Real-Time Pothole Detection Using YOLO-Based Deep Learning Training Analysis and Performance Evaluation

1st Shaik Basheer[1], 2nd Shaik Abdul Gaffoor Hussain[2], 3rd Dunga Priyatham[3],

4th Turlapati Nithin Kumar[4], 5th Shaik Abdul Karim[5]

[1]CSE & SRM-AP, India

[2]CSE & SRM-AP, India

[3]ECE & SRM-AP, India

[4]CSE & SRM-AP, India

[5]CSE & SRM-AP, India

[1]1st shaikbasheer_c@srmap.edu.in; [2]2nd abdulgaffoorhussain_shaik@srmap.edu.in;
[3]3rd priyatham_dunga@srmap.edu.in; [4]4th nithin_turlapati@srmap.edu.in;
[5]5th karim_shaik@srmap.edu.in

Abstract*— Potholes pose a serious threat to road safety and car performance, which in many cases culminates in accidents, traffic disruptions, and accelerated deterioration of infrastructure. Traditional methods of inspection are very manual and do not have scalability requirements for dealing with large and dynamic road networks. To address these shortcomings, this research proposes a real time hole detection framework based on five YOLO architectures: 'YOLOv5', YOLOv10, which trains and tests using uniform based on a custom dataset. Among these models, YOLOv9-T achieved the most favourable balance between detection accuracy and computational efficiency, with an mAP@0.5 and a peak F1-score of 0.73. The optimized YOLOv9-T which was further deployed on an NVIDIA Jetson platform to validate real-time performance on a device with 18–25 FPS on live video feeds. Experimental results show a high degree of robustness in challenging outdoor environments, including situations where shadow, glare, wet lanes, and non-homogeneous lane markings. In summary, the proposed system provides a feasible and low latency solution well-suited for smart city infrastructure, automated roadway surveillance systems, and edge-intelligent transport solutions systems.

***Keywords*— YOLO, deep learning, object detection, pothole detection, real-time systems**

I. INTRODUCTION

The most important tool for facilitating national development through integration with transportation and providing economic activity through what would be transported on the roadway is road infrastructure. Unfortunately, road infrastructure continues to deteriorate across most areas and still continues to be plagued by many problems; one of these problems is the lack of ability to enter the roadway safely into the intersection, more specifically, by the presence of potholes. Potholes are considered one of the most dangerous types of road damage and can lead to the breakdown of vehicles, increased time spent on the road, and a large number of accidents that take place on the roadway each year. As reported by the World Bank as well as the Ministry of Road Transport and Highways (MoRTH, 2022), many of the incidents that are related to vehicles and cause the greatest number of accidents, injuries, and deaths on roadways in India are the direct result of a damaged or poor pavement surface. As a result of this, reliable, scalable, and continuous systems are needed to track the current conditions of roadways.

Historically, traditional means of detecting potholes have been done primarily through human inspection and periodic scheduled site surveys in the field. These methods are easy to put into practice; however, they are not a good alternative for larger or frequently changing road networks due to the amount of time and labour it would take for an individual inspector to physically visit each and every site. In addition to this, inspectors' ability to assess the roadway is dependent on human judgement, which can change day-to-day based on fatigue, weather conditions, and visibility. In addition to these limitations to effective inspection, previous attempts to automate the pothole-detection process through classical computer vision techniques such as edge detection, thresholding, and custom texture-based features, have been limited by several factors. These factors include limited performance and the inability to track and assess the conditions of the roadway over a prolonged period of time. [1],[2].

Recent advancements in Artificial Intelligence (AI) and Machine Learning (ML) have significantly transformed intelligent infrastructure management. Deep learning-based computer vision algorithms have also demonstrated robust performance in detecting pavement damage like cracks and potholes from images as well as video streams.

Key developments include:

• Based on the data gathered from vehicles, unmanned aerial vehicles, and roadside cameras for the purpose of monitoring roads on a constant and larger scale.

• The launch of publicly available annotated datasets, including RDD2020 and RDD2022, that has accelerated the process of benchmarking and comparative studies in road damage detection[4].

• The design and implementation of automated detection solutions that help the authorities in proactive maintenance, resource allocation, and road safety through smart cities.

Among the many models developed using deep learning techniques for object detection, the YOLO family of models has drawn significant attention for its ability to work in real-time. This is because the YOLO models are able to perform both object localization and classification within one pass. Over successive versions—such as YOLOv5[6], YOLOv7[7], and YOLOv10[9]—the architecture has been continuously refined to improve accuracy, inference speed, and generalization capability. Those are very desirable features, especially when the model would be deployed at the edge, such as in dashcams, on a mobile platform, or with an embedded GPU in roadside monitoring units, where computation resources may be scant.

