A Thesis Submitted to the Sylhet Engineering College for the Degree of
Bachelor of Science in **Computer Science and Engineering**

A Study on Real-Time Pavement Crack Detection Using YOLOv9: Challenges and Opportunities

By

Sabib Ahmed

Jannatul Ferdus Aspia

Supervised By

Md. Abu Naser Mojumder

Assistant Professor

Department of Computer Science and Engineering

Sylhet Engineering College, Sylhet

22th July, 2025

Sylhet Engineering College, Sylhet

Affiliated with

Shahjalal University of Science & Technology (SUST)

RECOMMENDATION LETTER FROM THESIS SUPERVISOR

The thesis titled **“A Study on Real-Time Pavement Crack Detection Using YOLOv9: Challenges and Opportunities”** submitted by the group as mentioned below has been accepted as satisfactory in partial fulfilment of the requirements for the degree B. Sc. in Computer Science and Engineering in July, 2025.

Group Members:

Sabib Ahmed (2019331540)

Jannatul Ferdus Aspia (2019331554)

Supervisor:

Md. Abu Naser Mojumder

Assistant Professor and Head
Department of Computer Science and Engineering
Sylhet Engineering College, Sylhet.

CERTIFICATES OF ACCEPTANCE

The thesis is titled **“A Study on Real-Time Pavement Crack Detection Using YOLOv9: Challenges and Opportunities”** submitted by **Sabib Ahmed** and **Jannatul Ferdus Aspia**; Student ID. **2019331540** and **2019331554**; Session **2019-20**, to the Department of Computer Science and Engineering, Sylhet Engineering College, has been accepted as satisfactory in partial fulfilment of the requirement for the Degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

BOARD OF EXAMINERS

Internal
Nayan Kumar Nath
Lecturer
Department of Computer Science and Engineering
Sylhet Engineering College, Sylhet.

Internal
Md. Lysuzzaman
Lecturer
Department of Computer Science and Engineering
Sylhet Engineering College, Sylhet.

Internal
Md. Rasel Ahmed
Assistant Professor
Department of Computer Science and Engineering
Sylhet Engineering College, Sylhet.

Internal
Md. Nazrul Islam
Assistant Professor
Department of Computer Science and Engineering
Sylhet Engineering College, Sylhet.

Chairman
Md. Abu Naser Mojumder
Assistant Professor and Head
Department of Computer Science and Engineering
Sylhet Engineering College, Sylhet.

Member (External)
Mohammad Shahidur Rahman, Ph.D., SMIEEE
Professor
Department of Computer Science and Engineering
Shahjalal University of Science and Technology

ACKNOWLEDGEMENTS

First and foremost, we offer our deepest gratitude to the Almighty, whose boundless mercy and silent guidance have been our constant source of strength throughout this journey. Through every challenge and every success, His blessings have illuminated our path and sustained our resolve.

We would like to express our heartfelt appreciation to our respected supervisor, **Md. Abu Naser Mojumder**, for his invaluable support, continuous encouragement, and insightful feedback. His guidance has been pivotal in shaping the direction and quality of our research, and we are truly grateful for his mentorship.

Our sincere thanks also go to our respected teachers, **Md. Lysuzzaman** and **Nayan Kumar Nath**, whose dedication to teaching and depth of knowledge have left a lasting impact on our academic foundation. Their support and encouragement throughout our studies have been a source of inspiration.

We are especially thankful for the strong collaboration and mutual understanding we shared as thesis partners. The joint effort in completing our thesis, "**A Study on Real-Time Pavement Crack Detection Using YOLOv9: Challenges and Opportunities**," was made possible through shared dedication, teamwork, and a commitment to learning.

Lastly, with all our love and gratitude, we acknowledge the endless support of our families. Their unconditional love, sacrifices, and constant prayers have been the backbone of our academic journey. Without their unwavering belief in us, this achievement would not have been possible.

Abstract

For efficient road maintenance and infrastructure safety, pavement crack detection is essential. Nevertheless, manual inspection techniques are time-consuming, prone to mistakes, and unsuitable for extensive monitoring. An automated crack detection system utilizing the cutting-edge deep learning model YOLOv9 is presented in this work. Four different types of cracks were identified in the 1,200 annotated pavement images that made up the custom dataset: pothole, longitudinal, transverse, and alligator. Real-world variations in road conditions, surface textures, and lighting environments are captured in the dataset. This dataset was used to train and assess YOLOv9, which produced an F1-score of 73.2% and a mean Average Precision (mAP) of 70.4%. These outcomes reveal how well YOLOv9 performs in providing precise, scalable, and instantaneous crack detection. By facilitating prompt and well-informed decision-making by road authorities and civil engineers, the suggested system has the potential to greatly improve road safety and maintenance workflows.

Keywords: Crack Detection, Deep Learning, YOLO, Image Processing, Faster R-CNN, Data Augmentation

Table of Content

List of Figures

List of Tables

Section 1

Introduction

1.1 Background

Bridges, tunnels, nuclear power plants, and other civil and social infrastructure projects have been built in China since the 1990s. Because of its affordability, versatility, and wide range of applications, concrete has become one of the most widely used materials in these developments [1]. However, concrete structures may crack due to creep, shrinkage, and applied loads, endangering their structural integrity and safety [2]. Concrete pavements can also be harmed by low tensile strength, water overflow, and temperature fluctuations. Consequently, cracks are created as a result of this damage, endangering the pavement's overall durability and quality. Although most people are at ease driving on roads, low-durability pavements can present significant and unanticipated safety hazards. Road maintenance is an important area of pavement engineering [3]. A flexible pavement structure is typically composed of layers of various materials that increase in strength as users approach the surface [4]. Whereas, a non-flexible pavement structure uses concrete as the primary layer which is prone to cracking if not properly joined or reinforced. In addition, over time, cracks may enlarge and result in a more serious structural condition. These cracks could endanger human life and present serious safety hazards to concrete infrastructure if they are not fixed right away [3]. Cracks that are promptly identified and fixed can improve pavement durability, stop additional deterioration, and lower long-term maintenance expenses.

The civil infrastructure in the majority of developed nations, including Bangladesh, is deteriorating. Common flaws like cracks are typically seen on roads, bridges, dams, etc. [5] Rehabilitation and Maintenance of ageing bridges or roads have become a common problem in different countries [6][7]. In order to improve maintenance plans and guarantee road safety, a lot of research has been conducted since the late 20th century to create efficient techniques for identifying and categorizing different kinds of cracks in asphalt pavements. Road maintenance workers or engineers have manually inspected and evaluated the pavement damage. Conventional manual techniques for crack detection were frequently expensive, time-consuming, and labor-intensive. On the other hand, this field has undergone a revolution due to recent developments in image processing. However, manual feature extraction using standard image processing techniques is impractical due to the complexity of road environments [8]. Consequently, there is increasing interest in creating intelligent and automated systems that use computer vision and machine learning methods to identify pavement cracks. Image-based methods that use deep learning techniques have made it possible to detect cracks quickly, cheaply, and effectively with little assistance from humans.

1.2 Problem Statement

Accurately identifying and classifying different types of cracks in real-world settings is still difficult, even with the development of numerous methods. This disparity restricts the capacity of road maintenance authorities and civil engineers to make prompt and well-informed decisions. Many of the current pavement crack detection systems still have issues in practical applications, despite a great deal of research in this area. Because of differences in pavement texture, lighting, crack shapes, and ambient noise, models trained on public datasets frequently do not generalize across different geographic regions. Furthermore, there is still a lack of research assessing the performance of more sophisticated YOLO (You Only Look Once) models, such as YOLOv9, YOLOv11, and YOLOv12, especially on custom datasets that more accurately represent local road conditions, even though numerous studies have concentrated on well-known models like YOLOv5 and YOLOv8. Thus, the goal of this study is to create a model based on deep learning that can efficiently identify and differentiate between four main types of cracks in pavement images.

