ConvMixer, allowing the model to capture intricate crack patterns effectively. To enhance image quality, median filtering is applied during preprocessing. The detection and classification of cracks are then performed using an Elman neural network optimized by an improved black widow algorithm. This framework addresses critical challenges such as varying damage sizes and image blur, improving the accuracy and reliability of road crack detection. Robert G. De Luna et al. [11] present a road damage detection system leveraging Convolutional Neural Networks (CNNs) to classify road conditions into normal cracks and potholes. Three models with varying input sizes and convolutional layers were implemented using Python with Keras and TensorFlow. Among them, Model 3 (Faster RCNN) exhibited superior performance with the highest accuracy, lowest error rate, and best specificity, precision, and F1 score. In testing, Model 3 (Faster RCNN) and Model 1 (YOLO v7) accurately classified seven out of nine road images, while Model 2 correctly classified six, demonstrating the system's potential to enhance road safety and infrastructure management. Muhammad Waseem Khan et al. [12] leverage the RDD2022 dataset, which comprises images captured by UAVs and vehicle dashboard cameras, to train and evaluate road damage detection (RDD) models. Data augmentation techniques were employed to mitigate class imbalance within the dataset. The research encompasses both pure and mixed model architectures, assessing the performance of advanced detectors, including the two-stage detection algorithm Faster R-CNN with a ResNet101 backbone and one-stage models such as SSD MobileNet V1 FPN, YOLOv5, and EfficientDet D1.

3. DATASET DESCRIPTION

Two benchmark datasets, RDD-2020 [13] and RDD-2022 [14], were utilized, both comprising four distinct classes, as depicted in Figure 1. In addition, a custom dataset named "CrackBD-2024" was prepared, containing the same four classes as described below. The models were also evaluated using this dataset.

Longitudinal Cracks (D00): Longitudinal cracks are elongated fractures that run parallel to the road's centerline. These cracks are typically caused by stresses such as uneven roadbed settlement or thermal expansion and contraction. They often appear as long, straight lines and can vary in width and depth. If left untreated, longitudinal cracks can expand over time, leading to further road deterioration and potential safety hazards.

Transverse Cracks (D10): Transverse cracks are perpendicular to the road's centerline, creating short, straight lines that cross the road. These cracks often result from thermal stresses, especially in regions with significant temperature fluctuations. They may also form due to the shrinkage of pavement materials. Transverse cracks can compromise the road's structural integrity and allow water infiltration, accelerating road damage.

Fig. 1: Sample images illustrating the four damage classes

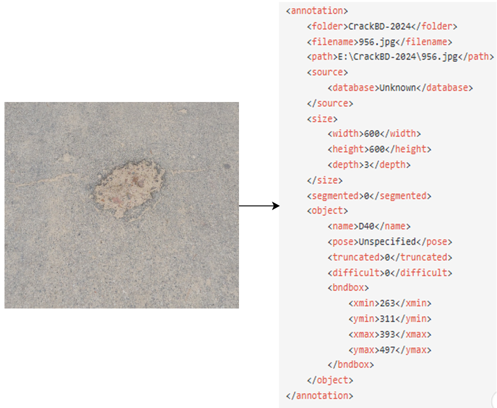

Fig. 2: Annotation output in xml format derived from raw image

Alligator Cracks (D20): Alligator cracks, also known as fatigue cracks, form a network of interconnected fractures resembling the scales of an alligator's skin. These cracks are typically caused by repeated traffic loads that exceed the pavement's capacity, leading to structural fatigue. Alligator cracking is often indicative of severe underlying issues, such as inadequate pavement thickness or base layer failure. If not repaired, it can quickly escalate into larger structural failures, such as potholes.
Potholes (D40): Potholes are depressions or cavities in the road surface caused by the combined effects of wear and tear, water infiltration, and freeze-thaw cycles. They vary in size and depth but are generally rounded or oval in shape. Potholes form when water seeps into cracks in the pavement, weakening the underlying layers. When vehicles pass over these weakened areas, the pavement collapses, creating a hole. Potholes pose significant risks to vehicle safety and comfort, potentially causing damage to tires, suspension systems, and even accidents.