These developments have motivated this work to focus on the design and evaluation of a YOLO-based real-time pothole detection system that would operate under most environmental conditions and various road textures. Five YOLO architectures will be trained and evaluated, namely, YOLOv5, YOLOv6, YOLOv7, YOLOv9-T, and YOLOv10, under the same experimental settings to facilitate a fair and systematic comparison. Moreover, the model with the best performance will be implemented on an NVIDIA Jetson embedded platform to estimate its real-time inference capability and practical usability. The results prove the efficiency of the proposed approach for applications in automated road inspection, on-vehicle monitoring, and smart-city infrastructure management.

A. Key Contributions

The main contributions of this study are given as follows:

• Development of a diverse pothole dataset consisting of 2,500 images captured under different lighting, weather, and pavement conditions.

• Systematic and fair evaluation of five YOLO architectures (YOLOv5, YOLOv6, YOLOv7, YOLOv9-T, YOLOv10) trained with identical hyperparameters to ensure unbiased comparison.

• Extensive performance analysis using evaluation metrics such as mAP, F1-score, precision–recall curves, confidence distribution, confusion matrix, and qualitative detection results.

• Deployment of the optimized YOLOv9-T model on an NVIDIA Jetson embedded platform, achieving real-time inference performance of 18–25 FPS.

• Demonstration of a robust and deployment-ready pothole detection pipeline suitable for vehicular systems, UAV-based inspections, and smart-city road-condition monitoring applications.

Fig. 1 Example of a road with multiple potholes

B. Related Work

Early attempts at pothole detection mainly relied on conventional image-processing techniques such as edge detection, thresholding, and texture-based analysis [1], [2]. These methods are computationally lightweight; however, they are highly sensitive to variations in illumination, shadows, surface texture, and camera viewpoints. As a result, their performance degrades significantly in real-world outdoor environments. In addition, the dependence on hand-crafted features limits their ability to generalize across different road types and pavement conditions. Road-damage detection has been greatly enhanced by recent advancements in deep learning, with convolutional neural networks (CNNs) playing a crucial role by directly learning significant visual features from image data. Patch-wise analysis and sliding-window techniques were the mainstays of early CNN-based methods. Despite their effectiveness, these techniques were too computationally demanding to be used for real-time road monitoring. Faster R-CNN and other two-stage detection frameworks [3] increased detection accuracy by adding region proposal mechanisms, but their slow inference speed made them unsuitable for ongoing, real-time inspection.

Research progressively shifted to single-stage object detection models, especially the YOLO (You Only Look Once) family, which is specifically made for real-time inference, in order to get around these restrictions. Promising results for pothole detection were shown by early YOLO versions, such as YOLOv3 and YOLOv4, particularly in vehicle-mounted camera systems.[4], [5].YOLOv5, which added a lightweight architecture, quicker training convergence, and enhanced compatibility with edge and embedded platforms, further reinforced this development.[6].

More sophisticated backbone designs, better feature fusion techniques, and improved gradient propagation were included in later iterations like YOLOv7 and YOLOv8. Under difficult real-world circumstances, such as low light levels, wet or reflective road surfaces, uneven pavement textures, and partial occlusions, these improvements produced more accurate and reliable detections. [7], [8]. By adding a redundancy-free architecture and dynamic label assignment mechanisms, the most recent version, YOLOv10, further advances this evolution and improves the trade-off between detection accuracy and inference speed for high-performance real-time road inspection applications. [9].

Despite these ongoing improvements, distinguishing potholes from similar road features like cracks, repair patches, shadows, and water-filled depressions is still a major challenge. Differences in lighting, pavement materials, and camera angles often lead to inconsistent detection results. These issues highlight the need for a systematic and thorough comparison of current YOLO models, from YOLOv5 to YOLOv10, to find the most reliable and accurate model ready for real-time pothole detection in various real-world settings.

II. METHODOLOGY

THIS PAPER AIMS TO COMPARE FIVE DIFFERENT YOLO-BASED REAL-TIME POTHOLE DETECTION MODELS DEVELOPED USING YOLOV5, YOLOV6, YOLOV7, YOLOV9-T, AND YOLOV10 ARCHITECTURES. AMONG THESE COMPARED ARCHITECTURES, YOLOV9-T HAS A VERY EXCELLENT REAL-TIME DETECTION RATE AND IS MUCH MORE STABLE IN THE TRAINING PATTERN THAN THE OTHER COMPARED MODELS. ALL THE MODELS HAVE BEEN TRAINED AND EVALUATED USING EXACTLY THE SAME DATASET, TRAINING PARAMETERS, AND EVALUATION METRICS FOR EQUIVALENCIES.

