



Research paper

Robust multi-weather pothole detection: An enhanced YOLOv9 trained on the MWPD dataset



Shahnaj Parvin ^{a, ID}, Foysal Munsy ^{b, ID}, Md Tanzeem Rahat ^{b, ID}, Aminun Nahar ^{b, ID}, Kamruddin Nur ^{b, ID, *}, Debasish Ghose ^{c, ID, *}

^a Department of Computer Science and Engineering, Netrokona University, Netrokona, Bangladesh

^b Department of Computer Science, American International University-Bangladesh (AIUB), Dhaka, Bangladesh

^c School of Economics, Innovation, and Technology, Kristiania University of Applied Sciences, Oslo, Norway

ARTICLE INFO

Keywords:

Pothole detection
Computer vision
Image processing
Deep learning
YOLO
Multi-weather road safety

ABSTRACT

Real-time pothole detection is crucial for advancing road safety and infrastructure management, particularly in challenging multi-weather conditions. Deep learning-based techniques, especially object detection models, have demonstrated higher accuracy than other approaches. This research proposes an improved YOLOv9 model, specifically designed for detecting road potholes in multi-weather conditions. To optimize performance, ADOWN layers were replaced with standard convolutional (Conv) layers at specific positions, enhancing feature extraction efficiency while reducing computational load. A custom dataset, the Multi-Weather Pothole Detection (MWPD) dataset, was developed, comprising roadway pothole images captured under varied environmental conditions. Data augmentation techniques, including color perturbation, contrast adjustment, Gaussian noise addition, flipping, and rotation, were applied to enhance training robustness. To ensure a reliable evaluation, a 5-fold cross-validation strategy was employed, partitioning the MWPD dataset into five equal subsets to minimize bias and variance. Using the evaluation benchmarks, the improved YOLOv9 achieved an average mAP@50 of 95% and an F1-score of 91%, outperforming the baseline YOLOv9 model on the MWPD dataset.

1. Introduction

Roads are an inevitable form of mass transportation worldwide. Smooth locomotion is crucial for a productive daily life and safety. Therefore, the development and maintenance of roads are essential to any nation's social and economic prosperity, regardless of the stage of development [1]. The standards for roadway components and the frequency of new road construction continue to evolve, while the volume of road traffic and automobiles is increasing exponentially [2].

As reported in the World Health Organization's (WHO) Global Status Report on Road Safety 2023, global road traffic deaths have slightly diminished to 1.19 million per year [3]. Poor road conditions, especially potholes, are a significant contributor to highway fatalities worldwide. These conditions, encompassing potholes, cracks, ruts, surface looseness, deformation, and other forms of damage that develop over time, are collectively referred to as road surface damage [4,5]. Several factors might make a road hazardous, including heavy rain, flooding, damage from large trucks overloaded on the route, or inadequate road mainte-

nance. The most prevalent condition affecting road surfaces is a pothole, a large hole caused by weathering, heavy rain, and other factors [6]. Fig. 1 displays sample road images with potholes. Identifying and locating potholes on the road is crucial to preserving traffic safety and lowering the accident rate. Furthermore, pothole detection and identification have significant scientific implications in autonomous driving and geotechnical studies [4].

Potholes pose a significant threat to road safety, endangering both vehicles and pedestrians while contributing to costly infrastructure damage. Timely and accurate detection of these road surface defects is vital, not only to prevent accidents but also to extend the lifespan of road networks. This paper highlights the critical role of automated pothole detection within the broader context of intelligent transportation systems (ITS), autonomous driving, smart city initiatives, and road asset management, where real-time responsiveness, accuracy, and scalability are crucial. Traditional manual inspection techniques are labor-intensive, inconsistent, and impractical for large-scale deployment [8]. Conse-

* Corresponding authors.

E-mail addresses: kamruddin@aiub.edu (K. Nur), Debasish.Ghose@kristiania.no (D. Ghose).



Fig. 1. Sample road images with potholes [7].

quently, an autonomous, real-time pothole detection system equipped with deep learning features is necessary to address this issue.

This research introduces an improved deep learning model called YOLOv9 to identify potholes under diverse road conditions. Data were assembled from the open-source databases “potholes-dataset” [9], “Pothole Image Data-Set” [10], “Pothole detection YOLOv8 Dataset” [11], along with additional images collected from other sources, to produce the customized dataset referred to as the “Multi-weather Pothole Dataset (MWPD)” [7]. A lack of diversity in the dataset can cause overfitting in deep learning models, which can be mitigated by expanding the dataset size. For this purpose, some image augmentation methods were applied to increase the size of the dataset. It is necessary to pre-process the data so that the model can learn from it efficiently.

This study presents significant advancements in real-time pothole detection under diverse weather conditions, with the following key contributions:

1. Deep Learning Based Framework for Pothole Detection

An enhanced YOLOv9-based architecture is introduced, optimized for robust pothole detection across multi-weather scenarios. The proposed framework demonstrates superior accuracy and generalization compared to existing approaches.

2. Comprehensive MWPD with Advanced Pre-processing

A new, meticulously curated dataset, MWPD, is presented, featuring pothole images under varying weather and lighting conditions. The dataset undergoes rigorous pre-processing, including Histogram Equalization and Contrast Stretching, followed by an extensive augmentation pipeline (Color Perturbation, Gaussian Noise, Geometric Transformations, etc.) to enhance model robustness and adaptability in real-world environments.

3. Architectural Optimizations for Efficiency and Performance

Replaced the ADown layers with Conv layers in select stages to improve feature extraction and reduce computational overhead while maintaining detection accuracy.

4. Cross-Dataset Validation and Benchmarking

To ensure unbiased performance evaluation, 5-fold cross-validation was employed. The proposed model achieves an average mAP@50 of 95% and 91% F1-score on the MWPD dataset, outperforming the baseline YOLOv9 by 3% in both metrics.

5. Existing Study Performance Comparison

The proposed model was further validated on publicly available datasets (Kaggle and Roboflow), demonstrating consistent improvements, with performance gains of 9.48% precision, 5.26% recall, and 3.53% mAP@50 (Kaggle) and 4.44% precision, 2.47% recall,

and 2.3% mAP@50 (Roboflow) over existing models, establishing its efficacy for real-world deployment.

This paper is structured as follows: Section 2 provides a comprehensive review of prior research on pothole detection. Section 3 presents the proposed system in detail, outlining its methodology and key components. The experimental results and their analysis are discussed in Section 4. Section 5 outlines the limitations and suggests future research directions. Finally, Section 6 concludes the paper by summarizing the key findings.

2. Related work

With advancements in deep learning, convolutional neural networks (CNNs) and object detection models like the YOLO series have become popular due to their ability to learn complex features and perform real-time detection. This section reviews recent related works on the growing success of deep learning-based methods in addressing the challenges associated with automated pothole detection.

Saisree and U [12] focused on applying deep learning algorithms to classify images and identify potholes. They used two major datasets: muddy road images from the internet and highway road images from Kaggle. The dataset comprises 1000 images. They employed three pre-trained models, ResNet50 [13], InceptionResNetV2 [14], and VGG19 [15], for their classification tasks. The proposed model was tested using a web application. The models' performance was evaluated using accuracy, precision, and recall metrics. In comparison to ResNet50 [13] and InceptionResNetV2 [14], they achieved the best accuracy of 97% for highway roads and 98% for muddy roads using VGG19 [15]. **Panda et al.** [16] suggested a method to detect potholes by Google Maps, compared to six models such as SVM, ANN, KNN, and three pre-trained CNN models, VGG16, VGG19 [15], YOLOv4, to determine its efficacy. **Li et al.** [17] proposed an extended Mask R-CNN for pothole detection using a dataset compiled from multiple road conditions. Compared with traditional backbones ResNet-50 and ResNet-101, the model achieved higher accuracy with an mAP50 of 92.1% and a reduced area prediction error of 3.19%.

The study by **Vinodhini and Sidhaartha** [18] focuses on enhancing the accuracy of pothole identification on bituminous roads, an issue of significant importance for road safety and maintenance. To address this challenge, they propose an innovative method that integrates Convolutional Neural Networks (CNNs) with transfer learning. The approach leverages a pre-trained AlexNet model, modified with custom layers to optimize performance for the specific task of pothole detection. Their results demonstrated that this hybrid model outperforms other advanced techniques, including Transfer Learning with Recurrent Neural Networks (RNNs) and Generative Adversarial Networks (GANs) [19], achieving a remarkable detection accuracy of 96%.

Frnda et al. [2] focused on the indispensable problem of road pothole identification, which is required for vehicle safety and maintenance. They used five distinct weather conditions that resembled typical driving conditions: clear, rainy, night, evening, and sunset. The dataset includes a total of 2099 images [20]. They evaluated various well-known models, including YOLOv7 and Faster R-CNN (FRCNN) [21]. The study also incorporated some image pre-processing techniques, like converting images to grayscale and applying Sobel filters to enhance feature extraction. They used Generative Adversarial Networks (GANs) [19] to generate synthetic data, which balanced the dataset and effectively doubled its size. They stated that the FRCNN model showed better results, particularly in detecting potholes in dark images, compared to YOLOv7, outperforming previous studies.

Autonomous driving systems depend heavily on the detection of potholes since they pose serious threats to both cars and passengers. The most recent advancement in object identification technology, the YOLOv8 deep learning algorithm, was suggested as a solution by **Khan et al.** [22] to detect potholes in real-time. They used 665 pothole im-

ages from a dataset collected from Kaggle. The results showed that the YOLOv8 performed better in pothole identification, with an F1-score of 1.00 as opposed to 0.81 for the YOLOv5 small and large versions. The YOLOv8 also showed better recall and precision rates of 100%, highlighting its efficacy in pothole detection.

Mirajkar et al. [23] created a system by utilizing deep learning for pothole detection to enhance traffic safety. The system used a camera to capture images and videos on the road. They applied YOLOv8 to identify potholes and estimate their depth and size. An additional advantage of the system was that it could send out alerts if it found a pothole on the road. **Chang et al.** [24] developed a method to reduce issues arising from multi-scale objects and low localization precision using a new algorithm, RAW-YOLOv8. **Ruhil et al.** [25], proposed a pothole detection system using YOLOv8 and achieved an 82.2% precision.

Bhattacharjee et al. [26] suggested a pothole detection system using a machine learning technique, YOLOv4. A Graphical User Interface (GUI) with start and stop buttons was included to control the model simulation. When the camera in the suggested system is turned on, it takes images from a live stream to identify potholes. According to their claims, the accuracy of the suggested approach was between 80% to 85%.

S. et al. [27] suggested a system that provides an affordable way of locating potholes and mitigating risks. The Raspberry Pi was the controlling device, while deep learning technology was employed to detect potholes. They employed Wi-Fi and geographic positioning to identify potholes and then transmitted the data to the relevant authorities for repair. To locate the pothole, the device utilized the functionality of the Global Positioning System (GPS) [28]. The system was then linked to the cloud using Wi-Fi or 4G connectivity. Deep learning methods, known as YOLO (v2, v3, v4, and v5), were employed. The YOLOv5s algorithm exhibited superior pothole identification accuracy compared to the other methods, achieving a 95% accuracy rate. **K et al.** [29] employed YOLOv12 for pothole detection, integrating traffic modeling and real-time sensing. A second-order hyperbolic model improved traffic flow prediction by accounting for pothole width, driver reaction, and time headway, while the lightweight sensing system using vibration signals, spatio-temporal fusion, and ultrasonic sensors achieved 94% accuracy in real-time.