By employing a custom dataset of road surface photos gathered from different parts of Bangladesh, covering a range of environmental conditions and crack types, this study seeks to overcome these limitations. The study also assesses the performance of the most recent YOLO models to ascertain which is best suited for detecting pavement cracks in the real world.

1.3 Objectives

The following are the main goals of this study:

- To produce a unique dataset of photos of pavement cracks that accurately depict the conditions in Bangladesh.
- To use this dataset to train and assess the YOLOv9, YOLOv11, YOLOv12 and Faster R-CNN models' performance.
- To evaluate and contrast each model's mAP, F1-score, recall, accuracy, and precision in identifying the longitudinal, transverse, pothole, and alligator crack types.
- To determine which model is best suited for real-world implementation in road monitoring and maintenance systems.

1.4 Scope & Limitations

This study uses deep learning-based object detection models to detect four main types of pavement cracks: longitudinal, transverse, alligator, and pothole cracks. Road photos gathered from different parts of Bangladesh were used to create a custom dataset that reflected the actual state of the local road system. To find out how well YOLO models detect cracks, the research entails training and assessing various iterations of the models, including the newest architectures. Standard evaluation metrics like precision, recall, F1-score, and mAP are used to evaluate the performance.

This study has certain limitations in spite of its contributions. The custom dataset might not optimization well to other nations or regions with different pavement textures and damage patterns, even though it is representative of local road conditions. Model performance may be impacted by environmental factors like lighting variations and image noise, particularly in low light levels. Furthermore, the model’s robustness and optimization in could be further enhanced by adding more images with a variety of conditions and crack patterns, as the dataset used is rather small. Furthermore, the study excludes other types of road damage like potholes and block cracks and only looks at four distinct crack types. Additionally, hardware performance optimization and real-time deployment are outside the purview of this work.

Section 2

Literature Review

2.1 Related Works

The significance of pavement crack detection for road maintenance and safety has drawn a lot of attention. Researchers have investigated both conventional and cutting-edge AI-based methods to automate crack detection during the last ten years.

Muhammad Sohaib et al. [9] examines several YOLO models for concrete crack detection, highlighting how well-suited they are for real-time applications because of their high speed and accuracy. To increase robustness against issues like occlusion and variable crack characteristics, the models were first trained on a general dataset and then refined with real-world data using transfer learning. With a mAP of 74.52% and an inference time of 19.5 ms per image, YOLOv10 outperformed the other tested models, making it the most useful for real-world crack detection situations.

Zhen Yu et. al [10] presents a YOLO V5s-based deep learning method for detecting cracks in concrete structures, addressing the limitations of manual inspection. A dataset of 4000 labeled images was used, with 3500 for training and 500 for testing. To improve accuracy, the authors applied Otsu's thresholding for denoising and used K-Means clustering to optimize anchor box sizes. Mosaic data augmentation was also employed to enhance model robustness, especially for small cracks. The optimized YOLO V5s model achieved significant improvements, with the best results showing an average precision (AP) of 72.75%, recall of 80.82%, F1-score of 76.57%, and mAP@0.5 of 76.26%. The study concludes that these optimizations enhance the model's detection capabilities, though further improvements are needed for detecting small, overlapping, or discontinuous cracks.

The RDD2022(Road Damage Detection 2022) [11] dataset, which is openly accessible and comprises roughly 48,000 road damage photos gathered from various nations, was used by Xuwei Dong et al. [1]. With an F1-score of 93.2%, mAP@50 of 93.5%, and mAP@50:95 of 71.4%, they achieved remarkable results with their improved YOLOv8-based crack detection model.

To create a unique YOLOv5-based model, Abdullah As Sami et al. also used the RDD2022 (Road Damage Detection 2022) dataset [11]. Their method performed moderately well in identifying road surface damage, as evidenced by an F1-score of 66.51% and a mAP@50 value of 67.81%.

Using a custom dataset of road damage photos taken with a smartphone, Norsuzila Ya'acob et al. [3] created a real-time pavement crack detection system. The system incorporates sophisticated image pre-processing methods, data augmentation tactics, and training via the Roboflow platform, and it is based on the YOLOv5 framework. The model exhibits successful detection of both longitudinal and transverse cracks in real-world scenarios, with mean Average Precision (mAP) of 31.5%, precision of 53.7%, and recall of 30.3%.

For their investigation, Syed Ali Hassan et al. [5] used a dataset of road images that had been independently gathered. They created a unique YOLOv3-based model. The model demonstrated strong performance in pavement crack detection, achieving an overall accuracy of 92% and a mean Average Precision (mAP) of 90%.

2.2 Research Gap

Despite the fact that many studies have investigated pavement crack detection using deep learning models like YOLOv3, YOLOv5, and YOLOv8, the majority of them rely on publicly available datasets like RDD2022 or Crack500, which frequently lack regional diversity and do not accurately reflect the unique features of road conditions in many developing nations. There are currently no localized models that are optimized for regional infrastructure since no study has presented a dataset based on road damage photos gathered from Bangladesh. Furthermore, current research mostly concentrates on earlier iterations of object detection models without assessing the improvements brought about by more recent YOLO architectures. In order to fill these gaps, this study uses a specially created dataset of road cracks unique to Bangladesh. It also applies and contrasts updated YOLOv9, YOLOv11, YOLOv12 models along with Faster R-CNN to determine the most accurate pavement crack detection method.