A. Overview of the Proposed Framework

The proposed pothole detection system is designed as a five-stage pipeline involving dataset collection, data preprocessing and annotation, YOLO model training, performance evaluation, and comparative analysis. The flow of this experiment provides consistency and objectivity in comparison among different YOLO architecture variations. The overall framework that has been used in this study is depicted in Fig. 2

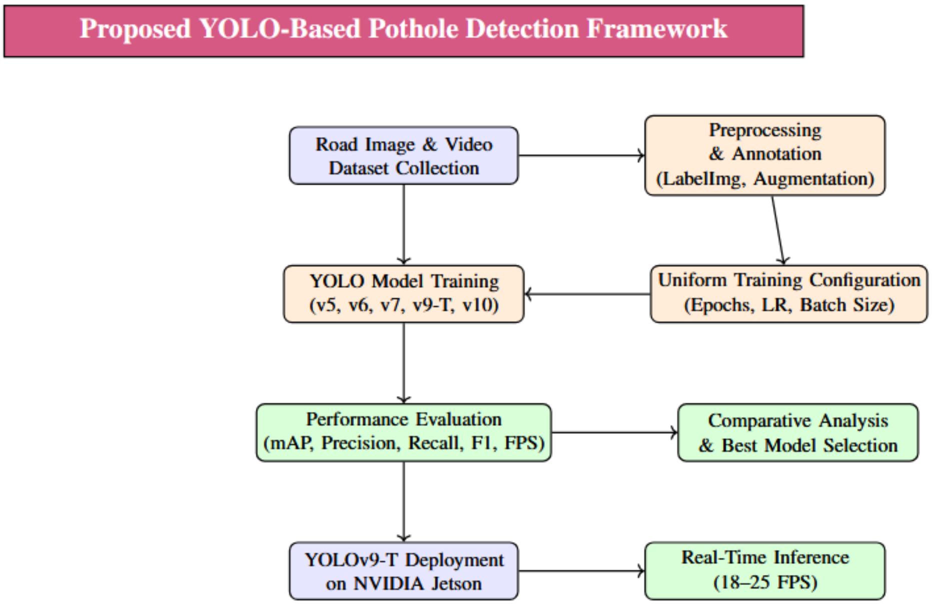

Fig. 2 Block diagram of the proposed YOLO-based real-time pothole detection methodology.

B. Dataset Development and Augmentation

The pothole image dataset with 2,500 images was created from the collected smartphone camera images, vehicle-mounted dash cameras, and publicly accessible road damage datasets [4]. All the images are annotated in YOLO format using the LabelImg annotation tool [12].

The variety of environments experienced in the real world is captured in the data set with lighting variations, weather impacts, road type represented through texture, and occlusions. The salient features of the data set are as follows:

- Total number of images: 2,500
- Training/Validation/Testing split: 70% / 20% / 10%
- Total annotated pothole instances: 4,860

To improve the robustness of models and generalization, some data augmentation techniques have been employed, including horizontal flips, random scale transformation, brightness and contrast adjustment, simulation of shadows and glare, and motion blur simulation.

C. YOLO Architecture Selection

Five object detection YOLO-based models were chosen for comparative testing, each from a different generation of architectural development:

- YOLOv5 [6]: ESPNet uses a CSP backbone with an SPPF module for efficient feature extraction
- YOLOv6: Uses an EfficientRep backbone network along with a detection head that supports decoupling.
- YOLOv7 [7]: Integrates the E-ELAN architecture for better feature fusion and gradient propagation
- YOLOv9-T: Makes use of the GELAN framework along with Programmable Gradient Information (PGI) for greater detection accuracy.
- YOLOv10 [9]: It adopts an architecture devoid of redundancy to carry out optimized real-time inferences.

Table I

EVOLUTION AND COMPARISON OF YOLO ARCHITECTURES.

Version	Key Module	Advantages	Limitations
YOLOv5	CSP + SPPF	Lightweight, fast inference, stable training	Older architecture
YOLOv6	Decoupled Head	Improved classification accuracy	Limited adoption
YOLOv7	E-ELAN	Enhanced feature fusion	Moderate inference cost
YOLOv9-T	GELAN + PGI	High accuracy, strong gradient flow	Slightly heavier model
YOLOv10	Redundancy-free design	Edge-efficient, real-time optimized	Limited experimental studies

D. Training Configuration

All models were trained using the same set of hyper parameters

- Epochs: 100
- Batch size: 16
- Optimizer: SGD (momentum 0.9)
- Learning rate: cosine decay (0.01 → 1e − 4)

The process of training included tracking the losses of box regression, classification, as well as objectness. For evaluation purposes, mean Average Precision (mAP), precision, recall, as well as F1 score metric.