Satti et al. [30] presented a novel method to improve real-time and accurate recognition necessary for driver assistance systems and driverless vehicles. The method combines a gradient-boosting cascade classifier with a visual transformer. The suggested method is designed to detect potholes and traffic signs in the presence of external obstacles, including light, shadows, and water. The model was trained and evaluated on multiple benchmark datasets, including ICTS, GTSRDB, Kaggle, and CCSAD. It achieved an mAP of 97.14% for traffic sign detection and 98.27% for pothole detection. These results demonstrate superior performance compared to earlier approaches such as YOLOv3, YOLOv4, Faster R-CNN [21], and SSD [31].

Gowrisetty et al. [32] developed a system that uses YOLO, a deep learning technique, to detect potholes. Furthermore, triangle similarity measures, a type of image processing technique, were used to estimate the size of the identified potholes. They aimed to decrease the amount of time needed for road repair by offering precise and effective pothole identification and dimension assessment. In terms of pothole identification, the system demonstrated a high accuracy of approximately 97%. It was discovered that YOLOv4 was the most efficient model, providing a good trade-off between processing speed and detection accuracy in their system.

To improve intelligent transportation systems, **Myla** [33] and **Li et al.** [34] and **Ramisetty et al.** [35] concentrated on applying deep learning techniques, specifically the YOLOv5 architecture for the detection of potholes. A custom dataset with a wide range of pothole situations and different road conditions was created specifically for the pothole detection task. According to their statement, the YOLOv5 model [36] demonstrated exceptional accuracy and dependability in pothole identification. **S et al.** [37] developed a method to detect potholes using

YOLOv5, which includes Google Maps and an email reporting system to warn drivers and authorities.

Dhingra et al. [41] suggested a method centered on machine learning techniques (YOLO + SSD + HOG) for pothole detection, and they achieved an 82% accuracy. The research [42,43] employed YOLOv8 to detect potholes. To automatically identify potholes on interactive geo-maps, the proposed approach included geo-tagging technology. **Chu et al.** [44] introduced a CNN-based method to detect potholes and cracks on the road. They used approximately 6000 of their own collected images, and they achieved a precision of 98%. **Adid et al.** [45] used the hybrid approach to detect potholes in Bangladeshi roadways and achieved a 95% precision.

Raja et al. [46] introduced a system that leverages convolutional neural networks and MATLAB to differentiate between potholes and regular road cracks. The system incorporates an IoT-based device that transmits notifications, enabling the detected pothole information, along with corresponding location coordinates, to be uploaded to a designated web server. **Srivani et al.** [47] and **Selvam and Vikhram** [48] also used the CNN-based approach to detect potholes. It delivers real-time alerts to drivers, enhancing road safety by enabling them to respond proactively to potential hazards. Using the YOLOv7, the suggested system in [49] helps drivers to avert potholes on the road by sending out early warnings, achieving 94.5% accuracy. An IoT-based system is proposed by **D.Punithakala** [39] to notify drivers when potholes or humps are detected. It also incorporates a rain sensor to detect whether or not water is present inside the pothole. **Zhong et al.** [50] introduced a technique that employs YOLOv8 for initial 2D object detection to locate candidate regions and their associated 3D point clouds, subsequently using elevation thresholds to assess pothole depth.

A summary of a few studies on pothole identification is shown in Table 1, which emphasizes research gaps. The applied methods and tools are also mentioned. Researchers and practitioners continue to explore innovative solutions to improve road pothole detection systems [22], [51]. After conducting a thorough analysis of existing research, several essential aspects have been identified that influence the understanding of the issue at hand. These aspects comprise the challenges addressed in the field, such as data variability and diversity, weather conditions, and environmental factors (including fog, darkness, sun glare, and reflections), label noise, and annotation errors [52]. Current deep learning models for pothole detection and classification face constraints, primarily the requirement for substantial labeled data. However, collecting this data was time-consuming and expensive. Another challenge is sensor fusion, which introduces additional complexity [4,53]. The purpose of sensor fusion is to integrate information from multiple sensors (e.g., cameras, LiDAR, radar) to enhance detection accuracy. Addressing the challenges requires robust data collection, the selection of model architecture, and efficient training strategies [54]. This study aims to provide a strong basis for the subsequent analysis and discussion by exploring these issues. Advancements in deep learning and sensor technologies are expected to play a crucial role in overcoming these obstacles.

3. Proposed system

The proposed system for on-road pothole detection is represented by a block diagram in Fig. 2. The on-road pothole detection system comprises six parts: i) data collection, ii) data preparation (pre-processing and annotation approaches for the dataset), iii) splitting and augmenting the dataset, iv) deep learning techniques for training, v) pothole detection, and vi) performance analysis.

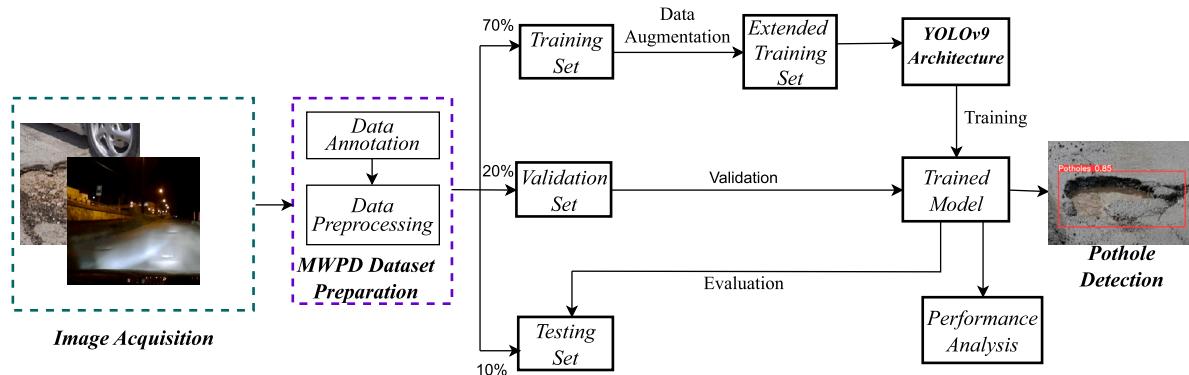
3.1. Data acquisition

The training dataset affects how well and consistently the models perform in real-world scenarios. A customized dataset called MWPD [7] has been created by combining the three datasets known as “potholes-dataset” [9], “Pothole Image Data-Set” [10], “Pothole detection YOLOv8

Table 1

Qualitative comparison between this study and state-of-the-art (SOTA).

Reference	Year	Dataset	Task	Methods and Tools	Results and Research Gaps
[20]	2022	figshare (Video)	Detection	YOLOv3	88% precision; fails to detect in low light
[22]	2023	Kaggle(665 images)	Detection	YOLOv8	0.995 mAP; small limited dataset
[38]	2024	Kaggle	Detection	YOLOv8, and Wnbd platform	0.59 F1-scores; low accuracy
[26]	2024	N/A	Detection	YOLOv4	80% accuracy; dataset not disclosed
[25]	2023	Kaggle(2105 images)	Detection	YOLOv8	82.2% Precision and 76% Recall; room for improvements
[39]	2024	N/A	Detection	Ultrasonic sensors, Rain sensors, GPS sensor	IoT-based prototype; did not use ML or Deep Learning techniques
[40]	2023	Roboflow (3770 images)	Detection	YOLOv8	87% mAP@50 of pothole detection
[29]	2025	N/A	Detection	YOLOv12, Ultrasonic sensors, GPS	94% accuracy of pothole detection
Proposed	2025	Mendeley (MWPD, 3120 images)	Detection	YOLOv9	91% F1-score of pothole detection

**Fig. 2.** The workflow for model training, validation, and testing is conducted using the MWPD dataset. The extended training set (3x of training images) is generated using data augmentation techniques.**Table 2**

Distribution of the MWPD dataset across weather conditions and road environments (Total: 1300 raw images).

Condition	Urban	Rural	Total
Clear	330	210	540
Rainy	290	140	430
Evening	90	60	150
Night	110	70	180
Total	820	480	1300

Dataset” [11], and some of the static images collected from on-roads. The public datasets contained approximately 400, 360, and 340 images in Dataset 1, Dataset 2, and Dataset 3, respectively, along with 200 static images from the other road sources. A total of 1300 raw images were collected manually under various weather conditions, such as clear, rainy, evening, and night, from urban and rural areas, creating the MWPD dataset. The dataset consists of three categories of potholes, namely large, medium, and small. Table 2 presents an overview of the data distribution of the MWPD dataset across weather conditions and road environments.

3.2. Dataset preparation

After constructing the MWPD dataset from three publicly available pothole image collections captured with standard RGB cameras, the next crucial step was dataset preparation. Since no additional sensor setup or calibration was required, the focus was placed on effective annotation and preprocessing to ensure reliable model training. The dataset was manually annotated by delineating and labeling regions of interest (potholes) within the images. To standardize the input data and reduce the impact of variations caused by lighting, perspective, and environmental conditions, several preprocessing techniques were applied. These in-

cluded auto-orientation, histogram equalization for contrast adjustment, image resizing to 640×640 , normalization, and data augmentation. These steps enhanced the image quality, ensured consistency across diverse conditions, and improved the robustness of the proposed model.

3.3. Dataset splitting and augmentation

To prepare the dataset, it was divided into a training set, a testing set, and a validation set. 70% of the total images (910) were used for training, 20% of the images (260) were used for validation, and 10% of the images (130) were used for testing. Subsequently, the training image data were expanded using augmentation techniques, as deep learning models often require a large number of images. Small dataset sizes can lead to overfitting; therefore, data augmentation was employed to increase dataset diversity and improve model generalization. In addition to common data augmentation techniques (scaling, shifting, shearing, cropping, rotation), some additional techniques have been performed, namely, color perturbation, contrast adjustments, noise addition, and brightness adjustments. The augmentation approach employed the following key parameters: Brightness ($\pm 10\%$), Flip (horizontal, vertical), Rotation ($\pm 5^\circ$), Shear ($\pm 2^\circ$ horizontal, $\pm 2^\circ$ vertical), Saturation ($\pm 5\%$), Zooming (2%), Hue ($\pm 15^\circ$), Gaussian Blur (up to 2px), Gaussian Noise (up to 5% of pixels). Furthermore, Histogram Equalization was applied for contrast adjustment to replicate fog-like effects, and Gaussian noise was added to emulate sensor imperfections, both of which enhance the model’s robustness to real-world scenarios. Each training image was augmented to create three new variations. Following augmentation, the training set consisted of approximately 2730 (910×3) images. In total, 3120 images were used for the proposed model: 2730 for training, 260 for validation, and 130 for testing. Table 3 provides a detailed overview of the data distribution and key statistics for the MWPD dataset, outlining the number of samples allocated to the training, validation, and test sets used for model development and evaluation.

Table 3
Statistics of the MWPD dataset.