Section 3

Methodology

3.1 Workflow Overview

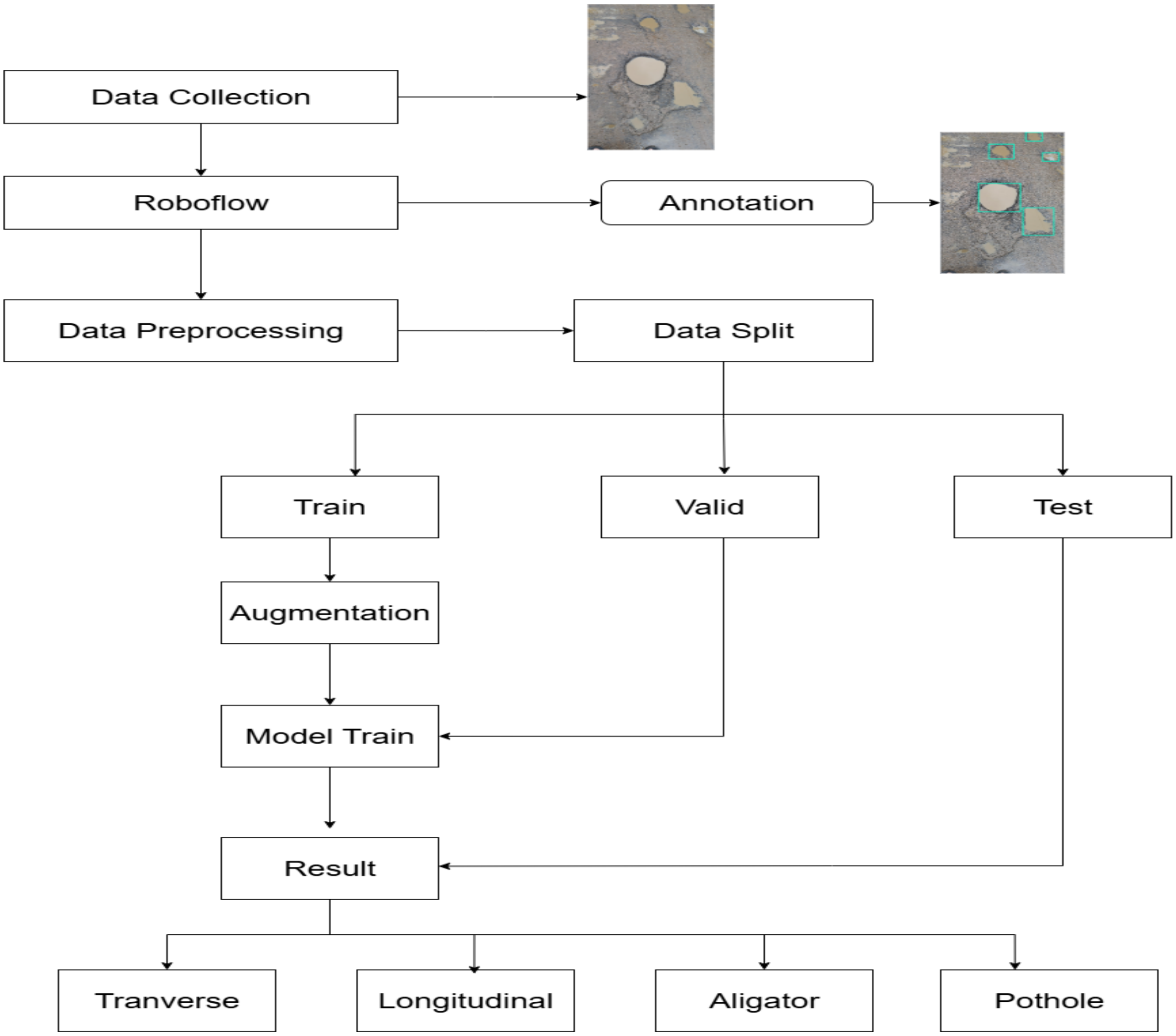


Fig 1. Workflow Diagram

3.2 Dataset Collection

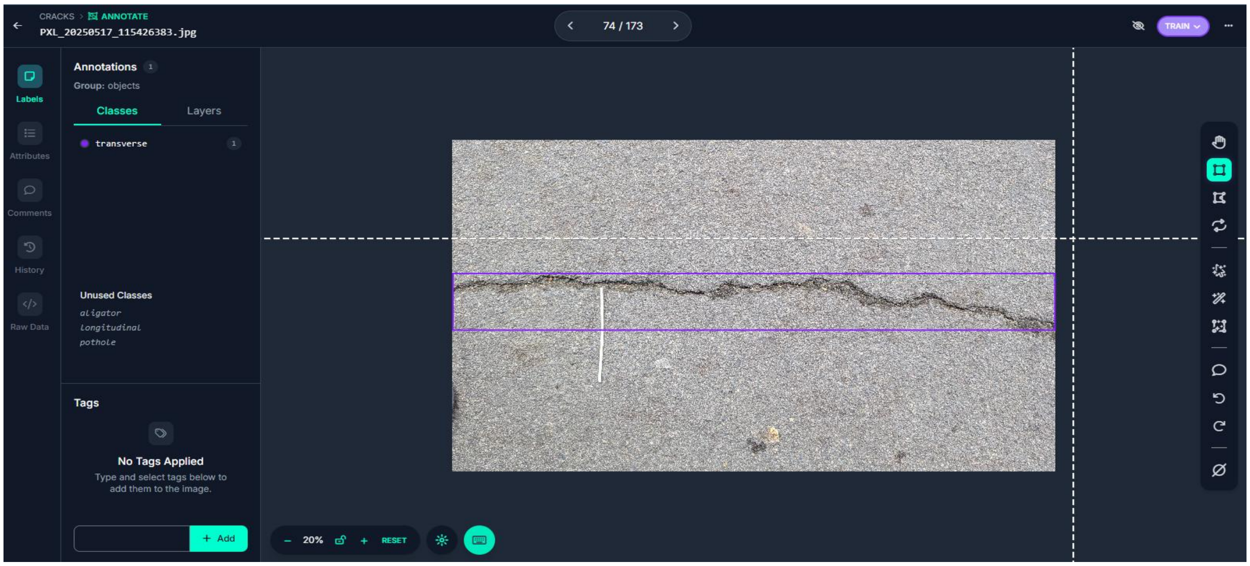
A special dataset of 1,200 images of the pavement surface was made for this investigation. In order to ensure variation in pavement conditions, lighting scenarios, and crack appearances, the photographs were gathered from a variety of road environments in Dhaka, Barisal, Sylhet and Chittagong, four major regions of Bangladesh, guaranteeing a broad range of pavement textures, illumination levels, and levels of damage. A Google Pixel 6 smartphone was used for mage acquisition, making the data collection process accessible and reasonably priced. Clear, high-resolution images appropriate for deep learning applications were guaranteed by the use of a top-notch, commercially available device. This method also mimics actual situations in which local government or civil engineers might utilize such tools for road inspections.

Alligator cracks, which are characterized by interconnected cracks that form a pattern resembling an alligator's skin, longitudinal cracks, which run parallel to the pavement centerline, transverse cracks, which are perpendicular to the pavement centerline, and potholes, which are bowl-shaped depressions in the pavement surface caused by the disintegration of road materials and frequently result in hazardous driving conditions, are the four main categories of pavement damage that are frequently encountered in road infrastructure maintenance. These kinds of damage were chosen because they are common and play a crucial part in determining the integrity of the pavement and scheduling preventative maintenance.

Fig 2. Collected Data Images

3.3 Data Preprocessing

The collected dataset was annotated using Roboflow, ensuring balanced representation across four crack types: longitudinal, transverse, alligator, and pothole cracks. To make object detection tasks easier, each image was carefully labeled. Using the Roboflow platform, each damage instance was manually annotated by drawing bounding boxes around it to show its exact location and type. A student with a solid academic background in civil engineering helped with the classification and annotation process to guarantee accuracy and domain relevance. Their knowledge improved the dataset's quality and dependability by accurately identifying pavement damage in accordance with accepted civil engineering standards. In order to train and assess deep learning models intended for automated pavement damage detection, high-quality ground truth data is ensured by this meticulous annotation process.



No Tags Applied

Type and select tags below to add them to the image.

Fig 3. Image Annotation

3.4 Data Split

Three subsets of the dataset were then created: 70% for training, 15% for validation, and 15% for testing. The model was fitted and its internal parameters were adjusted using the training set. In order to adjust hyperparameters and avoid overfitting, the validation set was essential for tracking the model's performance during training. Lastly, the testing set was used to assess the model's capacity for generalization on entirely unobserved data, offering an objective assessment of its performance in the real world.

Set	**Percentage**	**Image Count**
Training	70%	840
Validation	15%	180
Testing	15%	180
Total	100%	1200

Table 1. Data Split Distribution

3.5 Data Augmentation

To address the relatively small dataset size and enhance the model's generalization ability, an extensive augmentation pipeline was applied to the training set, increasing its size to 5,691 images. The augmentation pipeline is given below:

Category	Augmentation Technique	Purpose
Resizing	Resize to 640×640 pixels	Ensure consistent input size for YOLO models
Geometric Augmentations	Horizontal and vertical flips	Simulate mirrored crack orientations
Geometric Augmentations	90-degree rotations	Add rotational variety
Geometric Augmentations	Random rotations (±15°)	Introduce slight directional variations
Geometric Augmentations	Shear (X and Y axes)	Mimic angular distortion and camera tilts
Geometric Augmentations	Elastic deformation	Simulate surface warping and flexibility
Photometric Augmentations	Grayscale variations (None, Grayscale, CLAHE)	Simulate lighting/contrast differences, highlight edge features
Photometric Augmentations	Color jittering (Brightness, Contrast, Saturation, Hue)	Reflect variable daylight and weather conditions
Photometric Augmentations	Shadow overlay	Imitate natural shadows from vehicles, poles, etc.
Photometric Augmentations	Gaussian noise	Mimic real-world sensor or environmental noise

Occlusion Simulation	Cutout (random rectangular masks)	Improve detection under partial occlusion (e.g., dirt, leaves)
Bounding Box Handling	Auto recalibration & validation	Ensure accuracy of YOLO-format bounding boxes after transformations

Table 2. Augmentation Pipeline

3.6 Model Selection

3.6.1 YOLOv9

The YOLO team at Ultralytics created YOLOv9, a next-generation object detection algorithm, in 2024. By integrating sophisticated backbone and neck modules with anchor-free detection, it outperforms earlier YOLO models. Speed, accuracy, and efficiency are YOLOv9's primary goals, which makes it appropriate for real-time detection tasks like pavement crack detection.