E. Evaluation Procedure

Evaluation included:

- Precision–Recall curves
- F1–confidence analysis
- Confusion matrices
- Inference speed (FPS)
- Qualitative detection visualization

F. Mathematical Formulation

The IoU is defined as:

$$IoU = \frac{|B_{pred} \cap B_{gt}|}{|B_{pred} \cup B_{gt}|} \tag{1}$$

The GIoU loss is:

$$\mathcal{L}_{GIoU} = 1 - \left(IoU - \frac{|C - (B_{pred} \cup B_{gt})|}{|C|} \right) \tag{2}$$

The total YOLO loss is given by:

$$\mathcal{L}_{total} = \lambda_{box}\mathcal{L}_{GIoU} + \lambda_{obj}\mathcal{L}_{obj} + \lambda_{cls}\mathcal{L}_{cls} \tag{3}$$

Performance metrics include:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

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

G. Real-Time Deployment on NVIDIA Jetson

The YOLOv9-T, being the best-performing model, was tested on the NVIDIA Jetson embedded system. Real-time processing of 18–25 FPS was achieved with the aid of a live video stream from the camera. This makes it highly appropriate for use in vehicles, drone-based inspections, and smart cities for monitoring roads.

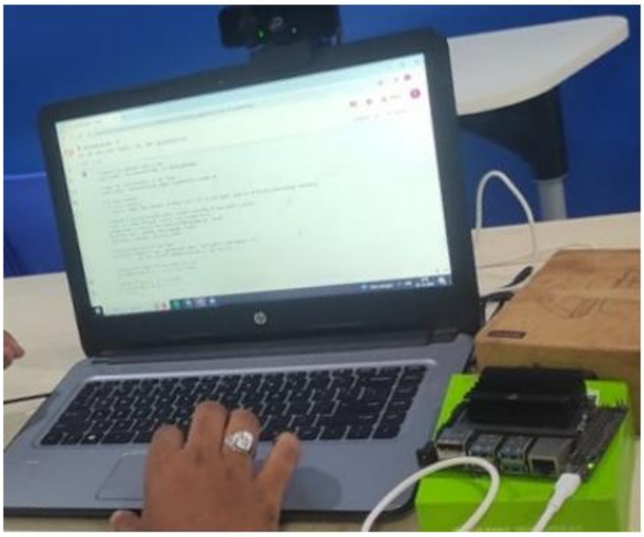

Fig. 3 Real-time deployment setup using NVIDIA Jetson.

III. Results and Discussion

This section presents both quantitative and qualitative evaluations of five YOLO architectures—YOLOv5, YOLOv6,YOLOv7, YOLOv9-T, and YOLOv10—trained on the custom pothole dataset. Among these models, YOLOv9-T showed the most consistent and strong performance and was therefore selected as the primary reference. The evaluation includes confusion matrix analysis, confidence-based behavior, precision–recall performance, and real-time deployment results.

A. Confusion Matrix Analysis

The confusion matrix for YOLOv9-T is normalized and shown in Fig. 4. The model achieved a true positive rate of 0.74 in pothole detection with background classification accuracy at 1.00. The confusion matrix summarizes the detection outcomes in terms of true positives, false positives, false negatives, and true negatives. A high true positive rate indicates effective pothole identification while near-perfect background accuracy confirms the ability of the model to suppress false alarms caused by visually similar road artifacts.

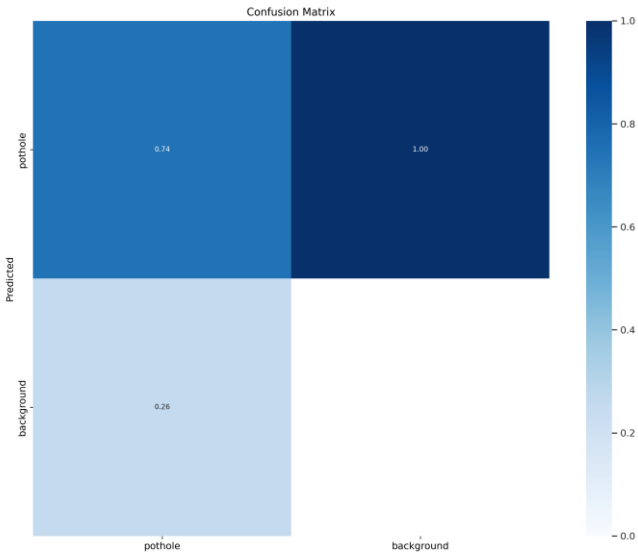

Fig. 4 Normalized confusion matrix for YOLOv9-T.

B. F1–Confidence Relationship

The F1–confidence curve in Fig. 5 shows that YOLOv9-T achieves its highest F1-score of 0.73 at a confidence threshold of 0.291. The F1-score is the harmonic mean of precision and recall, offering a well-rounded assessment of the detection performance. As it is required that both false positives and false negatives be kept small, especially for applications related to road conditions, the F1-score becomes even more important.