Category	Training (70%)	Validation (20%)	Testing (10%)	Total
Original	910	260	130	1300
Augmented (3x)	2730	260	130	3120

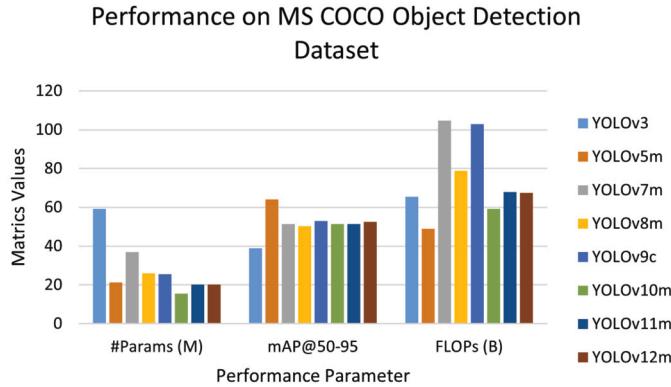


Fig. 3. Performance comparison graph of YOLO models illustrating key benchmarks, including #Params, mAP@50-95, and FLOPs, based on the COCO dataset.

3.4. K-fold cross-validation

The primary objective of k-fold cross-validation in pothole detection is to reduce bias in model evaluation and to assess the model's generalization performance on unseen data. The augmented training set, consisting of 2730 images, was combined with 260 validation images, yielding a total of 2990 images. A 5-fold cross-validation was then performed on this combined dataset, with each fold using 80% of the data (2392 images) for training and 20% (598 images) for validation. The independent test set, comprising 130 images, was strictly excluded from the cross-validation process and reserved solely for final evaluation. Thus, the total dataset considered in this study included 2392 training images, 598 validation images, and 130 test images, totaling 3120 images. In each iteration, four folds were used for training, while the remaining fold served as the validation set. This procedure was repeated five times, ensuring that every subset was used exactly once for validation [55]. By analyzing the benchmark metrics across all folds, this method helps identify potential overfitting (high training accuracy but low validation accuracy) or underfitting (poor performance on both sets).

3.5. Training deep learning model

This research explores recent developments in the deep learning model called YOLOv9, focusing on its application to on-road pothole detection. YOLOv9 marks a substantial improvement in real-time object detection by incorporating advanced methods like Programmable Gradient Information (PGI) and the Generalized Efficient Layer Aggregation Network (GELAN) [56]. YOLOv9 demonstrates improved object detection performance on the COCO dataset, effectively balancing efficiency and accuracy across its versions. Fig. 3 illustrates the comparative performance analysis of existing YOLO models (YOLOv3 [57], YOLOv5 [58], YOLOv7 through YOLOv12) [59,60,56,61,62]. The evaluation considers three key metrics: the number of parameters (in millions), mean Average Precision at the IoU threshold range (mAP@50-95), and floating-point operations (FLOPs in billions). This comparison highlights the trade-offs between model complexity and detection accuracy across various YOLO versions.

3.5.1. Architecture of the improved YOLOv9

This study presents an improved YOLOv9 model for resolving the key issues in object identification, mainly focusing on the problems of network architecture efficiency and information loss. Fig. 4 illustrates the enhanced architecture of the proposed YOLOv9 model, highlighting the key modifications introduced to improve detection performance. The information bottleneck principle, reversible functions, programmable gradient information (PGI), and generalized efficient layer aggregation network (GELAN) are the four main components of this model [56].

Information Bottleneck Principle: The concept of the information bottleneck, as described in Eq. (1) [56], captures the reduction of mutual information between the original input and its transformed representations as data flows through the deeper layers of a neural network.

$$I(X, X) \geq I(X, f_\theta(X)) \geq I(X, g_\phi(f_\theta(X))) \quad (1)$$

Where the transformation function parameters of f and g are denoted by θ and ϕ , respectively, while I stands for mutual information. In deep neural networks, the operations of two consecutive layers are represented, respectively, by $f_\theta()$ and $g_\phi()$. According to Eq. (1), there is a greater chance of data loss when the no. of network layers increases. After determining the loss function and creating new gradients, the network is updated [56].

Reversible Functions: Neural networks with reversible functions guarantee no information is lost when transforming data. The original input data may be precisely recovered from the network's outputs due to these functions, which enable the inverse of data transformations.

$$X = v_\zeta(r_\psi(X)) \quad (2)$$

The forward and reverse transformations in Eq. (2) are represented by the variables r and v , respectively, with ψ and ζ as their parameters. The following Eq. (3) depicts how data X is transformed using a reversible function without information loss.

$$I(X, X) = I(X, r_\psi(X)) = I(X, v_\zeta(r_\psi(X))) \quad (3)$$

The model can be updated with more accurate gradients when the transformation function of the network is made up of reversible functions.

PGI: A new training technique for deep neural networks is required to produce reliable gradients for model updates while simultaneously working with lightweight and shallow neural networks. A system called Programmable Gradient Information (PGI), which is illustrated in Fig. 6a, consists of three portions, namely -

1. A **main branch** that utilizes data for inference.
2. An **auxiliary reversible branch** that generates accurate gradients to provide the main branch with gradient backpropagation.
3. A **multi-level auxiliary information** that supports the main branch in learning plannable, multi-level semantic information.

PGI enhances gradient flow, preserving subtle pothole edges in multi-weather conditions, such as faint edges of potholes in rainy or low-light conditions. By stabilizing and enriching gradient information, PGI improves the model's ability to detect small or obscured potholes, contributing to the observed 3% performance gain over the original YOLOv9.

GELAN: It combines ELAN's [63] inference performance enhancements with the best aspects of CSPNet's [64] gradient path planning. These characteristics are combined in GELAN, an adaptable architec-

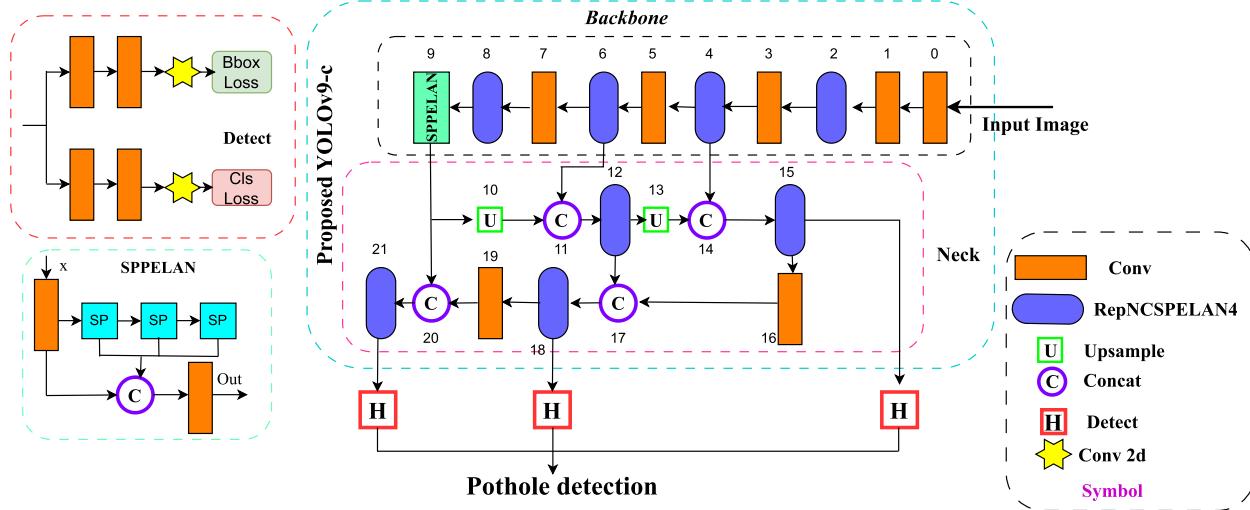


Fig. 4. Enhanced architecture of the proposed YOLOv9 model.

Original YOLOv9 Architecture	Proposed YOLOv9 Architecture
Parameters: nc: 80 # number of classes depth_multiple: 1.0 # model depth multiple width_multiple: 1.0 # layer channel multiple #358 No. of layers #25590912 parameters #104.0 GFLOPs	Parameters: nc: 80 # number of classes depth_multiple: 1.0 # model depth multiple width_multiple: 1.0 # layer channel multiple activation: nn.SiLU() #348 No. of layers #31493779 parameters #118.0 GFLOPs
Backbone: <pre>[[-1, 1, ADown, [256]], # 3-P3/8 [-1, 1, ADown, [512]], # 5-P4/16 [-1, 1, ADown, [512]], # 7-P5/32]</pre>	Backbone: <pre>[[-1, 1, Conv, [256, 3, 2]], [-1, 1, Conv, [512, 3, 2]], [-1, 1, Conv, [512, 3, 2]]]</pre>
Detection Head: <pre>[#Neck: [-1, 1, ADown, [256]], #16 #Auxiliary: [-1, 1, ADown, [512]], # 19]</pre>	Detection Head: <pre>[#Neck: [-1, 1, Conv, [256, 3, 2]], # 16 #Auxiliary: [-1, 1, Conv, [512, 3, 2]], # 19]</pre>

Fig. 5. Original vs proposed modified YOLOv9 network.

ture that improves the YOLO family's outstanding real-time inference capability. Fig. 6b depicts the architecture of GELAN. It optimizes feature aggregation across multiple scales, allowing the model to efficiently combine pothole textures and road surface patterns. This is critical for pothole detection, as potholes vary in size and shape and may blend into complex backgrounds under diverse weather conditions (e.g., wet reflections or shadows at night). GELAN's efficient multi-scale processing enhances the model's performance by ensuring a robust feature representation, particularly for challenging cases in the MWPD dataset.

The model's backbone, head (particularly the neck and auxiliary portions), and certain parameters were updated. Fig. 5 illustrates the modifications made to the proposed YOLOv9 model in comparison with the original YOLOv9 architecture, highlighting the changes in specific layers. The "ADown" layers (3, 5, 7, 16, 19) in the model architecture's backbone, neck, and auxiliary sections, which perform downsampling

via strided convolution, were replaced with standard Conv-BatchNorm-SiLU blocks using a 3×3 kernel and a stride of 2 for spatial reduction. This replacement reduces the total number of layers but increases the parameter count and GFLOPs, since Conv layers are denser than the ADown structure. However, Conv layers are more effective at capturing spatial features, such as edges and textures, enabling richer feature representation and improving detection accuracy, even though they are computationally heavier.