3.1 CrackBD-2024

A total of 2,038 images were collected from Bangladeshi road surfaces, forming the dataset named CrackBD-2024. This dataset contains 5060 instances of four distinct classes of road damage: longitudinal cracks (D00), transverse cracks (D10), alligator cracks (D20), and potholes (D40), consistent with the classification scheme employed in the well-established RDD-2020 and RDD-2022 datasets. Each damage instance is meticulously annotated with bounding boxes and corresponding labels, enabling precise training of object detection and classification algorithms. The "Labellmg" annotation tool was used for damage class labeling, and the annotation files are in .xml format. The dataset is available at the following link: CrackBD-2024 Dataset
Data collection and annotation: High-resolution images of damaged roads were collected via smartphone and preprocessed for uniformity. These images were resized to ensure consistency and renamed from 1 to 2,038 sequentially for easy identification and management. Annotations were created using the "Labellmg" tool in .xml format and then converted to .csv for better organization. To optimize compatibility with TensorFlow's object detection pipeline, the .csv files were transformed into TFRecords, ensuring a seamless workflow from data collection to model training. The initial format for saving annotations was .xml, as shown in Figure 2.

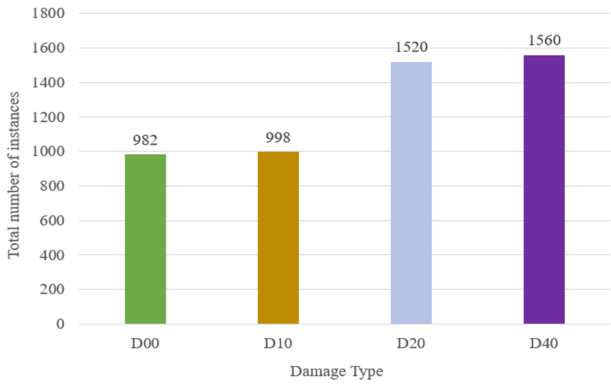

Fig. 3: Bar chart illustrating the number of instances across four different classes in the CrackBD-2024 dataset

Data Statistics: The bar chart in Figure 3 illustrates the total number of instances for various types of road damage, classified as D00, D10, D20, and D40. The chart reveals that D40 has the highest number of instances, totaling 1560, followed closely by D20 with 1520 instances. In contrast, D00 and D10 have significantly fewer instances, with 982 and 998, respectively. Each bar is color-coded to represent a specific damage type, with the y axis indicating the total number of instances and the x axis denoting the damage types. The exact counts are displayed above each bar for added clarity, effectively highlighting the distribution of types of road damage.

4. METHODOLOGY

This section outlines a framework, shown in Figure 4, to detect and categorize road damage in images. The system generates bounding boxes and assigns class labels with confidence scores. Pre-trained models are used for feature extraction, along with detection networks such as YOLO, Faster R-CNN, and SSD. The architecture has three stages: image preprocessing, feature extraction, and damage detection. The output layer uses a sigmoid activation function, with hidden layers utilizing ReLU. MSE is used for bounding box regression, and Sparse Categorical Cross Entropy is used for classification. The system outputs bounding boxes, predicted classes, and confidence scores.