3.6.2 YOLOv9 Architecture

YOLOv9 is built on the following main components:

Backbone: YOLOv9 retains a CNN-based backbone similar to YOLOv8 for multi-scale feature extraction but improves upon it by integrating GELAN. GELAN expands on the Efficient Layer Aggregation Network (ELAN) by incorporating multiple computational blocks, such as CSPblocks, Resblocks, and Darkblocks, without adding to computational complexity. This approach ensures efficient feature extraction while preserving key hierarchical features across the network’s layers, maintaining a balance between accuracy and computation.

Neck: The neck in YOLOv9 incorporates advances seen in YOLOv8's PANet but significantly enhances the feature fusion process using PGI. By combining multi-level auxiliary information from PGI, YOLOv9 improves the fusion of features from different layers, effectively addressing the problem of information loss that occurs as data moves through the network. This helps in stabilizing gradient computations, making YOLOv9 particularly adept at detecting objects of varying sizes.

Head: YOLOv9 continues with an anchor-free bounding box prediction method introduced in YOLOv8 but benefits from the reversible functions provided by PGI. The reversible architecture ensures that no crucial data is lost during the forward and backward passes, leading to more reliable predictions with lower computational overhead. This design improves both inference speed and accuracy, making it more efficient for real-time applications.

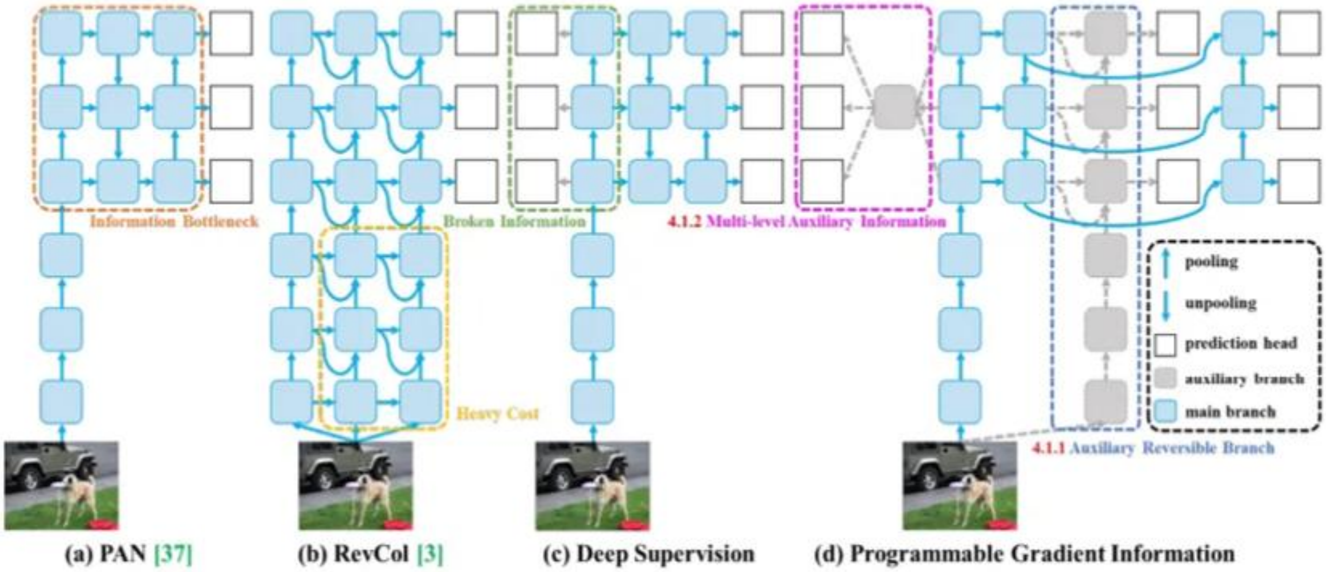


Fig 4. YOLOv9 Architecture [12]

3.7 Model Training Configuration

A well-organized training pipeline was used to effectively train the YOLOv9 model for pavement crack detection. GPU acceleration with CUDA support was used in the training process to speed up computation and boost efficiency. To ensure consistency across inputs, the dataset images were resized uniformly to 640×640 pixels. To maximize training effectiveness and prevent overfitting, a number of hyperparameters were carefully chosen. The main training configurations used in this investigation are compiled in Table 4.

Parameter	**Value**	**Description**
Model	YOLOv9-m	Medium-sized model variant of YOLOv9
Device	cuda:0	GPU-enabled training using CUDA
Image Size	640 × 640 pixels	All input images resized for consistency
Batch Size	16	Balance between GPU memory usage and training stability 640 × 640 pixels
Workers	0	Ensures data loading compatibility across systems
Epochs	80	Maximum number of training iterations
Early Stopping Patience	20	Stops training if no improvement in validation for 20 consecutive epochs

Optimizer	Stochastic Gradient Descent (SGD)	Standard optimizer for YOLO training
Learning Rate (lr)	0.01	Initial learning rate
Learning Rate Final Ratio (lrf)	0.01	Final learning rate relative to initial
Weight Decay	0.0005	Regularization to prevent overfitting
Momentum	0.937	Optimizer used unless specified otherwise

Table 3. Training Parameters

3.8 Experimental Setup

Every experiment was carried out on a powerful workstation running Windows 10 that had an Intel Xeon Silver 4216 processor, an NVIDIA RTX A4000 GPU (16 GB VRAM), and 32 GB of system RAM. To guarantee reproducibility and consistent handling of dependencies, the models were created and trained using Python 3.10.8 in a managed Anaconda virtual environment. PyTorch, a deep learning framework, was utilized, with cuDNN and CUDA Toolkit 11.8 accelerating it for effective GPU-based computation. The Roboflow platform, which offered a variety of image transformations, was used for preprocessing and data augmentation. A wide range of libraries were used in the training process, such as matplotlib for visualization, opencv-python for image processing, torch, torchvision, and ultralytics for model development, and albumentations for sophisticated data augmentation. Additional tools like pycocotools, pandas, pyyaml, and tqdm were also used for dataset handling and evaluation in order to facilitate annotations, configuration management, and progress tracking.

3.9 Evaluation Metric

Several common evaluation metrics were used to gauge how well the object detection models in this study performed. These metrics, which are frequently used to assess detection accuracy in object detection tasks like YOLO-based models, include Precision, Recall, F1-Score, and Mean Average Precision (mAP).

- **Precision:**

  Precision measures the proportion of correctly predicted crack instances out of all instances predicted by the model. It helps evaluate the model's ability to avoid false positives.

  Precision = (True Positives + False Positives) / True Positives

- **Recall:**

  Recall measures the proportion of actual crack instances that were correctly identified by the model, indicating its ability to minimize false negatives.

  Recall = (True Positives + False Negatives) / True Positives

- **F1-Score:**

  F1-Score is the harmonic mean of Precision and Recall. It provides a single score that balances both concerns, especially useful when class distribution is imbalanced.

  F1-Score = (2 × Precision + Recall) / (Precision × Recall)

- **Mean Average Precision (mAP):**

  Mean Average Precision (mAP) is a comprehensive metric widely used in object detection. mAP@0.5 measures the average precision when the IoU (Intersection over Union) threshold is set at 0.5, while mAP@0.5:0.95 computes the average over multiple IoU thresholds from 0.5 to 0.95 in steps of 0.05.