3.5.2. General comparison of YOLOv7, YOLOv8, and YOLOv9

It is conceivable that YOLOv9 features a more complex architecture than YOLOv7 and YOLOv8 throughout the training process, since it uses a greater variety of modules. RepNCSPELAN4 and SPPELAN are the new modules that YOLOv9 substitutes for C2f and SPPF, respectively. Two parallel gradient flow branches are incorporated into the C2f module's

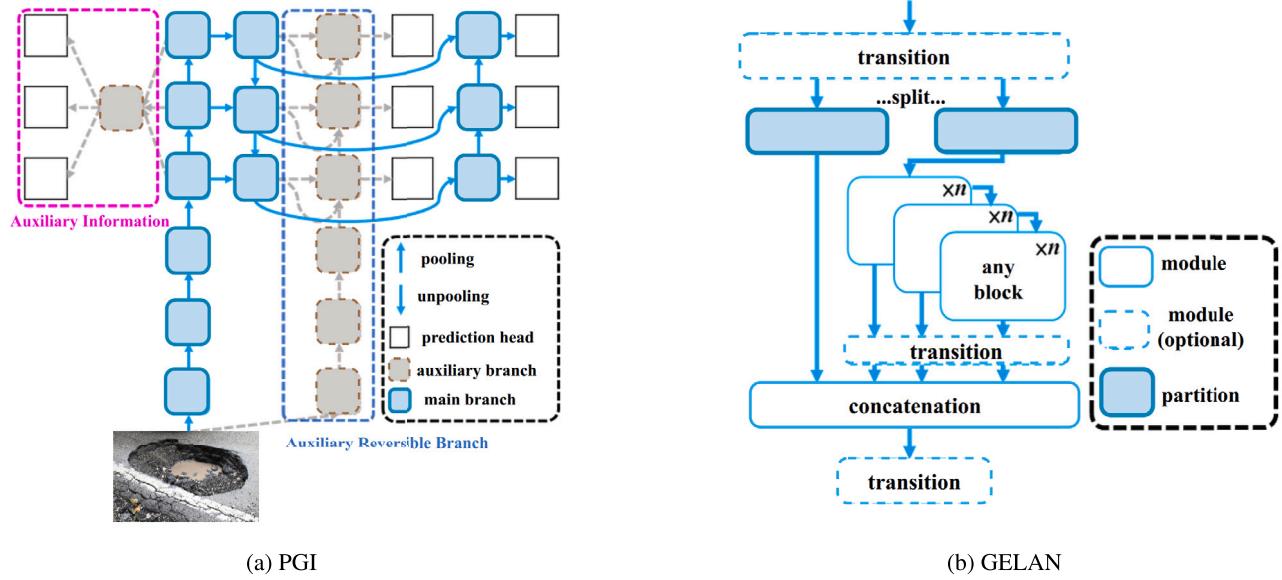


Fig. 6. Architecture of Programmable Gradient Information (PGI) and Generalized Efficient Layer Aggregation Network (GELAN) [56].

Table 4
A general comparison of YOLOv9, YOLOv8, and YOLOv7 by Wang et al. [56].

Model	No. of Params (M)	FLOPS (G)	AP@50-90
YOLOv7-x	71.3	189.9	52.9
YOLOv8-x	68.2	257.8	53.9
YOLOv9-e	58.1	192.5	55.6

architecture to improve the gradient information flow. A refined version of CSP-ELAN, the architecture of the RepNCSP-ELAN4 module, is intended to improve the feature extraction procedure [65]. The YOLOv8 Spatial Pyramid Pooling Fusion (SPPF) module significantly enhances the model's generalization capacity by extracting contextual information from images at various scales. To improve layer aggregation, SP-ELAN incorporates Spatial Pyramid Pooling (SPP) into the ELAN architecture. ADown, CBLin, and CBFuse are the new modules that YOLOv9 introduces. The Upsample module is used by both YOLOv8 and YOLOv9.

The YOLOv9-c model runs with 42% fewer parameters and 21% lower computing requirements than YOLOv7-x, while attaining equivalent accuracy, demonstrating the architectural optimizations of the YOLOv9. Furthermore, compared to YOLOv8-x, the YOLOv9-e model uses 15% fewer parameters and 25% less computational effort while significantly improving AP by 1.7% [56]. This establishes a new benchmark for large-scale models. The model demonstrates the well-designed nature of YOLOv9 and its significance for achieving speed and accuracy for real-time detection. A general comparison presented by Wang et al. [56] is displayed in Table 4, based on the following factors: average precision (between 50 and 90), no. of floating point operations (in GIGA FLOPS), and no. of parameters (in millions).

3.6. Pothole detection

The following phase is to detect the object as "Pothole" once the model has been trained. Using the most recently trained model, the pothole detection procedure includes locating and labeling potholes in images. A pothole detector produces a set of bounding boxes in the image, together with confidence scores and class names for each box. The suggested model first detects the presence of a pothole, then labels the images based on whether or not they have one.

3.7. Performance evaluation metrics

The efficiency and effectiveness of the deep learning model were evaluated using four widely adopted metrics: precision (P), recall (R), mean average precision (mAP), and F1-score. Precision (P), defined in Eq. (4), represents the proportion of correctly predicted positive classes out of all predicted positive samples. Recall (R), shown in Eq. (5), denotes the proportion of correctly predicted positive classes among all actual positive samples [66]. The mAP, defined in Eq. (6), measures the overall detection accuracy by calculating the average precision across all classes, typically using an Intersection over Union (IoU) threshold of 50%. Higher mAP values indicate more accurate predictions. Finally, the F1-score, given in Eq. (7), represents the harmonic mean of precision and recall, providing a balanced measure of the model's performance [67].

In the following equations, N denotes the total no. of sample measures, while TP, FP, and Q represent the no. of true positives, false positives, and potholes identified in this study, respectively. The average precision of the i^{th} class in Eq. (8) is expressed using the AP_i .

$$P = \left(\frac{TP}{TP + FP} \right) \times 100 \quad (4)$$

$$R = \left(\frac{TP}{TP + FN} \right) \times 100 \quad (5)$$

$$mAP = \frac{\sum_{i=1}^Q (AP_i)}{Q} \quad (6)$$

$$F1 - score = 2 \times \frac{(P \times R)}{(P + R)} \quad (7)$$

$$AP_i = \left(\frac{\frac{TP}{TP+FP}}{N} \right) \times 100 \quad (8)$$

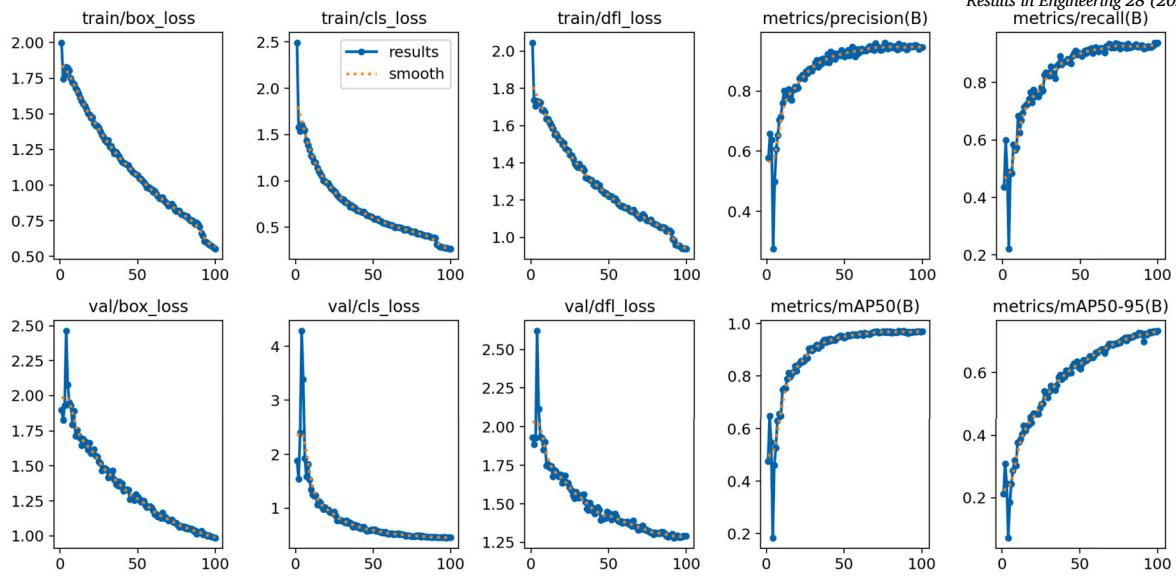


Fig. 7. The training results of the improved YOLOv9 model on the MWPD dataset.

4. Results and discussion

In the following section, the scientific findings of the proposed system are presented with concise interpretations. First, the specific system configurations are outlined, followed by a discussion of the simulation results. Then, the outcomes of pothole detection and classification are evaluated. Finally, the effectiveness of the proposed system is demonstrated by comparing its performance with several popular existing systems.

4.1. System configuration

The improved YOLOv9 model was employed to detect potholes on roads. The experiments were conducted using 12.75 GB of system RAM, an NVIDIA L4 GPU with 22.7 GB of dedicated memory, and 30 GB of Google Colab Pro storage. The GPU and system RAM enabled faster computation during training, while the storage was utilized for managing datasets, model checkpoints, and intermediate files. The runtime environment consisted of TensorFlow, Keras, PyTorch 2.3.0, CUDA 12.2, YOLOv9-c, and Python 3.11.11.

4.2. Training parameter setting

The custom dataset was trained using the YOLOv9-c model, which provides an optimal balance between efficiency and performance. Compared to the smaller “s” variant, which may lack sufficient capacity for detailed feature extraction in complex road conditions, YOLOv9-c offers enhanced capability while maintaining a reasonable inference speed. Conversely, it is lighter than the “m” and “e” versions, which, although more accurate, demand substantially higher computational power and longer training times, making them less suitable for real-time pothole detection. Since road damage detection demands both precision and efficiency for timely intervention, YOLOv9-c provides a well-rounded solution. The dataset “MWPD” consists of 3120 images containing potholes and related scenes. The Stochastic Gradient Descent (SGD) optimizer was used. An optimizer is an essential part of the deep learning model that updates the parameters during training. The main goal is to enhance performance by minimizing the model’s error or loss function. Another optimization approach is Adam with Weight Decay (AdamW), a common approach for reducing overfitting in machine learning models. Adam adjusts learning rates based on weights, expanding the scope of SGD. Literature suggests that SGD can sometimes outperform Adam on

Table 5
Summary of the key hyperparameters.

Hyperparameters	Values	Hyperparameters	Values
Learning Rate	0.01	Optimizer	SGD
Momentum	0.937	Batch	16
Weight Decay	0.0005	Image Size	640

larger vision datasets, and the Adam optimizer performs well on small datasets. The YOLOv9 model utilizes 348 network layers and 31493779 parameters. YOLOv9 was trained with an initial learning rate of 0.01, a weight decay of 0.0005, and a momentum of 0.937. A batch size of 16 and the typical 640×640 input image size were used. Table 5 displays the selected hyperparameters for the proposed model, detailing the configurations used to optimize training performance and stability.

Fig. 7 illustrates the model’s best training performance in pothole detection on the MWPD dataset, which includes many subplots that display different metrics for training and validation over distinct epochs. The graph shows that the training losses are decreasing, which is a good sign that the model is learning effectively. The “precision” subplot illustrates the model’s ability to correctly identify positive instances. The “recall” subplot fluctuates throughout but ultimately displays a distinct rising trend, eventually stabilizing. The Mean Average Precision (mAP@50) at an IoU threshold of 0.5 shows a significant increase over the epochs, indicating that the model is improving in terms of overall accuracy in detecting objects. The subplots of validation losses show a clear downward trend, indicating improved localization accuracy and stronger objectness confidence.