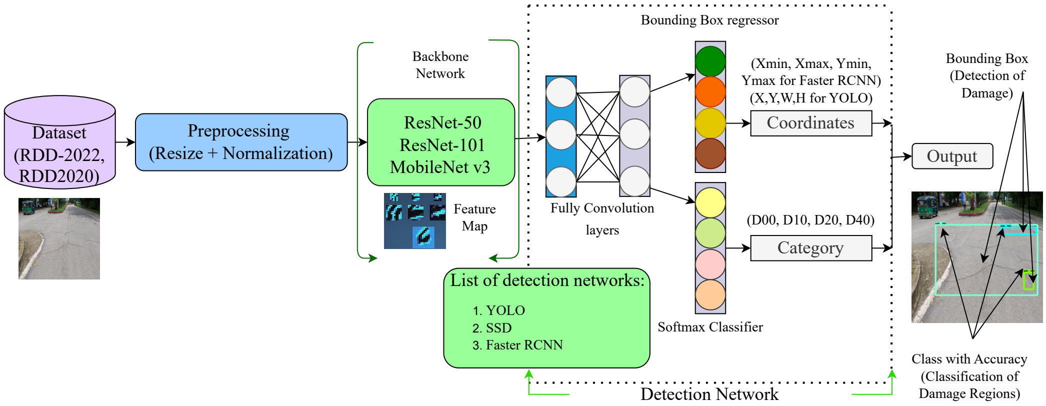

Fig. 4: Proposed methodology of the model

4.1 Data Preprocessing

The RDD-2020 and RDD-2022 datasets were pre-processed to ensure consistency and optimize training. Images were resized to 448×448 pixels, and annotations in xml format were converted to csv and then to TFRecords with UTF-8 encoding for efficient data handling. The pixel values were normalized to a range of 0 to 1 using Min-Max normalization as shown in equation 1, improving the stability of the model and the efficiency of training.

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

4.2 Backbone Network

To extract spatial features for this task, several deep learning architectures were experimented with popular transfer learning models such as ResNet-50, ResNet-101, and MobileNet-v3. Below is a detailed explanation of each.

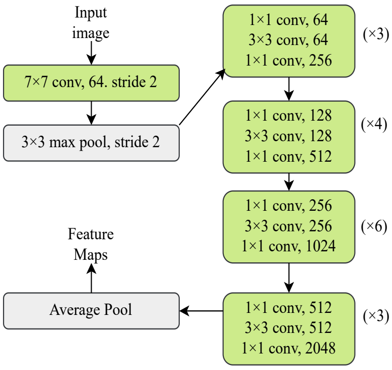

Fig. 5: Layer architecture of ResNet-50

ResNet-50 [15] is a highly effective pre-trained backbone network with 50 layers of convolution operations, as shown in Figure 5. Deep neural networks often struggle with overfitting, excessive parameters, and vanishing gradient problems. ResNet-50 addresses these challenges using skip connections to maintain gradient flow.

There are two main types of skip connections: identity blocks and convolutional blocks,

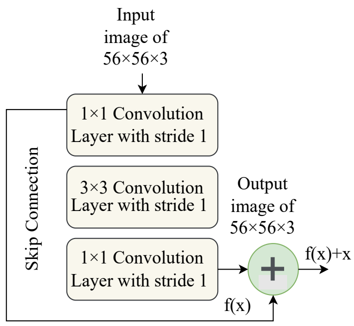

Fig. 6: The identity block: A core component of the ResNet-50 architecture

Identity Block: An identity block is used when the input and output dimensions of a convolutional block are the same. For example, in Figure 6, the input and output sizes are both $56 \times 56 \times 3$ because the three convolutional layers have a stride of 1, preserving the image dimensions. The identity block utilizes skip connections to help mitigate the vanishing gradient issue.

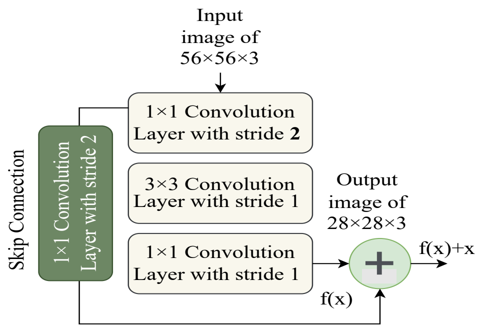

Fig. 7: The convolutional block: A core component of the ResNet-50 architecture

Convolutional block: When the input and output of a convolutional block are not the same, a convolutional block is used. For example, in Figure 7, the input size is $56 \times 56 \times 3$, and the output size is $28 \times 28 \times 3$, as there are two convolution layers with stride one and one convolution layer with stride two. The stride of 2 reduces the image dimensions by approximately half, following the equation 2

$$\mathtt{F} = \frac{(n + 2p - f)}{s} + 1 \tag{2}$$

Here, F is the output of the convolution layer, n is the input size of the image, p means padding, f means filter size, and s means stride. So, $\frac{56+0-3}{2} + 1 = 28$. Since the stride is 2, the image dimensions are effectively reduced.