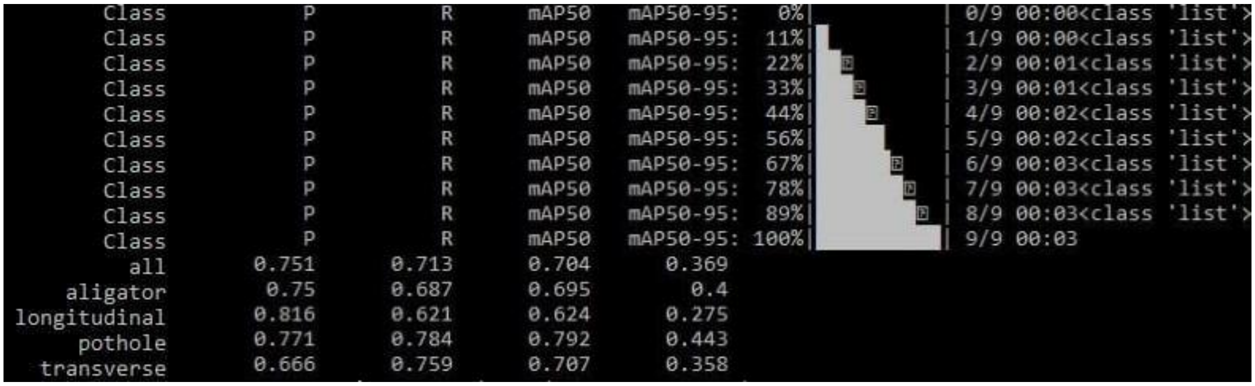
Section 4

Results & Discussions

The experimental results of training the suggested YOLOv9 model for pavement crack detection are shown in this section. Several performance metrics, such as mAP, precision, recall, and F1-score, are used to assess the outcomes. A comparison with other cutting-edge models is also offered. The analysis is supported by a performance comparison table and graphical representations**.**

4.1 Model Performance Evaluation

Four pavement crack classes—alligator, longitudinal, pothole, and transverse—were used to train and assess the YOLOv9 model. To give a thorough picture of the detection performance across all classes, the model's results are shown in terms of Precision (P), Recall (R), mAP@0.5, and mAP@0.5:0.95.



```
       Class          P          R      mAP50  mAP50-95:   0%|          | 0/9 00:00<class 'list'>
       Class          P          R      mAP50  mAP50-95:  11%|█         | 1/9 00:00<class 'list'>
       Class          P          R      mAP50  mAP50-95:  22%|██        | 2/9 00:01<class 'list'>
       Class          P          R      mAP50  mAP50-95:  33%|███       | 3/9 00:01<class 'list'>
       Class          P          R      mAP50  mAP50-95:  44%|████      | 4/9 00:02<class 'list'>
       Class          P          R      mAP50  mAP50-95:  56%|█████     | 5/9 00:02<class 'list'>
       Class          P          R      mAP50  mAP50-95:  67%|██████    | 6/9 00:03<class 'list'>
       Class          P          R      mAP50  mAP50-95:  78%|███████   | 7/9 00:03<class 'list'>
       Class          P          R      mAP50  mAP50-95:  89%|████████  | 8/9 00:03<class 'list'>
       Class          P          R      mAP50  mAP50-95: 100%|██████████| 9/9 00:03
         all      0.751      0.713      0.704      0.369
    aligator       0.75      0.687      0.695        0.4
longitudinal      0.816      0.621      0.624      0.275
     pothole      0.771      0.784      0.792      0.443
  transverse      0.666      0.759      0.707      0.358
```

Fig 5. Test Set Results

The model demonstrated strong object localization capabilities with an overall mAP@0.5 of 70.4% and a mAP@0.5:0.95 of 36.9%. All classes had average precision (P) and recall (R) of 0.751 and 0.713, respectively.

Class	**Precision**	**Recall**	**mAP@0.5**	**mAP@0.5:0.95**
Alligator	0.750	0.687	0.695	0.400
Longitudinal	0.816	0.621	0.624	0.275
Pothole	0.771	0.784	0.792	0.443
Transverse	0.666	0.759	0.707	0.358
All	0.751	0.713	0.704	0.369

Table 4. Class-wise detection performance of YOLOv9

A maximum of 80 epochs were allowed for the training process to run. However, based on the specified patience value, early stopping was initiated at epoch 77 because validation performance did not significantly improve for 20 consecutive epochs. It is noteworthy that the model reached its optimal generalization performance well before finishing all 80 epochs, as evidenced by the fact that the best-performing model weights were saved at epoch 65. This result illustrates how well the early stopping mechanism works to avoid overfitting and preserve computational power.

4.2 Detection Output

The YOLOv9 model's detection results on validation images are shown in Figure 6, which also shows how well it performs across various pavement crack classes.

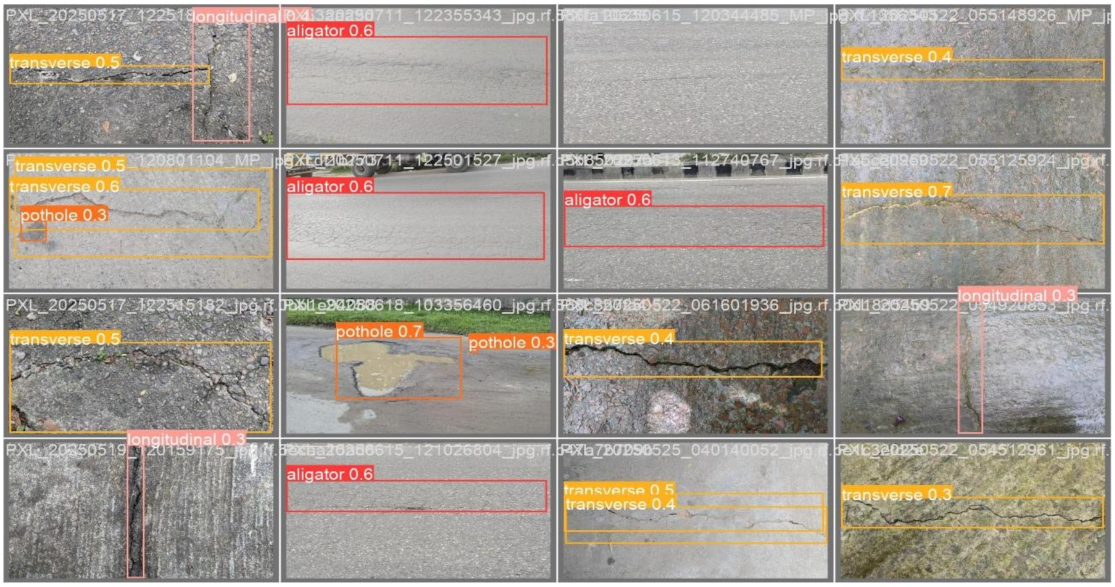


Fig 6. Validation images

All four target classes—alligator, longitudinal, pothole, and transverse—were successfully identified by the model with respectable confidence levels (between 0.3 and 0.7). The model's strong localization performance on more noticeable and well-defined crack types was confirmed by the notable detection of potholes and alligator cracks with higher confidence values (e.g., 0.6–0.7). Even with noisy or worn-out surfaces, the transverse cracks were reliably found in multiple samples, demonstrating the model's resilience to changing edge clarity.

The thinner structure and resemblance to other linear features in the background may be the reason for some detections' comparatively lower confidence scores (e.g., 0.3), particularly for longitudinal cracks. Furthermore, overlapping classes (such as transverse and pothole) were identified in a few instances, demonstrating the model's ability to identify multi-class features in a single frame.

Figure 7 illustrates the YOLOv9 model's detection results on test images, highlighting its performance across various pavement crack classes.