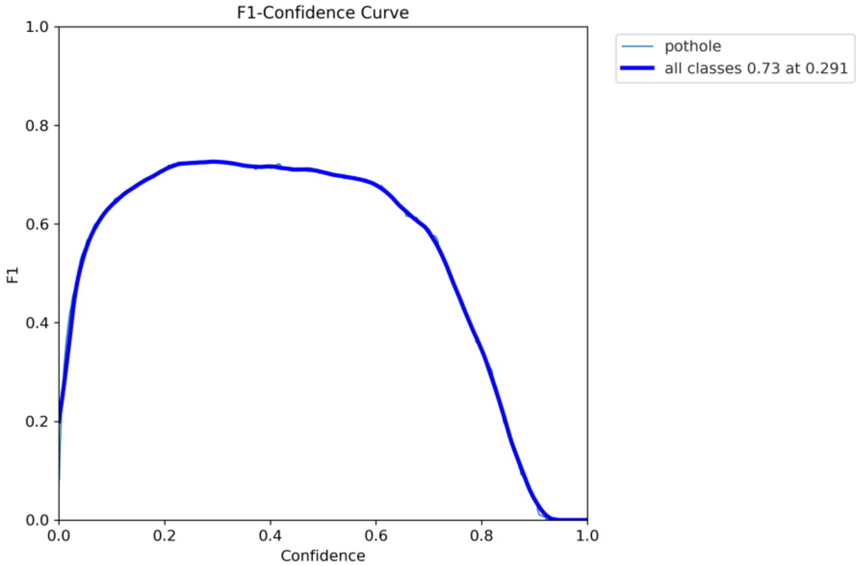

Fig. 5 F1-Confidence curve for YOLOv9-T(Peak F1 = 0.73)

C. Precision and Recall Behavior

Fig. 6 and **Fig. 7** illustrate the variation of precision and recall across different confidence thresholds.The observations has give as follows:

• The model attains a maximum **precision of 1.00** at a confidence threshold of 0.801.

• A high **recall value of 0.93** is achieved at lower confidence thresholds.

These results indicate that increasing the confidence threshold leads to highly reliable detections with very few false positives,while lower thresholds improve detection sensitivity by capturing most of the true pothole instances.

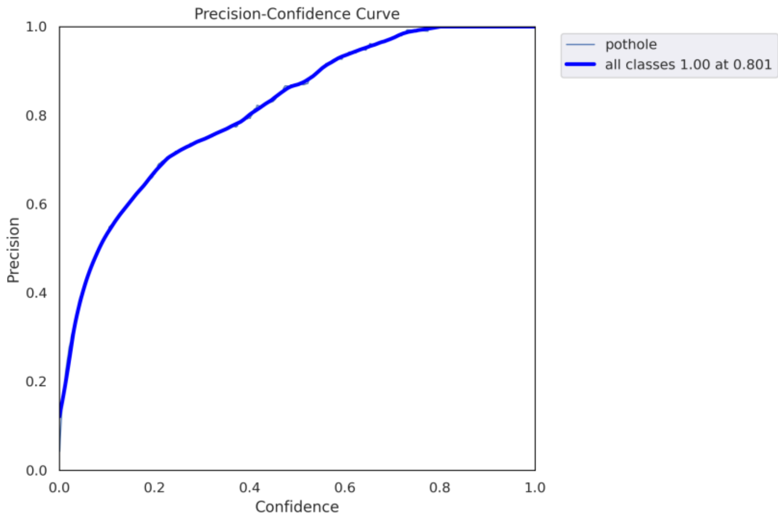

Fig. 6 Precision-Confidence curve for YOLOv9-T

Precision indicates the accuracy of the detected potholes, which signifies the ratio of the predicted to the actual potholes, while the recall indicates the model's capability for the detection of all potholes. This observed phenomenon ensures that the model, YOLOv9-T, maintains the balance of pothole detection accuracy as well as coverage.

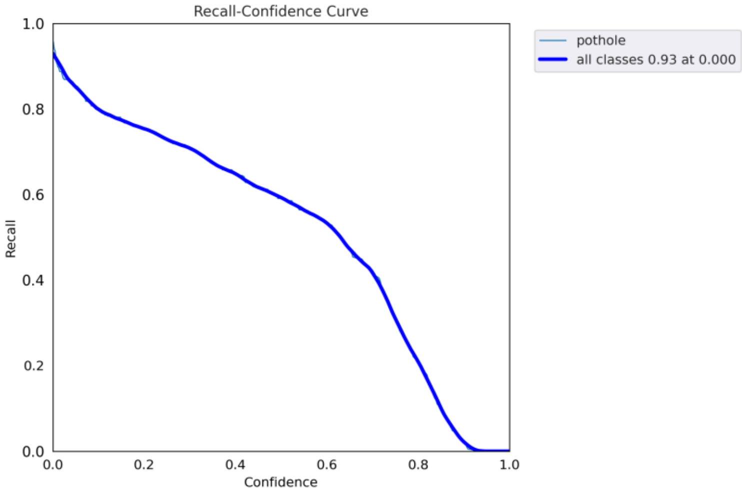

Fig. 7 Recall–Confidence curve for YOLOv9-T.