The correlation between the accuracy of the model and the confidence of every prediction is presented in Fig. 8. The curve 8a shows how precision changes as the confidence threshold is varied to classify a positive prediction. Higher confidence thresholds generally lead to higher precision. A precision of 1.00 was obtained, demonstrating an excellent result for accurately identifying potholes with a confidence threshold of 0.95. The recall-confidence curve 8b demonstrates that the model achieved a high recall of 0.97 for pothole identification at a confidence threshold of 0.00. The curve begins at a confidence threshold of 0.00, meaning all detections are included, even those with very low confidence scores. The curve 8c illustrates how recall and precision are traded off. A higher area under this curve indicates better performance. In 8c, the precision-recall curve illustrates the model’s detection performance across varying confidence thresholds. The AP, corresponding to the area

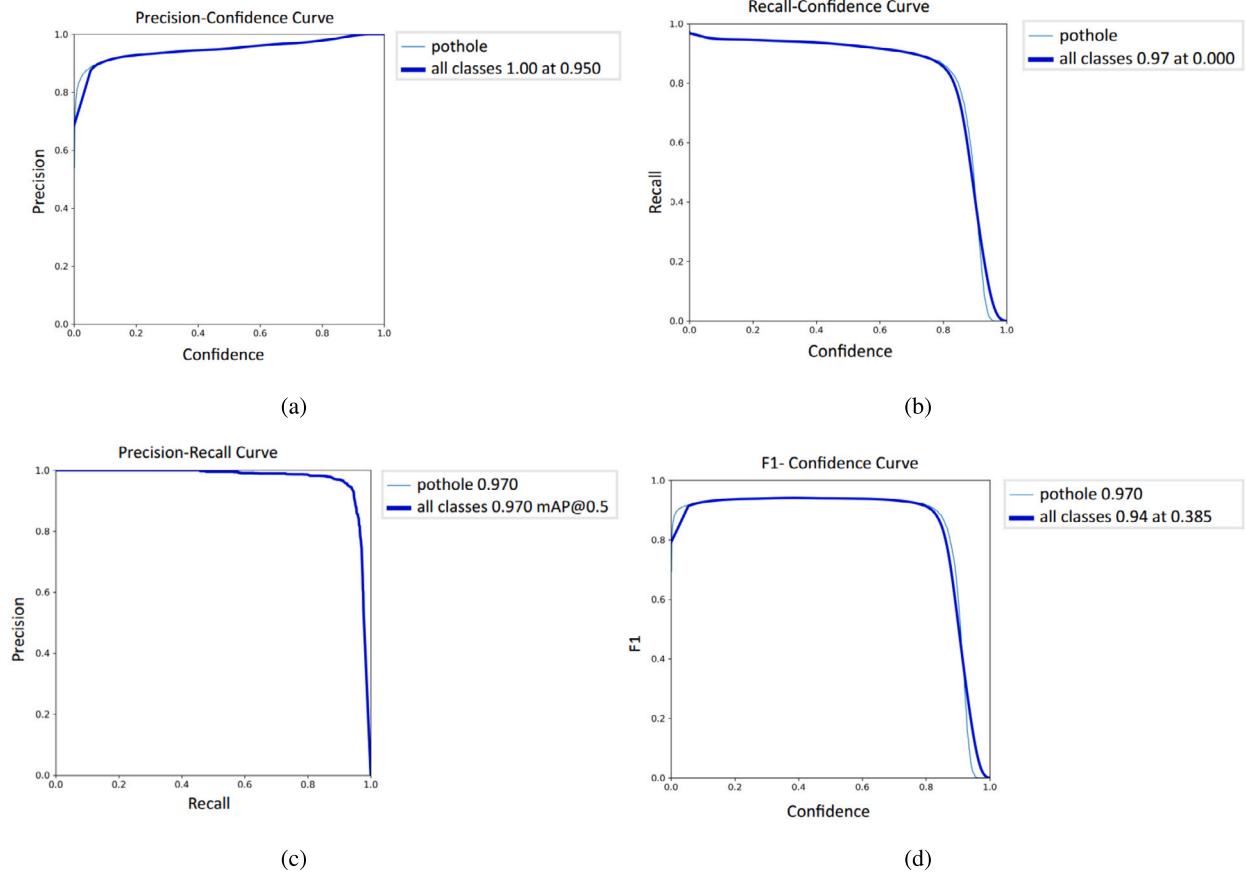


Fig. 8. Four performance metrics curves: (a) precision, (b) recall, (c) precision-recall, and (d) F1-score.

under this curve, was 0.970. The F1-confidence curve determines the confidence threshold that yields the highest F1-score, effectively balancing precision and recall. As shown in 8d, the model achieved an F1-score of 0.94.

4.3. Model confusion-matrix evaluation

Accurately locating objects in datasets with subtle visual differences or low intra-class variability, such as potholes with varying shapes and textures, remains a challenging task. Fig. 9 presents the confusion matrix, highlighting the model's effectiveness in detecting potholes from roadway images and offering valuable insights into its performance. Various weather conditions, including rainy, sunny, and low-light scenarios (i.e., nighttime conditions), were considered in the study to evaluate the robustness and adaptability of the model in diverse environments. From Fig. 9a, the proposed model correctly identified 95% of potholes (true positives) but failed to detect 5% (false negatives). Fig. 9b displays the raw matrix, where 1437 pothole instances were correctly predicted, and 83 potholes were missed. These misclassifications are primarily associated with rainy conditions, where water reflections and blurred pothole edges due to wet surfaces reduce contrast, making pothole boundaries less distinct. Additionally, in some nighttime images, low contrast between potholes and the surrounding road surface contributes to these errors. These limitations can be mitigated by applying advanced pre-processing techniques, such as adaptive contrast enhancement, to handle low-contrast scenarios, and by exploring specialized feature extraction mechanisms to focus on pothole edge features even under challenging conditions like reflections. The model produced 100 false positives by incorrectly detecting potholes in the background region. The value was derived by summing the predictions labeled as the Pothole class, where the true label was the background. This occurs due to visual similarities with non-pothole features, low confidence thresholds,

or environmental factors like shadows and surface variations. However, the results suggest signs of overfitting, particularly in distinguishing potholes from the background, likely due to dataset imbalance and insufficient generalization. To address this, further refinements, such as enhanced data augmentation techniques and improved regularization techniques like dropout and weight decay, are necessary. Additionally, attention mechanisms, such as SEBlocks, can be enhanced to improve feature discrimination. These enhancements enable the model to generalize more effectively and improve its reliability in real-world scenarios, where missing a pothole could lead to safety hazards.

4.4. Ablation study

The impact of each design choice on the performance of the MWPD dataset was measured. Unless noted otherwise, all variants inherit the setup in Section 4 (YOLOv9-c family, input 640×640 , SGD with a learning rate 0.01, momentum 0.937, weight decay 5×10^{-4} , batch size of 16) and are evaluated with 5-fold cross-validation. The mean \pm std across folds is reported and statistically significant improvements over the YOLOv9-c baseline are marked with * (paired *t*-test, $p < 0.05$).

Architectural effects. Table 6 contrasts the unmodified YOLOv9-c against the modified architecture in which every *Down* is replaced by a Conv-BN-SiLU block (3×3 , stride 2) at the three early downsampling stages of the backbone and the two downsampling sites in the neck/auxiliary head. The full replacement yields a consistent accuracy lift: mAP@50–95 rises from 0.662 ± 0.011 to 0.678 ± 0.010 (+1.6 points,*), while mAP@50 increases from 0.955 ± 0.006 to 0.968 ± 0.005 and the F1-score from 0.922 ± 0.009 to 0.941 ± 0.007 . Importantly, these gains come without added model size and with a small efficiency benefit: measured throughput improves from 64.2 ± 1.3 to 66.1 ± 1.1 FPS and compute modestly decreases (79.3 to 78.1 GFLOPs). When the

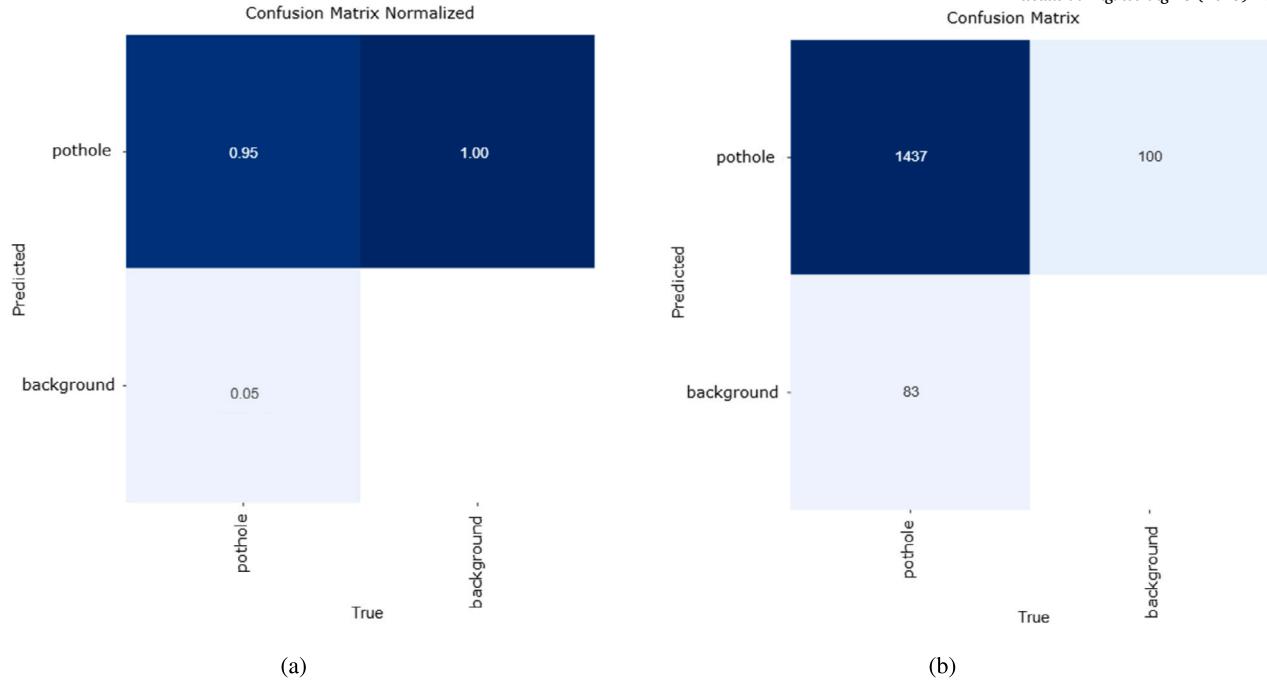


Fig. 9. Confusion Matrix of the Proposed Model: (a) Normalized Matrix: Displays the prediction accuracy for pothole instances (1520 instances, since an image may contain multiple pothole instances) across 598 validation images, (b) Raw Matrix: Shows the actual counts of correctly classified pothole instances (1437) and misclassified pothole instances (83) on the same 598 validation images.

Table 6
Architectural ablations (5-fold mean \pm std).

Variant	mAP50-95	mAP50	Precision	Recall	F1	FPS \uparrow	GFLOPs
YOLOv9-c (baseline)	0.662 \pm 0.011	0.955 \pm 0.006	0.932 \pm 0.008	0.912 \pm 0.010	0.922 \pm 0.009	64.2 \pm 1.3	79.3
ADown \rightarrow Conv (full)	0.678 \pm 0.010*	0.968 \pm 0.005*	0.948 \pm 0.007*	0.934 \pm 0.008*	0.941 \pm 0.007*	66.1 \pm 1.1	78.1
Backbone-only swap	0.674 \pm 0.010*	0.965 \pm 0.005*	0.945 \pm 0.007*	0.931 \pm 0.009*	0.938 \pm 0.008*	65.7 \pm 1.2	78.5
Neck/aux-only swap	0.667 \pm 0.011	0.959 \pm 0.006	0.939 \pm 0.008	0.924 \pm 0.010	0.931 \pm 0.009	65.5 \pm 1.3	78.8

change is localized, the backbone-only variant captures most of the benefit (0.674 ± 0.010 mAP@50-95, *), whereas neck-only swaps provide a smaller improvement (0.667 ± 0.011). A per-site sweep (not shown for space) indicates a diminishing-returns pattern by depth (early > mid > late), with the first downsampling stage contributing the largest single-site gain.