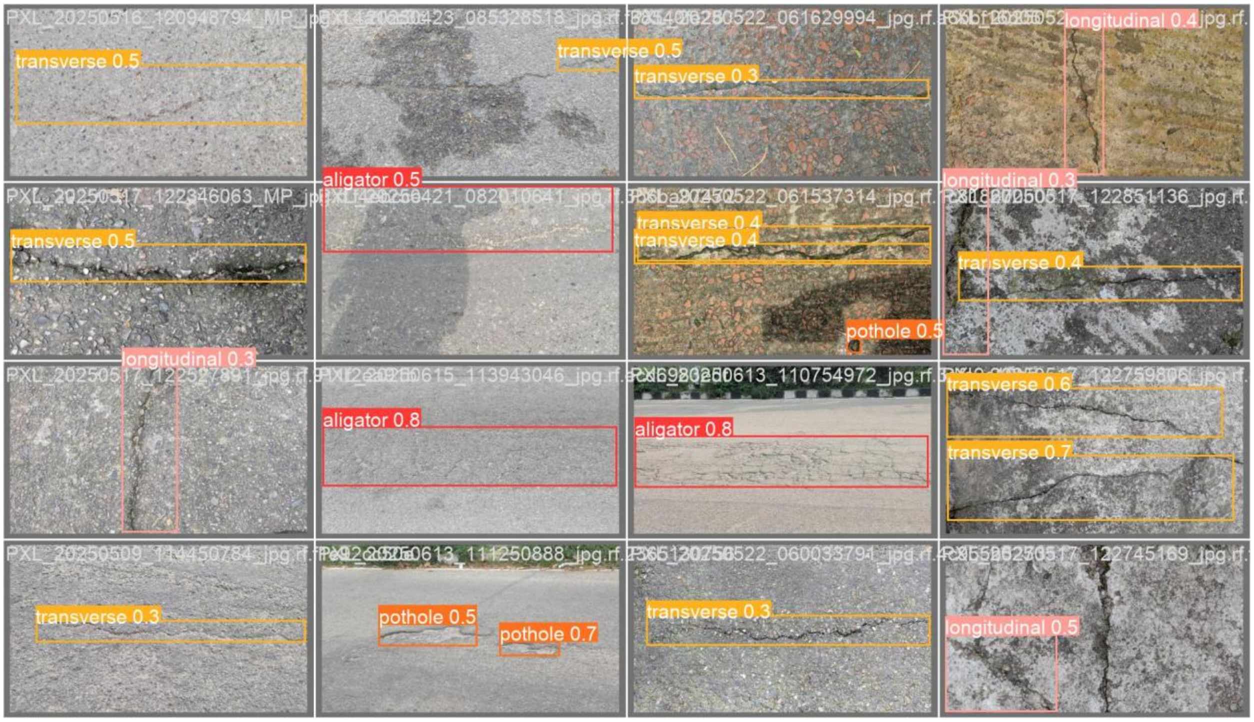


Fig 7. Test images

With multiple bounding boxes scoring between 0.5 and 0.8, the model demonstrates a high degree of certainty in detecting alligator and pothole cracks. Even when overlapping or present in noisy backgrounds, the transverse cracks were consistently identified with a reasonable degree of confidence, despite being more elongated and frequently less distinct. The narrow structure and visual resemblance to surface grooves or shadows of longitudinal cracks, on the other hand, may have contributed to their relatively lower confidence scores (0.3–0.5).

To further evaluate the classification accuracy of the YOLOv9-m model on the test dataset, a normalized confusion matrix was generated. This matrix provides insights into how well the model distinguishes between the four pavement crack classes and the background.

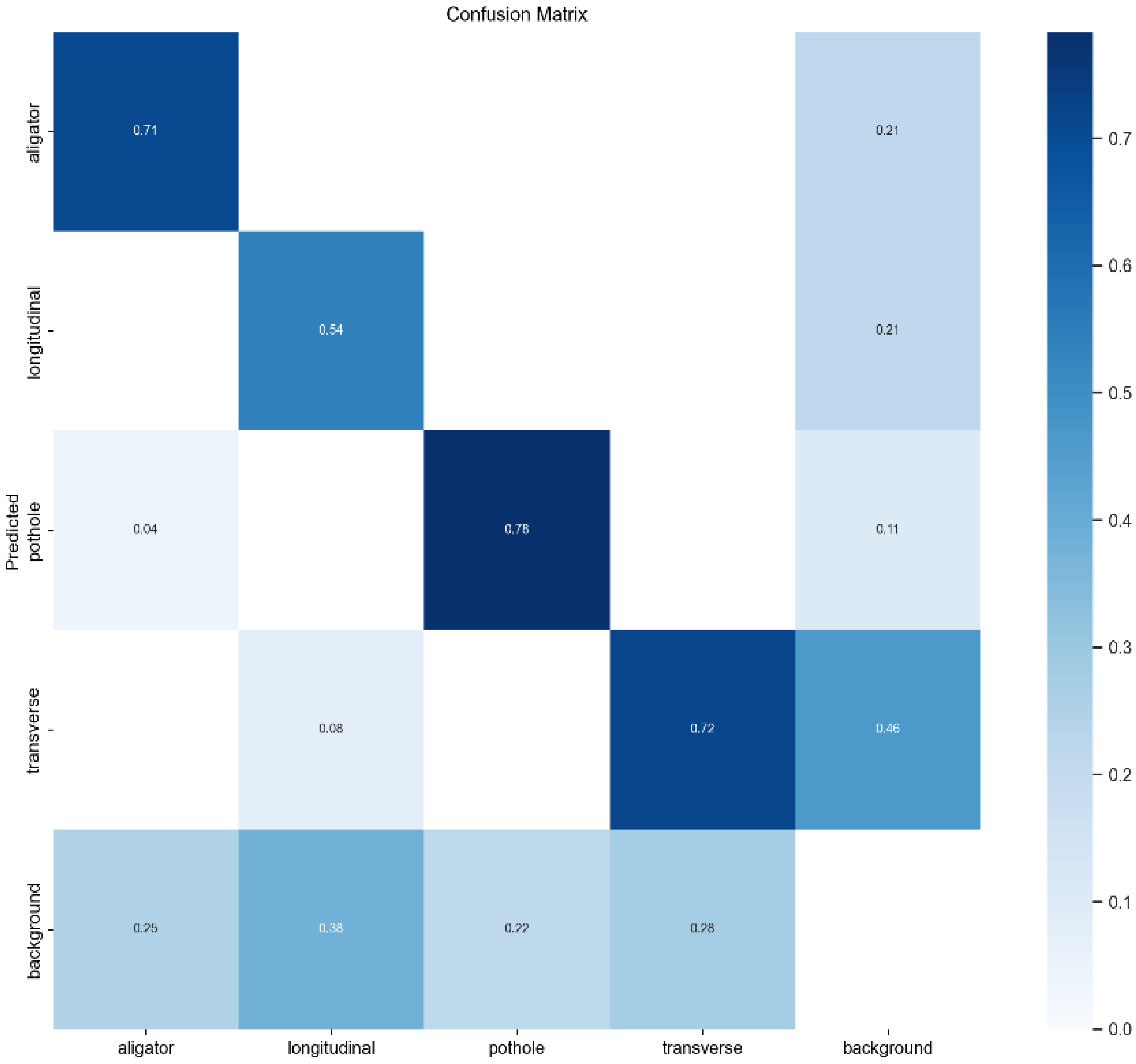


Fig 8. Confusion Matrix

Potholes had the best class-wise accuracy (0.78), closely followed by transverse cracks (0.72) and alligator cracks (0.71). With a true positive rate of just 0.54 and a clear overlap with background and transverse categories, longitudinal cracks were less accurately classified.

A large percentage of all classes, particularly longitudinal and transverse cracks, were mistakenly classified as background in the background column, which is where misclassifications are most noticeable. This implies that under actual test conditions, finer, less noticeable cracks are frequently mistaken for non-crack areas.