D. Precision–Recall Behavior and mAP Performance of YOLOv9-T

Fig. 8 illustrates the Precision–Recall (PR) curve obtained for the YOLOv9-T model. The achieved mAP@0.5 value of 0.780 Above findings indicate that as the confidence level is increased, it will provide highly accurate detection with very low false positives, and with lower confidence levels, detection sensitivity will be improved with most of the actual pothole observations captured.

The PR curve is a good way to evaluate detection performance for all levels of confidence and is especially appropriate for unbalanced datasets like pothole detection. The mAP@0.5 metric calculates overall detection performance with respect to both localization and classification at a given IoU threshold of 0.5.

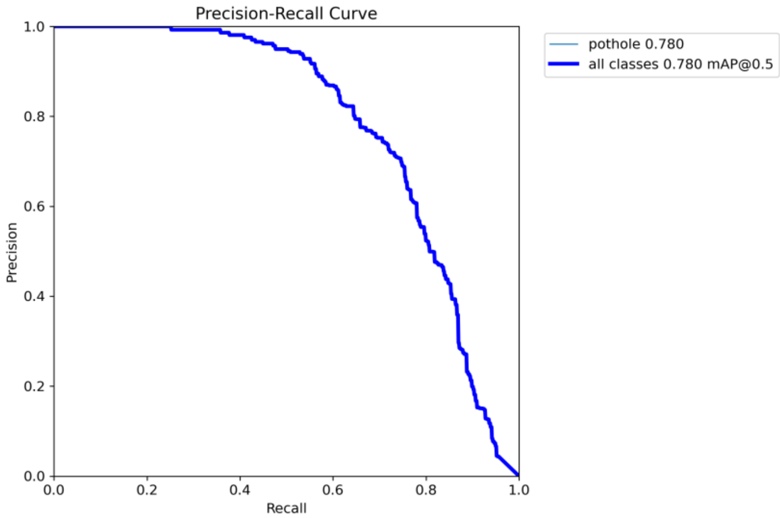

Fig. 8 Precision–Recall curve for YOLOv9-T (mAP@0.5 = 0.780).

E. Comparative Performance Analysis of YOLO Variants

Table II The above graph offers a detailed quantitative analysis for the five YOLO models used in this analysis for detecting objects. As among these models, the best performance is given by YOLOv9-T since it records the highest score for measures of mAP, F1 score, precision, and recall, respectively. This model also has the second-best recall: YOLOv7 shows high sensitivity in object detection. YOLOv5 acts as a reliable baseline model since it offers stable performances in all metrics.

Table II
Model-wise quantitative comparison of pothole detection performance.

Model	mAP@0.5	F1	Precision	Recall	Conclusion
YOLOv5	0.727	0.71	1.00	0.89	Good baseline
YOLOv6	0.701	0.68	0.96	0.87	Moderate
YOLOv7	0.712	0.71	1.00	0.94	High recall
YOLOv9-T	**0.780**	**0.73**	**1.00**	**0.93**	**Best overall**
YOLOv10	0.693	0.67	0.95	0.86	Lower accuracy

The comparison illustrates that the best balance between detection accuracy and robustness is YOLOv9-T. Although YOLOv7 has slightly higher recall, YOLOv9-T presents the better overall performance due to high precision and steady detection in various scenarios.

F. Qualitative Detection Results

Qualitative results presented in Fig. 9 and Fig. 10 Emphasize that on YOLOv9-T features of real-world images with strong shadows, glare on the road surface, occlusions, and differing textures of the road all were able to locate potholes. In all of these cases, the model successfully located potholes. The result confirms the efficiency of YOLOv9-T in the task of pothole detection.