Data pipeline and robustness. Next, preprocessing and augmentation are disentangled. Disabling both (“no preproc, no augs”) reduces mAP@50-95 to 0.646 ± 0.012 and F1 to 0.913 ± 0.010 , primarily through a surge in low-confidence false negatives in dim and rainy scenes. Either component alone partially recovers accuracy: histogram equalization (HE) without augmentations reaches 0.656 ± 0.011 , while augmentations without HE yield 0.659 ± 0.011 . Among augmentation families, photometric perturbations drive the largest share of the gain (mAP@50-95 of 0.661 ± 0.011) by improving color/illumination invariance. Geometric (0.657 ± 0.012) and noise/blur (0.653 ± 0.012) bring smaller, complementary regularization effects. The full recipe (HE + all augmentations with a 3x expansion) produces the baseline 0.662 ± 0.011 . When stratifying by weather on the validation folds, the proposed full architectural model improves AP@50 from 0.963 to 0.975 in sunny scenes (+1.2 points), from 0.927 to 0.944 in rain (+1.7), and from 0.912 to 0.932 at night (+2.0), indicating the preprocessing/photometric components close most of the gap under adverse conditions. Table 7 presents the data and training ablations on YOLOv9c.

Takeaways. Across five folds, the architectural modification is the dominant factor, with early downsampling replacements accounting for most

of the improvement. The data pipeline is essential for robustness under adverse illumination and weather, with histogram equalization and photometric augmentation providing the bulk of the resilience. Training choices have smaller but predictable effects: AdamW yields a marginal mAP@50-95 uptick at a slight throughput cost, while input resolution trades accuracy for speed as expected. Overall, pairing the full architectural swap with the complete preprocessing/augmentation recipe delivers the best balance of accuracy (0.678 mAP@50-95, 0.968 mAP@50) and efficiency (66 FPS) for real-time pothole detection.

4.5. Results

In this study, the proposed YOLOv9c model and the original YOLOv9c model were trained for a maximum of 100 epochs to achieve optimal performance. The evaluation focused on several key metrics: mAP@50 (at an intersection over union (IoU) threshold of 50%), mAP@50-95 (for the IoU thresholds of 50%, 55%, 60%,..., 95%), as well as the computations (FLOPs). Table 8 and Table 9 present the 5-fold cross-validation results of the default and proposed YOLOv9 for the MWPD dataset. Each fold was trained and validated on 598 images, and the model demonstrated consistent performance across all folds. The proposed model consistently outperformed the default configuration across all folds. It achieved an average precision of 0.92 and a recall of 0.90, indicating reliable detection performance with minimal false positives and false negatives. Moreover, the proposed model achieved a 3% higher average mAP@50, reaching 0.95, which indicates improved localization accuracy. Its average F1-score of 0.91 further demonstrates a well-balanced trade-off between precision and recall. These results

Table 7Data & training ablations on YOLOv9-c (5-fold mean \pm std).

Setting	mAP50-95	mAP50	F1	FPS ↑
No preproc, no augs	0.646 \pm 0.012	0.947 \pm 0.007	0.913 \pm 0.010	64.4 \pm 1.2
HE only	0.656 \pm 0.011	0.952 \pm 0.006	0.918 \pm 0.010	64.3 \pm 1.2
Augmentations only	0.659 \pm 0.011	0.953 \pm 0.006	0.920 \pm 0.009	64.2 \pm 1.3
Photometric only	0.661 \pm 0.011	0.954 \pm 0.006	0.921 \pm 0.009	64.1 \pm 1.2
Geometric only	0.657 \pm 0.012	0.952 \pm 0.006	0.919 \pm 0.010	64.2 \pm 1.2
Noise/blur only	0.653 \pm 0.012	0.951 \pm 0.006	0.918 \pm 0.010	64.3 \pm 1.3
HE + all augs (3x)	0.662 \pm 0.011	0.955 \pm 0.006	0.922 \pm 0.009	64.2 \pm 1.3
AdamW (vs. SGD)	0.666 \pm 0.010	0.957 \pm 0.006	0.924 \pm 0.009	63.0 \pm 1.4
Image size 512	0.649 \pm 0.012	0.949 \pm 0.006	0.916 \pm 0.010	76.0 \pm 1.5
Image size 768	0.671 \pm 0.010	0.960 \pm 0.006	0.928 \pm 0.009	55.1 \pm 1.1

Table 8

5-fold cross-validation results of default YOLOv9 model for MWPD dataset.

Metrics	KF1	KF2	KF3	KF4	KF5	Average
Images	598	598	598	598	598	–
Precision	0.92	0.88	0.91	0.92	0.87	0.90
Recall	0.90	0.83	0.88	0.90	0.89	0.88
mAP@50	0.93	0.89	0.92	0.92	0.91	0.92
mAP@50:95	0.61	0.56	0.57	0.61	0.58	0.58
F1-score	0.90	0.86	0.89	0.90	0.87	0.88
Preprocess	0.3	0.3	0.3	0.2	0.3	0.28
Inference	15.5	15.5	15.5	9.3	11.2	13.04
Postprocess	3.2	2.9	3.1	2.4	1.6	2.36

Table 9

5-fold cross-validation results of proposed YOLOv9 model for MWPD dataset.

Metrics	KF1	KF2	KF3	KF4	KF5	Average
Images	598	598	598	598	598	–
Precision	0.94	0.90	0.92	0.94	0.88	0.92
Recall	0.93	0.86	0.90	0.92	0.91	0.90
mAP@50	0.96	0.93	0.94	0.96	0.94	0.95
mAP@50:95	0.67	0.61	0.63	0.66	0.64	0.64
F1-score	0.93	0.90	0.91	0.93	0.89	0.91
Preprocess	0.3	0.2	0.2	0.2	0.2	0.22
Inference	14.4	13.5	13.3	9.3	9.4	11.98
Postprocess	3.1	2.6	2.4	2.3	1.3	2.34

confirm the robustness and effectiveness of the proposed model in accurately detecting potholes across diverse validation sets.

To demonstrate the effectiveness of the trained model, the model was tested on road images from the test dataset. All tests were performed using the same environment and parameters. Fig. 10 shows the observable outcomes of these assessments using the proposed YOLOv9 algorithms, displaying accurate detections. Figs. 10a–10l present the prediction outcomes of the proposed model, which identifies potholes effectively, achieving a maximum detection confidence of approximately 0.95. To further evaluate the model's robustness, predictions were made using test images of the MWPD dataset. Figs. 10a, 10f, and 10i display the model's detection performance under low-light conditions, where it successfully identifies a pothole despite limited illumination. Fig. 10e showcases the model's performance in a foggy environment, demonstrating consistent detection capability even with reduced visibility and blurred image features. In Figs. 10b, 10c, 10g, and 10l, the detection results in rainy weather conditions are presented. While the model maintains reasonable accuracy, Figs. 10g and 10h reveal a noticeable performance drop. Some potholes remain undetected due to the wet road surface and shadows, which diminish the contrast between the pothole region and the background, making detection more challenging during the daytime. Conversely, under nighttime conditions in Fig. 10l, the model occasionally misclassifies shadows as potholes, likely due to low illumination and the lack of clear texture information. These cases highlight the sensitiv-

ity of the model to lighting variations and suggest the need for additional preprocessing to improve robustness.

Strengths: As shown in Table 8 and Table 9, the improved YOLOv9 increases the F1-score from 0.88 to 0.91, mAP@50 from 0.92 to 0.95, and mAP@50:95 from 0.58 to 0.64 compared to the default YOLOv9. Beyond these quantitative gains, the proposed model demonstrates several additional strengths: it achieves higher precision and recall, reduces preprocessing, 0.28 ms to 0.22 ms and inference time, 13.04 ms to 11.98 ms, shows greater stability across folds, and lowers false detections under challenging conditions, making it more accurate and efficient for real-time applications.

4.6. Models performance comparison

Table 10 provides a comparative analysis between the proposed system and existing approaches, demonstrating the effectiveness and superiority of the proposed model. The evaluation considers different datasets, including the “Pothole Detection” dataset from Kaggle and Roboflow, and measures key performance metrics such as precision, recall, mAP@50, and mAP@50:95. The comparative results highlight the effectiveness of the proposed YOLOv9 model, which consistently surpasses YOLOv8 across all performance metrics. On the Kaggle dataset, the model achieved notable gains, 9.48% in precision, 5.26% in recall, and 3.53% in mAP@50, reflecting its enhanced detection capability in detecting potholes. Likewise, on the Roboflow dataset, YOLOv9 achieved 4.44% higher precision, a 2.47% improvement in recall, and a 2.3% improvement in mAP@50, demonstrating its effectiveness across different datasets. This enhancement is primarily attributed to the optimized layers of the model, which allow for better discrimination between potholes and background regions. Notably, the proposed model shows a significant improvement in mAP@50:95, particularly on the Kaggle dataset, where it outperforms previous work by a large margin (0.51 vs. 0.37). These results highlight the model's superiority in improving detection accuracy for real-world road scenarios. However, despite these improvements, the proposed model has certain limitations. It requires more computational power, and its performance may drop if trained on small datasets. Future research will aim to enhance the model's speed and efficiency for practical, real-world deployment. In summary, each approach offers distinct advantages, and selecting the appropriate model depends on the specific needs of the task.

5. Limitations and future works

The proposed system, while effective, has several limitations. First, the MWPD dataset may not include extreme weather conditions such as heavy fog or snowstorms, which could impact the model's performance in those scenarios. Second, the model's performance may degrade in complex urban environments with high visual clutter, such as heavy traffic, shadows, or occlusions, where distinguishing potholes becomes more challenging. Third, the cross-validation procedure did not employ a group-wise splitting strategy (GroupKFold), meaning that augmented variants of the same image may have been distributed across