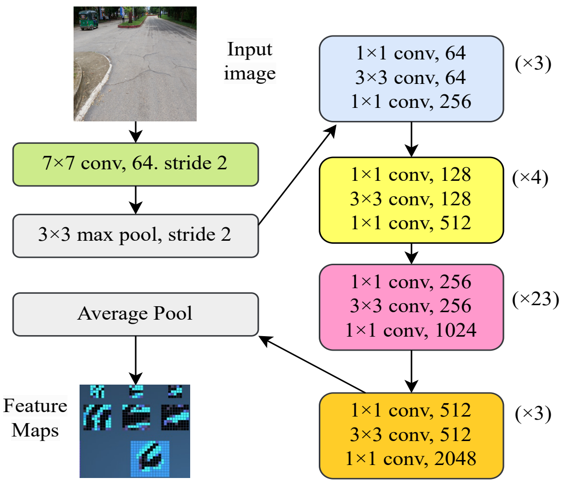

Fig. 8: Architecture of the pre-trained ResNet-101 backbone network

ResNet-101 [16] is a deep learning architecture with 101 convolutional layers, as shown in Figure 8, compared to ResNet-50's 50 layers. This increased depth enhances its ability to extract features for more complex tasks. However, it also faces challenges like vanishing gradients and overfitting. To mitigate these, ResNet-101 uses residual connections with identity blocks for matching dimensions and convolutional blocks. A key difference lies in the fourth convolutional block: ResNet-101 has 23 layers, while ResNet-50 has only 6, which boosts its ability to handle more intricate tasks.
MobileNet v3 [17] is a lightweight architecture designed for resource-limited settings. It achieves high accuracy with 78 layers and depth-wise separable convolutions, which reduce computational complexity. The architecture optimizes efficiency by grouping five depth-wise convolutions into a unit paired with a point-wise convolution. This design enables effective spatial and channel-wise feature processing, making it ideal for balancing performance and resource usage.

4.3 Detection Head

The detection head receives feature maps from the backbone network. It uses these feature maps to learn models and generate bounding boxes with confidence scores to identify damaged regions. The final layer consists of a bounding box layer and a softmax classifier layer. The bounding box layer generates the coordinates of the bounding boxes, while the softmax classifier layer assigns a corresponding class and generates a class confidence score. The methods differ by algorithm. Faster R-CNN, SSD, and YOLO were evaluated, with each employing unique strategies for effective damage detection and classification.

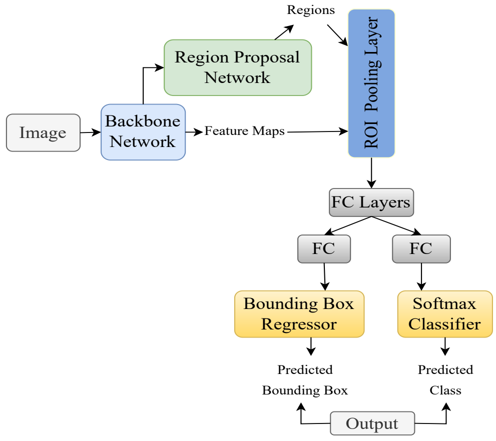

Fig. 9: A deep dive into the Faster R-CNN architecture

Faster R-CNN, shown in Figure 9, is a two-stage object detection algorithm known for its accuracy and versatility. In the first stage, the Region Proposal Network (RPN) generates region proposals by creating anchor boxes of varying sizes and filtering them based on Intersection over Union (IoU) scores. These regions are of different sizes. The task of the pre-trained backbone transfer learning model is to generate feature maps of varying sizes. The Region of Interest (ROI) pooling layer is used to convert these varying sizes feature maps and regions into a fixed dimension. In the second stage, fully connected layers are used to learn the features, and based on the learning, the softmax classifier classifies regions, while the bounding box regressor layer predicts the four coordinate values of the bounding boxes.