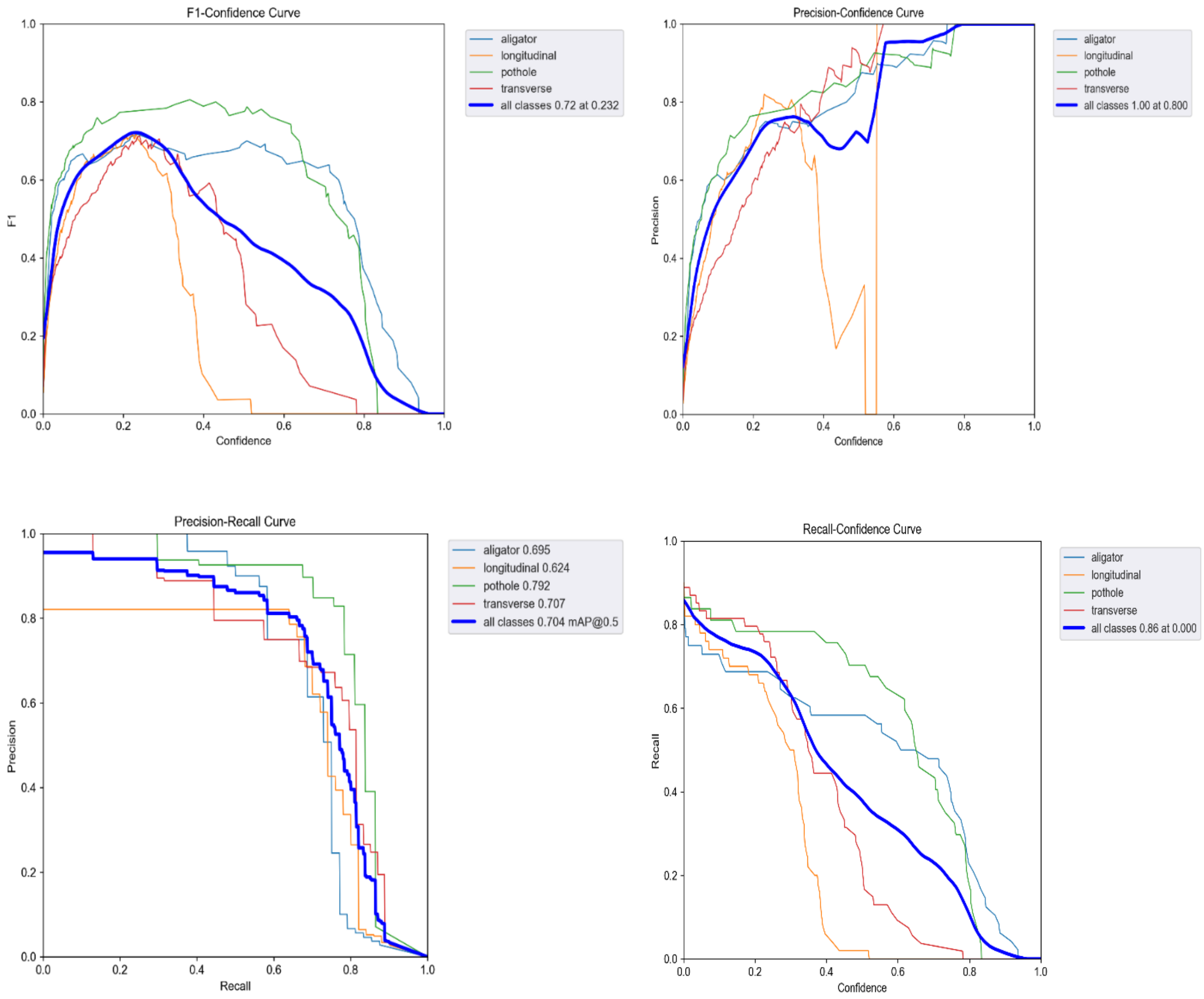


Fig 9. F1-Confidence, Precision-Confidence, Precision-Recall, Recall-Confidence Curves

In the F1-Confidence curve, the model’s ability to balance precision and recall is evaluated. The transverse and pothole classes exhibit higher F1-scores over a broader confidence range, peaking around 0.3 to 0.6. The highest overall F1-score is recorded at 0.72 when the confidence threshold is set at 0.232. This indicates that the model maintains an optimal balance between identifying true positives and minimizing false positives at moderate confidence levels. In contrast, the alligator and longitudinal classes show a more rapid decline in F1-score as confidence increases, indicating less robust performance on those crack types.

The Precision-Confidence curve reveals that the model achieves perfect precision (1.00) at a confidence level of 0.800. This means that when the model is highly confident in its predictions, it is extremely accurate. Among the classes, pothole and transverse cracks consistently show high precision across the entire range, while alligator and longitudinal cracks have more fluctuations, possibly due to visual similarities or labeling inconsistencies in the dataset.

In the Precision-Recall curve, a crucial metric for object detection, the model achieves an overall mAP@0.5 of 0.704. The pothole class again outperforms others with a precision of 0.792, followed closely by transverse (0.707), while alligator (0.695) and longitudinal (0.624) trails slightly. This suggests that the model is particularly well-suited for detecting potholes, likely due to their distinct visual patterns compared to the other crack types.

The Recall-Confidence curve illustrates how recall diminishes with increasing confidence thresholds. All classes achieve their highest recall (0.68) at a confidence threshold of 0.0. As the confidence increases, the model becomes more conservative, prioritizing precision over recall. Among the classes, transverse and pothole cracks sustain relatively higher recall values over a broader range, indicating that these crack types are more reliably detected across different confidence settings.

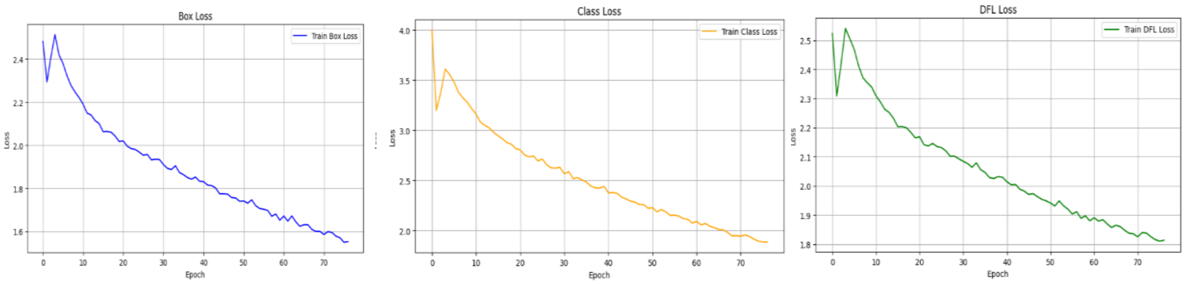


Fig 10. Training Loss Curves

1. **Box Loss (Left)**: The Box Loss graph shows the reduction in error related to bounding box regression, i.e., how accurately the model predicts the location and size of objects. Initially, the box loss starts at around 2.4 and displays minor fluctuations in the early epochs due to rapid learning and weight adjustments. However, it steadily decreases as training progresses and stabilizes around 1.6 by the final epoch. This consistent downward trend indicates that the model is becoming increasingly accurate at localizing cracks in the images.
2. **Class Loss (Middle)**: This graph reflects the error in classifying detected objects into the correct categories. The class loss starts at approximately 3.6 and shows a steep decline during the initial training phase. As the model continues learning, the class loss continues to decrease more gradually, reaching around 2.0 by the 75th epoch. This trend demonstrates effective learning of class features and improved accuracy in distinguishing between different crack types.
3. **DFL Loss (Right)**: The DFL (Distribution Focal Loss) curve evaluates the model’s confidence and precision in localizing bounding boxes with more granular accuracy. The loss begins near 2.5, spikes briefly around the third epoch, and then consistently decreases throughout the training period, ending close to 1.8. The steady downward

pattern signifies that the model is gaining confidence in its predictions and refining the spatial precision of detected objects.

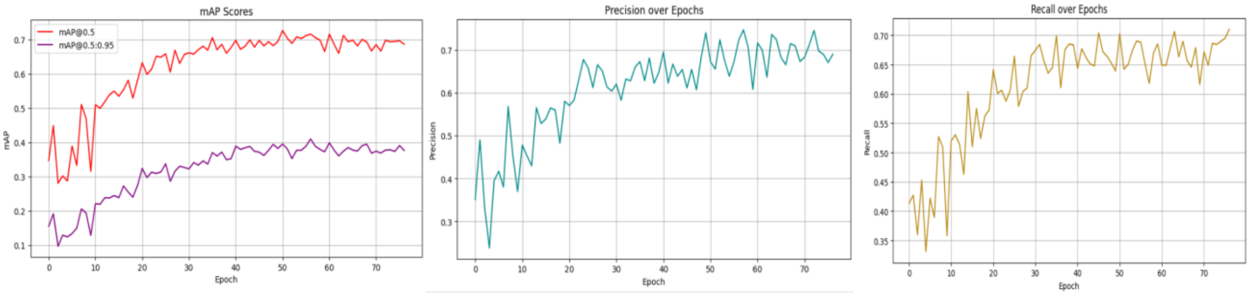


Fig 11. mAP, Precision, Recall Curves (Validation)

1. **mAP Scores**: This graph displays the mean Average Precision (mAP) at two IoU thresholds such as mAP@0.5 (red line) and mAP@0.5:0.95 (purple line). The mAP@0.5 starts low around 0.3 but rises steadily, stabilizing above 0.70 after around 35 epochs. This indicates that the model becomes very confident and accurate in predicting object locations with at least 50% IoU match. The mAP@0.5:0.95, a stricter metric that averages mAP across multiple IoU thresholds, also increases gradually, reaching a peak of around 0.45, which shows decent localization precision under stricter evaluation.
2. **Precision over Epochs**: Precision represents how many of the predicted bounding boxes are correct. Initially fluctuating due to unstable learning in early epochs, it quickly improves and oscillates between 0.65 and 0.75 in later stages of training. This reflects the model's increasing ability to make accurate predictions without many false positives.
3. **Recall over Epochs**: Recall measures how many actual objects are correctly detected. The recall improves gradually from below 0.4 and stabilizes around 0.68 to 0.70 by the final epochs. This suggests that the model becomes increasingly capable of detecting most relevant instances with fewer misses.