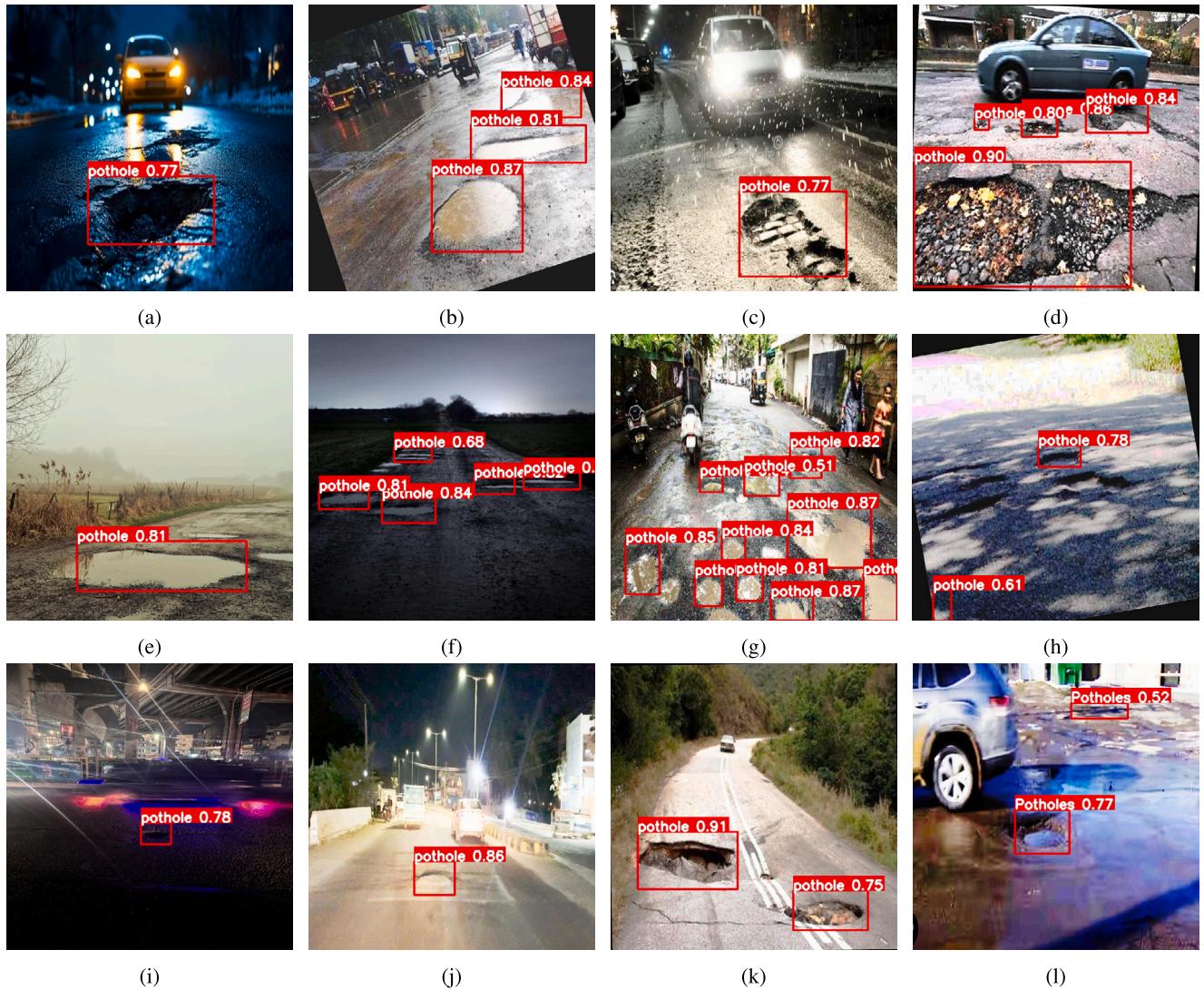


Fig. 10. Prediction results of the proposed on-road potholes detection system.

Table 10

Comparison between the proposed system with the existing systems having different datasets.

Reference	Dataset	No. of Images	Model	Performance measure				
				Precision (%)	Recall (%)	mAP (%)	mAP50-95	F1-score
Ruhil et al. [25]	Kaggle	2105	YOLOv8	82.2	76	85	0.37	0.81
Proposed	Kaggle	2105	YOLOv9	90	80	88	0.51	0.84
Khare et al. [40]	Roboflow	3770	YOLOv8	90	81	87	0.64	0.85
Proposed	Roboflow	3770	YOLOv9	94	83	89	0.62	0.88

different folds, potentially leading to minor data leakage and inflated validation scores. Future research can explore several directions to further enhance the model's performance and applicability. Expanding the MWPD dataset by incorporating more diverse weather scenarios, such as fog, snow, and varying lighting conditions, would further improve the model's ability to handle real-world environments. In addition, utilizing advanced annotation techniques may enhance detection accuracy. Evaluating the hardware feasibility of the model by measuring inference times on embedded systems, such as the NVIDIA Jetson Nano, is another essential step toward enabling real-time deployment on low-power devices. Integrating supplementary sensor data, such as LiDAR, could improve depth perception and enhance detection performance in

challenging weather conditions. Furthermore, extending the model to recognize and classify other road hazards and developing a feature to measure pothole dimensions and categorize them by severity could provide more actionable insights for infrastructure maintenance and road safety planning. Finally, future work will adopt a GroupKFold strategy to prevent data leakage and ensure more reliable performance evaluation.

6. Conclusion

This research demonstrates the effectiveness of an improved YOLOv9 deep learning model for detecting potholes under diverse weather con-

ditions. By creating a customized dataset (MWPD), combining images from various weather, including sunny, rainy, and nighttime environments, the proposed model conditions achieved a 3% improvement over the default YOLOv9, with an average mAP@50 of 95% and an F1-score of 91%. The proposed model exhibited strong generalizability across existing studies, confirming its potential for real-time applications in intelligent transportation systems, autonomous driving, and road maintenance.

CRediT authorship contribution statement

Shahnaj Parvin: Writing – original draft, Formal analysis, Conceptualization. **Foysal Munsy:** Conceptualization. **Md Tanzeem Rahat:** Formal analysis. **Aminun Nahar:** Methodology. **Kamruddin Nur:** Writing – review & editing. **Debasish Ghose:** Writing – review & editing.

Declaration of competing interest

The authors declare that this work is original and has not been published elsewhere. Additionally, there are no conflicts of interest concerning the publication of this paper.

Data availability

This study utilizes a publicly available dataset titled Multi-Weather Pothole Detection Dataset, hosted on Mendeley Data and introduced by [7].

References

- [1] Krishna Singh Basnet, Jagat Kumar Shrestha, Rabindra Nath Shrestha, Pavement Performance Model for Road Maintenance and Repair Planning: a Review of Predictive Techniques, 2023.
- [2] Jaroslav Frndá, Srijita Bandyopadhyay, Michal Pavlicko, Marek Durica, Mihails Savrasovs, Soumen Banerjee, Analysis of pothole detection accuracy of selected object detection models under adverse conditions, *Transp. Telecommun. J.* 25 (2) (April 2024) 209–217, <https://doi.org/10.2478/ttj-2024-0016>.
- [3] World Health Organization, Global status report on road safety 2023, <https://www.who.int/teams/social-determinants-of-health/safety-and-mobility/global-status-report-on-road-safety-2023>, 2023. (Accessed 11 January 2024).
- [4] Yuying Mao, Wenzhe Su, Haotian Chen, Pothole road detection and identification based on transfer learning, in: Lijun Wu, Zhongpan Qiu (Eds.), Fourth International Conference on Sensors and Information Technology (ICSI 2024), vol. 13107, International Society for Optics and Photonics, SPIE, 2024, pages 13107R-1–13107R-6.
- [5] M. Sheeta, K. Prasanna, Intelligent deep learning based pothole detection and alerting system, *Int. J. Comput. Intell. Res.* 19 (1) (2023) 25–35, <https://doi.org/10.3762/IJCI.19.1.2023.25-35>.
- [6] Jinlei Wang, Rui Feng Meng, Yuanhao Huang, Lin Zhou, Lujia Huo, Zhi Qiao, Changchang Niu, Road defect detection based on improved yolov8s model, *Sci. Rep.* 14 (1) (Jul 2024) 16758, <https://doi.org/10.1038/s41598-024-67953-3>.
- [7] **Shahnaj Parvin, Foysal Munsy, Kamruddin Nur, Multi-Weather Pothole Detection — data.mendeley.com**, 2025.
- [8] Songling Huang, Hao Chen, Lingbo Yan, Xiaoling Zou, Bin Li, Yanqiu Bi, A Review of the Progress in Machine Vision-Based Crack Detection and Identification Technology for Asphalt Pavements, 2025.
- [9] Boris Bučko, Eva Lieskovská, Katarína Zábovká, Michal Zábovský, Pothole detection using computer vision in challenging conditions, in: Figshare, 2022.
- [10] S. Patel, Pothole image data-set, in: Kaggle, 2019, <https://www.kaggle.com/datasets/sachinpatel21/pothole-image-dataset>.
- [11] GeraPotHole, Pothole detection yolov8 dataset, <https://universe.roboflow.com/gerapothole/pothole-detection-yolov8>, apr 2023.
- [12] Chemikala Saisree, U. Kumaran, Pothole detection using deep learning classification method, in: International Conference on Machine Learning and Data Engineering, Proc. Comput. Sci. (ISSN 1877-0509) 218 (2023) 2143–2152, <https://doi.org/10.1016/j.procs.2023.01.190>.
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep Residual Learning for Image Recognition, 2015.
- [14] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, Alex Alemi, Inception-V4, Inception-Resnet and the Impact of Residual Connections on Learning, 2016.
- [15] Karen Simonyan, Andrew Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, 2015.
- [16] Lopamudra Panda, Kanneganti Bhavya Sri, Reeja SR, Real time pothole detection system – an application facilitating public safety, in: 2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE), 2023, pp. 389–395.
- [17] Linchao Li, Jiazen Liu, Jiabao Xing, Zhiyang Liu, Kai Lin, Bowen Du, Road pothole detection based on crowdsourced data and extended mask r-cnn, *IEEE Trans. Intell. Transp. Syst.* PP:1–13 (2024) 09, <https://doi.org/10.1109/TITS.2024.3360725>.
- [18] Kanchi Anantharaman Vinodhini, Kovilvenni Ramachandran Aswin Sidhaartha, Pothole detection in bituminous road using cnn with transfer learning, *Meas. Sens.* (ISSN 2665-9174) 31 (2024) 100940, <https://doi.org/10.1016/j.measen.2023.100940>.
- [19] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, Generative Adversarial Networks, 2014.
- [20] Boris Bučko, Eva Lieskovská, Katarína Zábovká, Michal Zábovský, Computer vision based pothole detection under challenging conditions, *Sensors* 22 (22) (2022), <https://doi.org/10.3390/s22228878>.
- [21] Shaogang Ren, Kaiming He, Ross Girshick, Jian Sun, Faster r-cnn: Towards Real-Time Object Detection with Region Proposal Networks, 2016.
- [22] Malhar Khan, Muhammad Amir Raza, Ghulam Abbas, Salwa Othmen, Amr Yousef, Touqeer Ahmed Jumani, Pothole detection for autonomous vehicles using deep learning: a robust and efficient solution, *Front. Built Environ.* 9 (2024) 1323792, <https://doi.org/10.3389/fbuil.2023.1323792>.
- [23] Riddhi Mirjakar, Anuradha Yenkar, Shreyash Nawalkar, Rishabh Kaul, Aditya Rokade, Kedarnath Rothe, Enhanced pothole detection in road condition assessment using yolov8, in: 2024 IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE), 2024, pp. 429–433.
- [24] Jiarui Chang, Zhan Chen, E. Xia, Improved yolov8 method for multi-scale pothole detection, in: Advanced Intelligent Computing Technology and Applications: 20th International Conference, Proceedings, Part XI, ICIC 2024, Tianjin, China, August 5–8, 2024, Springer-Verlag, Berlin, Heidelberg, ISBN 978-981-97-5611-7, 2024, pp. 383–395.
- [25] Nidhi Ruhil, Devansh Sahni, Anushaka, Anurag Wadhwa, Anjali Sharma, Pothole detection and reporting system implementation using yolov8 and tensorflow.js, *Int. J. Comput. Sci. Eng.* 11 (12) (2023) 26–31, https://ijcseonline.isroset.org/pub_paper/5-IJCSE-09292.pdf.
- [26] Hindol Bhattacharjee, Shaik Shaik, Javed an Nasreen, E. Varshith Reddy, Pothole detection using machine learning (yolov4), *Int. Res. J. Modern. Eng. Technol. Sci.* 6 (3) (2024) 4725–4728, <https://doi.org/10.56726/IRJMETS51256>.
- [27] Kamalamkannan S, Navaneethan S, Yogesh S. Deshmukh, Deshmukh Sujay V, Sivakami Sundari M, Venkadesan Ramalingam Jagadeesan D, Venkatesh C, A novel pothole detection model based on yolo algorithm for vanet, *Int. J. Intell. Syst. Appl. Eng.* 12 (11s) (2024) 56–61, <https://ijisae.org/index.php/LIJSAE/article/view/4419>.
- [28] Zachary Jeffreys, Kshama Kumar, Zhuojing Xie, Wan D. Bae, Shayma Alkobaisi, Sada Narayanappa, Potholevision: an automated pothole detection and reporting system using computer vision, in: Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing, SAC '24, Association for Computing Machinery, New York, NY, USA, ISBN 9798400702433, 2024, pp. 695–697.
- [29] Raghavendra K, Sakshi Shankar Gumaste, Sanjana Singh, Tejas K, Varsha N, Roadeye-a yolov12-based approach for real-time road pothole detection, *Int. Res. J. Eng. Technol.* 12 (5) (2025).
- [30] Satish Kumar Satti, Goluguri N.V. Rajareddy, Kaushik Mishra, Amir H. Gandomi, Potholes and traffic signs detection by classifier with vision transformers, *Sci. Rep.* (ISSN 2045-2322) 14 (1) (Jan 2024) 2215, <https://doi.org/10.1038/s41598-024-52426-4>.
- [31] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg, SSD: Single Shot MultiBox Detector, Springer International Publishing, ISBN 9783319464480, 2016, pp. 21–37.
- [32] Bhargav Ram Gowrisetty, Harees Reddy Emani, Vineeth Kumar Ganta, Doushik Chinthapudi, Shaik Riyazuddien, Pothole detection and dimension estimation using deep learning (yolo) and image processing, *Int. J. Adv. Res. Innov. Ideas Educ.* (ISSN 2395-4396) 10 (2) (2024) 748–754, <https://www.ijariie.com>.
- [33] Saahas Myla Myla, Pothole data analysis using deep learning, *SSRN* 75 (2024) 1863–1881, <https://doi.org/10.2139/ssrn.4810248>.
- [34] Qian Li, Yanjuan Shi, Qing Liu, Gang Liu, Deep learning-based pothole detection for intelligent transportation: a yolov5 approach, *Int. J. Adv. Comput. Sci. Appl.* 14 (12) (2023), <https://doi.org/10.14569/IJACSA.2023.0141242>.
- [35] Srividhya Ramisetty, Sanjana R, Satish Kanti, Simriti H, Real time pot hole detection for safe roadways, in: 2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), 2023, pp. 1–7.
- [36] Aditya Singh, Aryan Mehta, Ali Asgar Padaria, Nilesh Kumar Jadav, Rebakah Geddam, Sudeep Tanwar, Enhanced pothole detection using yolov5 and federated learning, in: 2024 14th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2024, pp. 549–554.
- [37] Sophia S, Ajish Moses Raj A, R. Janani, Stewart Kirubakaran S, Dhivinkumar AJ, Princely Nesaraj A, Integrating Google maps and deep learning in path hole detection alert system, in: 2024 4th International Conference on Pervasive Computing and Social Networking (ICPCSN), 2024, pp. 1006–1012.
- [38] Smt.A. Koteswaramma, M. Divya Sri, M. Manohar, K. Srihari, M. Yabbeju, Advancing road safety: pothole detection using yolov8 and wandb deep learning, *J. Adv. Zool.* 45 (S2) (2024) 116–122, <https://doi.org/10.53555/jaz.v45iS2.3848>.
- [39] Y. Baby Kalpana, E. Subhashini, A.S. Nashrin Taaj, D. Punithakala, Detection and notification of potholes and humps on roads to aid drivers using iot, *Int. J. Multidiscipl. Res. Sci. Eng. Technol.* 7 (13) (2024) 155–162.