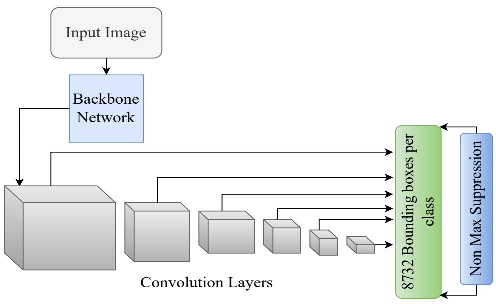

Fig. 10: The architectural components of the Single Shot Detector (SSD)

Single Shot Detection (SSD) [18] is shown in Figure 10, detects objects and predicts their locations by processing input images through a backbone network and multiple convolutional layers, generating 8,732 bounding boxes per object. To refine the results, SSD uses Non-Maximum Suppression (NMS) to eliminate overlaps and selects the top 200 predictions based on confidence scores. During training, SSD matches predicted boxes with ground truth using Intersection over Union (IoU), retaining those with an overlap above 0.6.

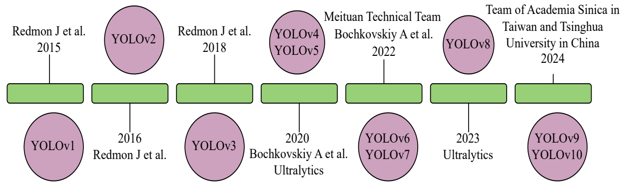

Fig. 11: The progression of the YOLO algorithm over time

The YOLO (You Only Look Once) series [19] shown in Figure 11, has evolved significantly for real-time object detection. YOLOv1 introduced a grid-based approach but struggled with detecting small and overlapping objects. YOLOv2 added anchor boxes, batch normalization, and multi-scale training to improve accuracy. YOLOv3 enhanced multi-scale detection with Darknet-53 and feature pyramid networks. YOLOv4 leveraged CSPDarknet53, IoU-based loss, and advanced augmentation techniques. YOLOv5 emphasized usability with a modular PyTorch framework, while YOLOv6 focused on industrial use with anchor-free detection. YOLOv7 introduced dynamic label assignment and attention modules. YOLOv8 offers anchor-free detection, C2f modules, and adaptive mosaic augmentation for improved speed and accuracy.

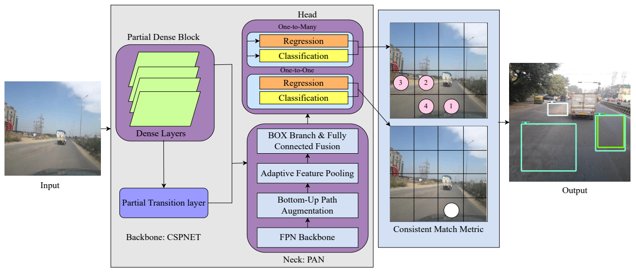

Fig. 12: Architecture of YOLOv10.

YOLOv10 is an advanced object detection model designed to enhance both speed and accuracy through various improvements. It incorporates a CSPNet backbone for efficient feature extraction, a Path Aggregation Network (PAN) in the neck to improve multi-scale feature aggregation, and a C2f module that streamlines gradient flow while minimizing computational redundancy. The head of YOLOv10 produces bounding boxes, class labels, and confidence scores, applying Non-Maximum Suppression (NMS) to remove duplicate predictions. The architecture of YOLOv10 is shown in Figure 12. CSPNet in YOLOv10 improves computational efficiency and feature representation by splitting the feature map into two streams: one bypasses dense computations, and the other transforms through dense layers. These streams merge to enhance gradient flow and reduce vanishing gradients, making YOLOv10 more accurate and versatile for real-time object detection across various scales. PAN in the neck module enhances multi-scale feature aggregation, which is essential for detecting objects of different sizes. By using top-down and bottom-up pathways, PAN refines feature maps and optimizes gradient flow, improving detection accuracy and robustness for both small and large objects. The head of YOLOv10 converts feature maps into bounding box coordinates, class labels, and confidence scores, filtering irrelevant predictions using objectness scores. It employs Softmax for class probability estimation and uses Non-Maximum Suppression (NMS) to refine detections, ensuring fast, accurate, and scalable object detection across various configurations.