4.3 Discussion

It is clear from the above results that the YOLOv9-m model works well for all types of cracks. Potholes, probably because of their unique shape and characteristics, have the highest mAP@0.5 (79.2%) and mAP@0.5:0.95 (44.3%). The model showed good generalization in detecting potholes even though it had the fewest instances, perhaps thanks to the variety of augmentations and image clarity.

The model's high level of confidence in its predictions is demonstrated by the longitudinal cracks, which had the highest precision (81.6%). Its comparatively low recall (62.1%), however, suggests a propensity to overlook some true positives, perhaps as a result of the subtle and limited appearance of these cracks.

With a slightly lower precision (66.6%), which suggests some false positives, transverse cracks showed the highest recall (75.9%), suggesting that the model was able to detect the majority of instances of this class. Despite having a block-like structure that makes them highly distinguishable, the alligator cracks only received moderate scores on all metrics, striking a balance between recall (68.7%) and precision (75%).

Overall, the model's superior performance in identifying longitudinal cracks and potholes demonstrated the benefit of having distinct shapes or high contrast features. Transverse and alligator cracks' comparatively poorer performance raises the possibility that they need more class-specific augmentation or improved dataset representation.

4.4 Model Comparison

The comparative outcomes of the two models are compiled in Table 5 to give a cohesive performance overview.

Model Name	**Precision**	**Recall**	**F1-Score**	**mAP@0.5**
Faster R-CNN	0.692	0.652	0.6724	0.641
YOLOv11	0.685	0.655	0.684	0.667
YOLOv12	0.712	0.695	0.704	0.699
YOLOv9	**0.751**	**0.713**	**0.732**	**0.704**

Table 5. Different model comparison

YOLOv9 outperforms all other models in every important metric, including precision, recall, F1-score, and mAP@0.5, according to the performance evaluation of four distinct models: Faster R-CNN, YOLOv11, YOLOv12, and YOLOv9. The best F1-score (0.732) was obtained by YOLOv9, which had the highest precision (0.751) and recall (0.713), demonstrating a good balance between identifying true positives and reducing false positives. It also achieved the highest mean Average Precision (mAP@0.5) of 0.704, with YOLOv12 coming in second at 0.699. Although YOLOv12 outperformed YOLOv11 and Faster R-CNN, YOLOv9 is unquestionably the best model in this comparative study for concrete crack detection tasks due to its higher scores on all evaluation metrics.

Section 5

Conclusion & Future Work

5.1 Conclusion

Using specially gathered data from actual road conditions, this study effectively created a deep learning-based pavement crack detection system. The YOLOv9, YOLOv11, YOLOv12, and Faster R-CNN models were trained and evaluated using the dataset, which included 1,200 photos classified into four crack categories (potholes, longitudinal, transverse, and alligator). YOLOv9 outperformed the others in terms of F1-score and mean Average Precision (mAP), confirming the efficacy of the suggested strategy.

The model demonstrated resilience in identifying different kinds of cracks in a range of surface, texture, and lighting conditions. Additionally, the system was feasible for widespread implementation due to the utilization of low-cost image capture (through smartphones) and model training on powerful hardware. The model's generalizability was improved while avoiding overfitting thanks to early stopping during training, which showed stable convergence and the best performance at epoch 65 out of a maximum of 80.

By providing an automated, scalable, and precise crack classification system, this work seeks to support policymakers, road maintenance authorities, and civil engineers in making timely repair decisions and managing infrastructure effectively.

5.2 Future Work

While the results are promising, several directions can be explored to enhance this research:

- **Dataset Expansion**: The model's generalizability would be enhanced by expanding the dataset size and including more varied environmental conditions (such as rainy, nighttime, or rural roads).
- **Real-Time Deployment**: Real-time crack localization and detection may be made possible by incorporating the model into drone or mobile applications.
- **Post-Processing & Severity Analysis**: Deeper understanding of maintenance prioritization may be possible by combining detection with the use of crack severity estimation or segmentation models.
- **Cross-Country Generalization**: The model's performance across various pavement types and standards can be assessed by testing it on crack images from other nations.
- **Model Optimization**: The model could be deployed on edge devices with constrained resources by using additional quantization or compression techniques.
- **Comparison with Other SOTA Models**: To further validate the effectiveness of the architecture, Transformer-based detectors (such as DINO and DETR) could be used for benchmarking against YOLO.

References

1. Dong, Xuwei, Yang Liu, and Jinpeng Dai. "Concrete surface crack detection algorithm based on improved YOLOv8." Sensors 24.16 (2024): 5252
2. Pan, Y., et al. "Road pavement condition mapping and assessment using remote sensing data based on MESMA." IOP Conference Series: Earth and Environmental Science. Vol. 34. No. 1. IOP Publishing, 2016.
3. Ya’acob, Norsuzila, et al. "Real-time pavement crack detection based on artificial intelligence." J. Adv. Res. Appl. Sci. Eng. Technol 38.2 (2024): 71-82.
4. Muslim, Nurul Hidayah, Mohamad Ibrahim Mohamed, Zulkarnaini Mat Amin, Arezou Shafaghat, Mohammad Ismail, and Ali Keyvanfar. "Pavement Structural Assessment Using Automated Tools: A Comparative Study." Malaysian Journal of Civil Engineering 29, no. 1 (2017): 129-152.
5. Hassan, Syed Ali, Seung Heon Han, and Soo Young Shin. "Real-time road cracks detection based on improved deep convolutional neural network." 2020 IEEE Canadian conference on electrical and computer engineering (CCECE). IEEE, 2020.
6. AAOSHAT (2008), Bridging the Gap: Restoring and Rebuilding the Nation’s Bridges, American Association of State Highway and Transportation Officials, Washington D
7. Adeli, H. & Jiang, X. (2008), Intelligent Infrastructure: Neural Networks, Wavelets, and Chaos Theory for Intelligent Transportation Systems and Smart Structures, CRC Press, Taylor & Francis, Boca Raton, Florida
8. Sami, Abdullah As, et al. "Improved YOLOv5-based real-time road pavement damage detection in road infrastructure management." Algorithms 16.9 (2023): 452.
9. Sohaib, Muhammad, Muzamal Arif, and Jong-Myon Kim. "Evaluating YOLO models for efficient crack detection in concrete structures using transfer learning." Buildings 14.12 (2024): 3928.

10. Yu, Zhen. "YOLO V5s-based deep learning approach for concrete cracks detection." SHS web of conferences. Vol. 144. EDP Sciences, 2022.
11. Arya, D., et al. "RDD2022-The multi-national Road Damage Dataset released through CRDDC'2022." (No Title) (2022).
12. Wang, Chien-Yao, I-Hau Yeh, and Hong-Yuan Mark Liao. "Yolov9: Learning what you want to learn using programmable gradient information." European conference on computer vision. Cham: Springer Nature Switzerland, 2024.