- [40] Om Khare, Shubham Gandhi, Aditya Rahalkar, Sunil Mane, Yolov8-based visual detection of road hazards: potholes, sewer covers, and manholes, in: 2023 IEEE Pune Section International Conference (PuneCon), 2023, pp. 1–6.
- [41] Mayank Dhingra, Rahul Dhingra, Meghna Sharma, Pothole detection using machine learning models, Int. J. Sci. Res. Sci. Eng. Technol. 11 (2) (2024) 94–105, <https://doi.org/10.32628/IJSRSET241126>.
- [42] Dinesh Swami, Mahesh Jangid, Pothole detection and prediction using deep learning with cnn and yolov8, in: Rajesh Kumar, Ajit Kumar Verma, Om Prakash Verma, Tanu Wadehra (Eds.), Soft Computing: Theories and Applications, Nature Singapore Springer, Singapore, ISBN 978-981-97-2031-6, 2024, pp. 321–334.
- [43] M. Divya, G. Divyashree, B. Uma Maheswari, Pothole detection using yolov8 and cnn, in: 2024 3rd International Conference for Innovation in Technology (INOCON), 2024, pp. 1–6.
- [44] Hong-Hu Chu, Muhammad Rizwan Saeed, Javed Rashid, Muhammad Tahir Mehmood, Rao Ahmad, Sohail Iqbal, Ghulam Ali, Deep learning method to detect the road cracks and potholes for smartcities, Comput. Mater. Continua (ISSN 1546-2226) 75 (1) (2023) 1863–1881, <https://doi.org/10.32604/cmc.2023.035287>.
- [45] Shafi Ullah Adid, Md. Emon, Taofica Amrine, A hybrid approach to detect and classify pothole on Bangladeshi roads using deep learning, Int. J. Sci. Res. Arch. 12 (01) (2024) 1045–1053, <https://doi.org/10.30574/ijjsra.2024.12.1.0950>.
- [46] S. Kanaga Suba Raja, B. Chandra, Sharon Hajhan, Prassanth, Tamilvanan, Pothole detection and compliant notification, AIP Conf. Proc. 2802 (1) (2024), <https://doi.org/10.1063/5.0184610>.
- [47] B. Srivani, Ch. Kamala, S. Renu Deepthi, G. Aakash, Pothole detection using convolutional neural network, AIP Conf. Proc. 2935 (1) (2024), <https://doi.org/10.1063/5.0198902>.
- [48] Arvindh Kumar Selvam, G.Y.R. Vikhram, A real-time cnn-based pothole detection system for road safety, Int. Adv. Res. J. Sci. Eng. Technol. 11 (4) (2024) 711–717, <https://doi.org/10.17148/IARJSET.2024.114105>.
- [49] A. Lincy, G. Dhanarajan, S. Sanjay Kumar, B. Gobinath, Road pothole detection system, ITM Web Conf. 53 (2023) 01008, <https://doi.org/10.1051/itmconf/20235301008>.
- [50] Junkui Zhong, Deyi Kong, Yuliang Wei, Bin Pan, Yolov8 and point cloud fusion for enhanced road pothole detection and quantification, Sci. Rep. 15 (1) (Apr 2025) 11260, <https://doi.org/10.1038/s41598-025-94993-0>.
- [51] Y. Aneesh Chowdary, V. Sai Teja, V. Vamsi Krishna, N. Venkaiah Naidu, R. Karthika, Pothole Detection Approach Based on Deep Learning Algorithms, Soft Computing and Signal Processing, vol. 313, Springer Nature Singapore, Singapore, ISBN 978-981-19-8669-7, 2023, pp. 597–606.
- [52] Sarthak Babbar, Jatin Bedi, Real-time traffic, accident, and potholes detection by deep learning techniques: a modern approach for traffic management, Neural Comput. Appl. (ISSN 1433-3058) 35 (26) (Sep 2023) 19465–19479, <https://doi.org/10.1007/s00521-023-08767-8>.
- [53] Mohd Omar, Pradeep Kumar, Pd-its: pothole detection using yolo variants for intelligent transport system, SN Comput. Sci. (ISSN 2661-8907) 5 (5) (May 2024) 552, <https://doi.org/10.1007/s42979-024-02887-1>.
- [54] P.D.S.S. Lakshmi Kumari, Gidugu Srinija Sivasatya Ramacharanteja, S. Suresh Kumar, Gorrela Bhuvana Sri, Gottumukkala Sai Naga Jyotsna, Aki Hari Keerthi Naga Safalya, Developing an automated system for pothole detection and management using deep learning, in: Advanced Communication and Intelligent Systems, Springer Nature Switzerland, Cham, ISBN 978-3-031-45124-9, 2023, pp. 12–22.
- [55] Danyan Xie, Wenyi Yao, Wenbo Sun, Zhenyu Song, Real-time identification of strawberry pests and diseases using an improved yolov8 algorithm, Symmetry 16 (10) (2024), <https://doi.org/10.3390/sym16101280>.
- [56] Chien-Yao Wang, I-Hau Yeh, Hong-Yuan Mark Liao, Yolov9: learning what you want to learn using programmable gradient information, <https://arxiv.org/abs/2402.13616>, 2024.
- [57] Joseph Redmon, Ali Farhadi, Yolov3: an incremental improvement, <https://arxiv.org/abs/1804.02767>, 2018.
- [58] Bin Yan, Pan Fan, Xiaoyan Lei, Zhijie Liu, Fuzeng Yang, A real-time apple targets detection method for picking robot based on improved yolov5, Remote Sens. (ISSN 2072-4292) 13 (9) (2021), <https://doi.org/10.3390/rs13091619>.
- [59] Chien-Yao Wang, Alexey Bochkovskiy, Hong-Yuan Mark Liao, Yolov7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, in: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 7464–7475.
- [60] Fatma M. Talaat, Hanaa ZainEldin, An improved fire detection approach based on yolo-v8 for smart cities, Neural Comput. Appl. 35 (28) (2023) 20939–20954, <https://doi.org/10.1007/s00521-023-08809-1>.
- [61] Rahima Khanam, Muhammad Hussain, Yolov11: an overview of the key architectural enhancements, <https://arxiv.org/abs/2410.17725>, 2024.
- [62] Yunjie Tian, Qixiang Ye, David Doermann, Yolov12: attention-centric real-time object detectors, <https://arxiv.org/abs/2502.12524>, 2025.
- [63] Chien-Yao Wang, Hong-Yuan Mark Liao, I-Hau Yeh, Designing network design strategies through gradient path analysis, J. Inf. Sci. Eng. 39 (4) (2023) 975–995, [https://doi.org/10.6688/JISE.202307_39\(4\).0016](https://doi.org/10.6688/JISE.202307_39(4).0016).
- [64] Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, I-Hau Yeh, CspNet: a new backbone that can enhance learning capability of cnn, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 1571–1580.
- [65] Hafedh Mahmoud Zayani, Unveiling the potential of yolov9 through comparison with yolov8, Int. J. Intell. Syst. Appl. Eng. 12 (3) (Mar. 2024) 2845–2854, <https://ijisae.org/index.php/IJISAE/article/view/5794>.
- [66] Tianyong Wu, Youkou Dong, Yolo-se: improved yolov8 for remote sensing object detection and recognition, Appl. Sci. (ISSN 2076-3417) 13 (24) (2023), <https://doi.org/10.3390/app132412977>.
- [67] Prasanta Das, Angshuman Chakraborty, Ravi Sankar, Om Krishan Singh, Hena Ray, Alokes Ghosh, Deep learning-based object detection algorithms on image and video, in: 2023 3rd International Conference on Intelligent Technologies (CONIT), 2023, pp. 1–6.