5. EXPERIMENT RESULT

To assess the performance of various object detection models in detecting road damage, experiments were conducted using the RDD-2022 and RDD-2020 datasets. The dataset description is provided below:

5.1 RDD-2020

The RDD-2020 dataset, comprising 26,336 high-resolution images from India, Japan, and the Czech Republic, provides a valuable resource for advancing automated road damage detection and classification research. This diverse dataset includes over 31,000 instances of four distinct damage types: longitudinal cracks (D00), transverse cracks (D10), alligator cracks (D20), and potholes (D40). The total image count for the four different classes and three countries in the RDD-2020 dataset is shown in Figure 13, while the total image count for six counties in the RDD-2022 dataset is shown in Figure 14.

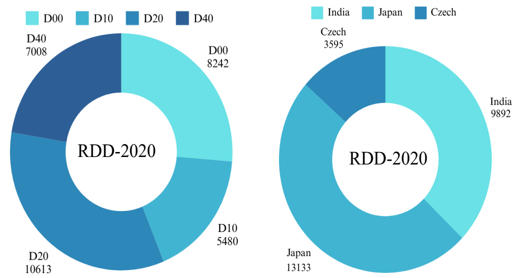

Fig. 13: Image count for four different classes and three different countries in RDD-2020 datasets

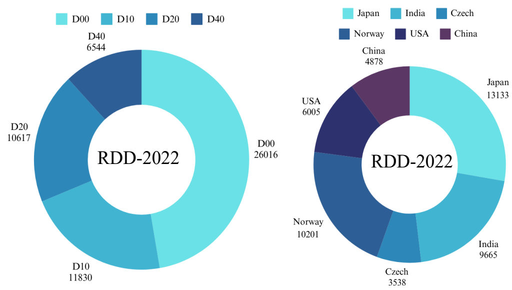

Fig. 14: Image count for four different classes and six different countries in RDD-2022 datasets

5.2 RDD-2022

The RDD-2022 dataset builds upon its predecessor, RDD-2020, by significantly expanding its scope and diversity. It includes a vast collection of 47,420 high-resolution road images gathered from six countries: India, Japan, the Czech Republic, Norway, the United States, and China.
The models were evaluated using three primary metrics: Precision, Recall, and F1-Score shown in equation 3, 4, and 5. These metrics are widely used to assess the accuracy and balance of detection algorithms by accounting for false positives and false negatives.

$$\texttt{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$\texttt{Precision} = \frac{TP}{TP + FP} \tag{4}$$

$$\texttt{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

Table 1. : Performance score of different model

Dataset	Model	Precision	Recall	F1-Score
RDD-2020	YOLOv10	**0.65**	**0.61**	**0.63**
	YOLOv8	0.64	0.57	0.60
	Faster RCNN+ResNet-50	0.58	0.59	0.59
	Faster RCNN+ResNet-101 [1]	0.59	0.60	0.60
	SSD+MobileNet v3	0.54	0.52	0.53
RDD-2022	YOLOv10	**0.67**	**0.68**	**0.67**
	YOLOv8 [2]	0.64	0.62	0.63
	Faster RCNN+ResNet-50	0.59	0.62	0.60
	Faster RCNN+ResNet-101	0.58	0.64	0.61
	SSD+MobileNet v3	0.54	0.57	0.56

Table 2. : Performance score on CrackBD-2024 dataset

Dataset	Model	Precision	Recall	F1-Score
CrackBD-2024	YOLOv10	0.61	0.58	0.59
	YOLOv8	0.57	0.56	0.56
	Faster RCNN with ResNet-50	0.54	0.61	0.57
	Faster RCNN with ResNet-101	0.57	0.60	0.58
	SSD with MobileNet v3	0.48	0.51	0.50

5.3 Result Analysis

Table 1 compares the performance of the YOLOv10, YOLOv8, Faster R-CNN, and SSD models with different backbone networks (ResNet-50, ResNet-101, MobileNetv3) across the RDD-2020, RDD-2022, and Crack-BD-2024 datasets. YOLOv10 consistently outperformed the others, achieving the highest F1 scores of 0.63 on RDD-2020 and 0.67 on RDD-2022. The models outperform previous models for both datasets [1] [2]. The Crack-BD-2024 dataset led to lower performance in all models compared to the RDD datasets, with YOLOv10 achieving the highest F1 score of 0.59. This drop is mainly due to the dataset's limited sample images. Crack-BD-2024 introduces valuable challenges, including diverse crack patterns and real-world factors like lighting and occlusions. The confusion matrix of YOLOv10 on the RDD-2022 dataset is shown in Figure 15. The loss of YOLOv10 training and validation in the RDD-2022 data set is shown in Figure 16.