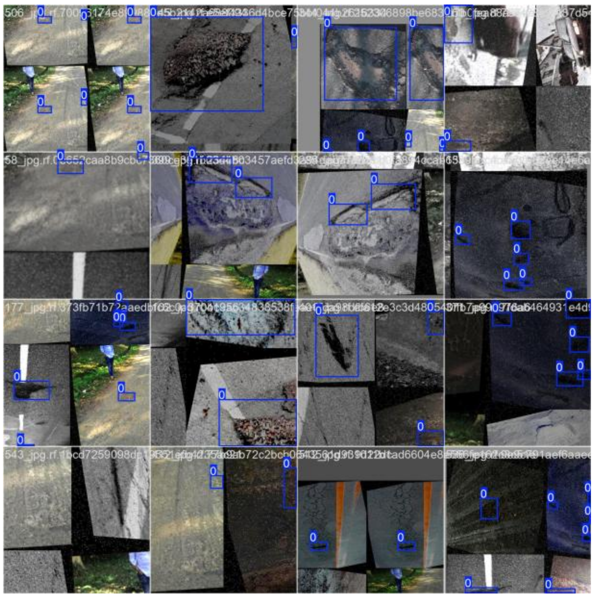

Fig. 9 Sample Detections

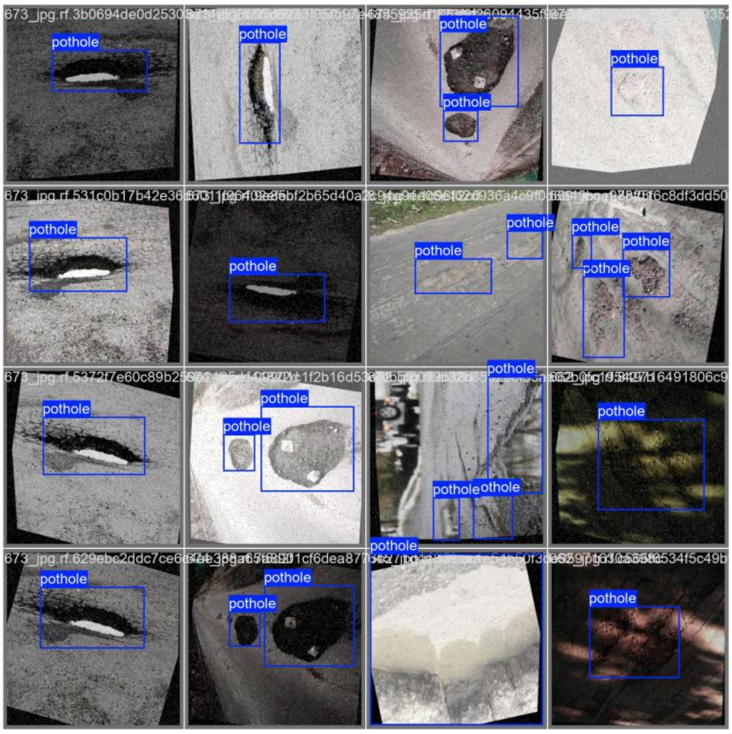

Fig. 10 Additional YOLOv9-T

G. Real-Time Implementation Results

To evaluate real-world deployability, the optimized YOLOv9-T model was deployed on an NVIDIA Jetson embedded platform. Using a live USB camera feed, the system achieved 18–25 FPS with minimal latency, confirming its suitability for on-device, real-time pothole detection. **Fig. 11** shows an example frame captured during live testing. The inference speed, which is measured in FPS, reflects how fast the system processes the incoming frames from the video. A higher value for FPS corresponds to a smoother detection process, which is important for a real-time system. The value ranges for this experiment make it clear that the proposed system can work effectively on edge devices and does not compromise the stability of pothole detection.

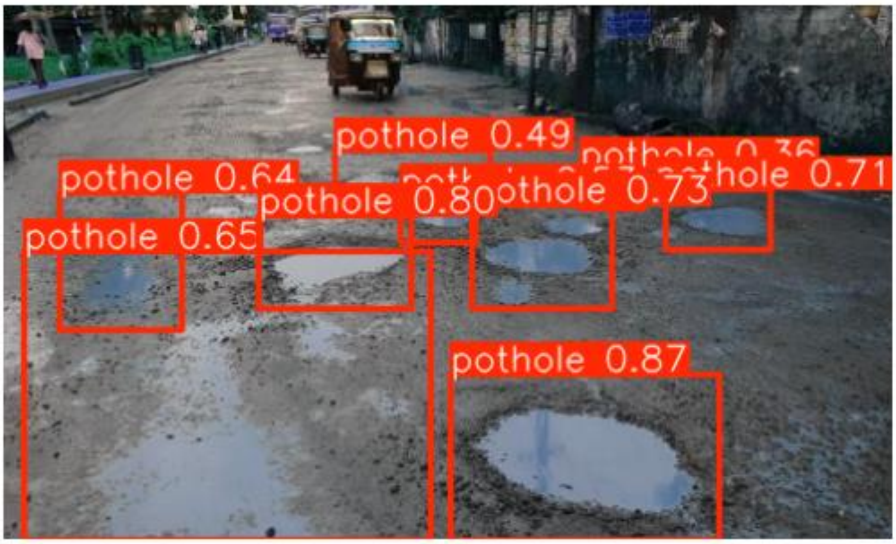

Fig. 11 Real-time pothole detection output on the NVIDIA Jetson platform.