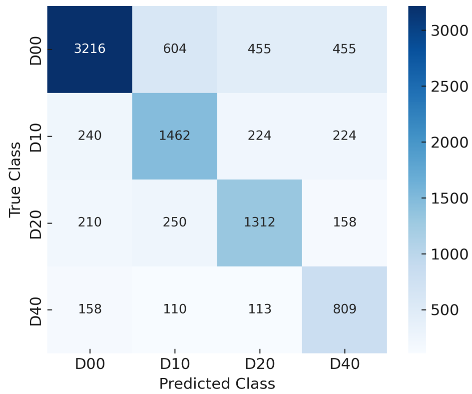

Fig. 15: Confusion Matrix of the best model on RDD-2022 dataset

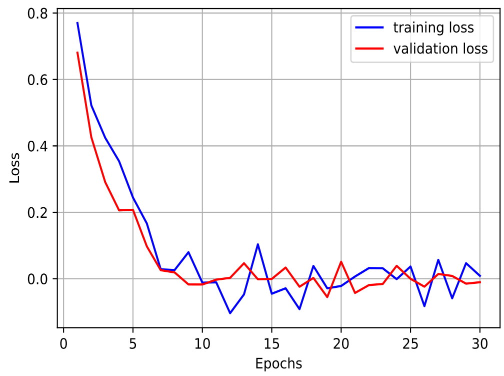

Fig. 16: Training vs Validation Loss

Figure 17 shows the detections of different models in the same image, highlighting their varying abilities to handle complex and overlapping damage scenarios. Expanding the Crack-BD-2024 dataset with more samples and diversity is key to improving accuracy and ensuring reliable performance in future research.

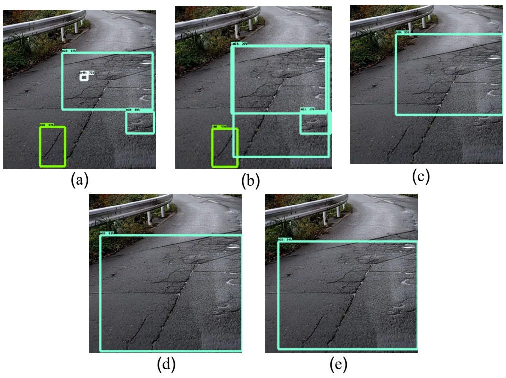

Fig. 17: Detection and classification result of five models in the same image (a) YOLOv10 (b) YOLOv8 (c) Faster RCNN with ResNet-50 (d) Faster RCNN with ResNet-101 (e) SSD with MobileNet-v3

6. CONCLUSION

In this study, an automated system for detecting and classifying road damage was developed using YOLOv10, achieving exceptional precision on the RDD-2020 and RDD-2022 datasets. The model demonstrated superior performance, particularly in identifying overlapping damage regions within a single image, as reflected in its high precision, recall, and F1 score metrics. In addition, the system was tested on a custom dataset, CrackBD-2024, further validating its effectiveness. By offering a faster and more accurate alternative to manual inspections, this approach improves road safety and meets the needs of autonomous vehicles. Future research will improve the ability of the model to handle complex damage scenarios and improve real-time performance for continuous road monitoring. To increase accuracy, the models will be retrained using a custom dataset with more images from Bangladeshi roads, addressing limitations caused by initial training in images from foreign roads (RDD-2022 and RDD-2020). This work also holds potential applications in the Bangladesh transportation system, such as monitoring road surface.

7. REFERENCES

[1] Md Mahadi Hasan, Saadman Sakib, and Kaushik Deb. Road damage detection and classification using deep neural network. In *2022 4th International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)*, pages 1–6. IEEE, 2022.