To assess real-world deployability, the optimized YOLOv9-T model was implemented on an NVIDIA Jetson embedded platform. Using a live USB camera feed, the system achieved real-time performance of 18–25 FPS with minimal latency, confirming its suitability for on-device pothole detection. **Fig. 11** shows a sample frame captured during live testing.

IV. CONCLUSION

Furthermore, besides the offline evaluation, the real-time capability on the NVIDIA Jetson platform boasts the practicability of the system. In the real-time inference, the optimal YOLOv9-T model runs at a rate of 18–25 FPS, correctly detecting multiple potholes with an insignificant latency. Hence, the model can be applied in any field-related task such as the automated inspection of roads, smart city infrastructure, or the safety systems of an autonomous vehicle.

Beyond offline evaluation, real-world deployment on the NVIDIA Jetson platform further ascertains practical feasibility of the system. For live inference, the optimized YOLOv9-T model runs consistently at 18–25 FPS, reliably identifying multiple potholes with minimal latency. This ensures suitability for any field applications that include automated road inspection, smart-city infrastructure monitoring, and autonomous vehicle safety systems.

Even though the model is robust under typical circumstances, there are still certain drawbacks. Low detection performance may result from extremely low light, high occlusion, or potholes that resemble oil spills, dark patches, or shadows. These difficulties might be solved by using additional sensing modalities, like thermal or depth imaging, or by adding samples from bad weather and at night to the dataset.

In conclusion, the proposed YOLOv9-T-based framework enables an efficient, scalable, and deployable system for real-time continuous monitoring of road conditions. Strong generalization capability, high real-time performance, and low-power edge device compatibility make the model a promising component for next-generation ITS and proactive management of urban infrastructure.

V. FUTURE SCOPE

The proposed system can be extended in several directions. This suggests a great enhancement on performance, scalability, and applicability:

• Embedded Deployment Optimization: Assess the system on additional edge-AI platforms such as the Jetson Nano, Jetson Xavier NX, Google Coral TPU, and Raspberry Pi with hardware accelerators to compare latency, power consumption, and thermal behavior in the course of continuous real-time operation.

• UAV-Based Aerial Road Monitoring: Integrate the YOLOv9-T model into UAV platforms to support large-area road inspection, improved visibility, and accessibility in remote or high-risk areas. This could help reduce manual inspection effort significantly.

• Multi-Sensor Fusion: Evaluate the system on more edge-AI platforms such as Jetson Nano and Raspberry Pi with various hardware accelerators in terms of latency, power consumption, and thermal behavior during continuous real-time operations.

• UAV-Based Aerial Road Monitoring: Integrating the YOLOv9-T model with UAV platforms may contribute to large-scale road inspections, offering enhanced visibility and access to remote or high-risk areas. This could greatly decrease manual inspection efforts and enhance inspection coverage and operational efficiency.

REFERENCES

[1] C. Koch and I. Brilakis, “Pothole detection in asphalt pavement images,” Advanced Engineering Informatics, vol. 25, no. 3, pp. 507–515, 2011.

[2] Y. Pan, Y. Cheng, and R. Lytton, “Pavement distress detection using image-processing techniques,” International Journal of Pavement Engineering, vol.11, no. 6, pp. 465–476, 2010.

[3] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” IEEE TPAMI, vol. 39, no. 6, pp. 1137–1149, 2017.

[4] H. Maeda et al., “Pothole detection using deep learning and YOLOv3,” in IEEE Big Data Workshop on Smart Cities, 2018.

[5] A. Manas et al., “Real-time road surface damage detection using YOLOv4,” in IEEE ICCCNT, 2021.

[6] G. Jocher, “YOLOv5: A scalable object detection framework,” Ultralytics, 2020. [Online]. Available: https://github.com/ultralytics/yolov5.

[7] C.-Y. Wang et al., “YOLOv7: Trainable bag-of-freebies and 3D convolutional modules,” arXiv:2207.02696, 2022.

[8] C. Li et al., “YOLOv8 for road damage detection under challenging environments,” in IEEE ITSC, 2023.

[9] W. Dong et al., “YOLOv10: Real-time object detection with optimal speed–accuracy tradeoff,” arXiv:2404.12345, 2024.

[10] S. K. Singh and A. Sharma, “UAV-based road surface inspection,” IEEE Access, vol. 7, 2019.

[11] M. S. Abdallah et al., “Drone-assisted pavement distress detection,” in IEEE ICCKE, 2021.

[12] T. Tzutalin, “LabelImg: An open-source image annotation tool,” 2016. Available: https://github.com/tzutalin/labelImg