[2] Yingchao Zhang and Cheng Liu. Real-time pavement damage detection with damage shape adaptation. *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[3] Dipankar Dutta and Rudranath Banerjee. Indian road damage detection using deep learning. In *Advancements in Intelligent Process Automation*, pages 223–262. IGI Global, 2025.

[4] Nik Ahmad Farihin Mohd Zulkifli, Zuriani Mustaffa, and Mohd Herwan Sulaiman. Road damage detection for autonomous driving vehicles using yolov8 and salp swarm algorithm. *Applications of Modelling and Simulation*, 9:1–11, 2025.

[5] Devesh Mothilall and Terence van Zyl. Improving road damage maintenance in south africa using deep learning. In *2024 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA)*, pages 1–5. IEEE, 2024.

[6] Yiwen Jiang. Road damage detection and classification using deep neural networks. *Discover Applied Sciences*, 6(8):421, 2024.

[7] Yue Li, Chang Yin, Yutian Lei, Jiale Zhang, and Yiting Yan. Rdd-yolo: Road damage detection algorithm based on improved you only look once version 8. *Applied Sciences*, 14(8):3360, 2024.

[8] Tian-Yi Jiang, Zhi-Yang Liu, and Guang-Zhu Zhang. Yolov5s-road: Road surface defect detection under engineering environments based on cnn-transformer and adaptively spatial feature fusion. *Measurement*, 242:115990, 2025.

[9] Qihan He, Zhongxu Li, and Wenyuan Yang. Lsf-rdd: a local sensing feature network for road damage detection. *Pattern Analysis and Applications*, 27(3):99, 2024.

[10] Vaishnavee V Rathod, Dipti P Rana, and Rupa G Mehta. Deep learning-driven uav vision for automated road crack detection and classification. *Nondestructive Testing and Evaluation*, pages 1–30, 2025.

[11] Robert G De Luna, Mariano G Abejo, Beatriz M Aldave, Ruth Hannah P Barrete, Jewel L Lim, Hazel M Mendoza, Myka Sydney Morcoso, and Karina L Enriquez. Enhancing road safety: A cnn-based approach to road damage classification. In *2024 IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 1–6. IEEE, 2024.

[12] Muhammad Waseem Khan, Mohammad S Obaidat, Khalid Mahmood, Dania Batool, Hafiz Muhammad Sanaullah Badar, Muhammad Aamir, and Wu Gao. Real-time road damage detection and infrastructure evaluation leveraging unmanned aerial vehicles and tiny machine learning. *IEEE Internet of Things Journal*, 2024.

[13] Deeksha Arya, Hiroya Maeda, Sanjay Kumar Ghosh, Durga Toshniwal, and Yoshihide Sekimoto. Rdd2020: An annotated image dataset for automatic road damage detection using deep learning. *Data in brief*, 36:107133, 2021.

[14] Deeksha Arya, Hiroya Maeda, Sanjay Kumar Ghosh, Durga Toshniwal, and Yoshihide Sekimoto. Rdd2022: A multi-national image dataset for automatic road damage detection. *Geoscience Data Journal*, 11(4):846–862, 2024.

[15] Brett Koonce and B Efficientnet Koonce. *Convolutional neural networks with swift for tensorflow: Image recognition and dataset categorization*. Springer, 2021.

[16] Qi Zhang. A novel resnet101 model based on dense dilated convolution for image classification. *SN Applied Sciences*, 4:1–13, 2022.

[17] Prasad SP Kavyashree and Mohamed El-Sharkawy. Compressed mobilenet v3: a light weight variant for resource-constrained platforms. In *2021 IEEE 11th annual computing and communication workshop and conference (CCWC)*, pages 0104–0107. IEEE, 2021.

[18] Alexander Womg, Mohammad Javad Shafiee, Francis Li, and Brendan Chwyl. Tiny ssd: A tiny single-shot detection deep convolutional neural network for real-time embedded object detection. In *2018 15th Conference on computer and robot vision (CRV)*, pages 95–101. IEEE, 2018.

[19] Mujadded Al Rabbani Alif and Muhammad Hussain. Yolov1 to yolov10: A comprehensive review of yolo variants and their application in the agricultural domain. *arXiv preprint arXiv:2406.10139*, 2024.