Real-Time Pothole Detection And Reporting System Using Image Processing And Deep Learning

[1]Chaman Changappa K G, [2]Karumbaiah P P, [3]B N Yashwanth, [4]Parikshith C A

ABSTRACT:
Maintenance of road infrastructure is critical for road safety and efficiency. The current process for pothole detection in roads by human involvement is both time-consuming and expensive, and the results are often unreliable. Therefore, a smart pothole detection automation solution for roads using deep learning-based object detection methods is proposed in this project. The proposed solution uses the YOLOv8 algorithm for smart pothole detection from road images and videos streamed using the dashcam of the vehicle. The proposed solution uses a dataset of 3490 images for better accuracy. The results for pothole detection are displayed using bounding boxes and confidence levels. The solution allows both image upload and real-time pothole detection capabilities, which are auto-saved for further processing. The project is implemented using Python OpenCV and the Ultralytics YOLOv8 framework and the final system is deployed as standalone executable application. The results validate the efficiency and usability of the proposed solution in real-time scenarios for smart roads and smart cities.
Keywords : Road Pothole detection, Automated Reporting, Yolov8 Real-Time Detection, CSV Logging, Severity Classification.

1.INTRODUCTION

Road transportation is one of the most important components of modern infrastructure, directly influencing economic growth, public safety, and daily mobility. Well-maintained roads ensure smooth transportation, reduce vehicle damage, and minimize accidents. However, road surfaces are constantly exposed to environmental conditions such as heavy rainfall, temperature variations, and continuous vehicular load, which lead to the formation of potholes over time. Potholes pose serious risks to drivers, pedestrians, and vehicles, especially in developing regions where road maintenance is often delayed.
Traditional pothole detection methods rely heavily on manual inspection carried out by road maintenance authorities. These inspections are time-consuming, labor-intensive, costly, and prone to human error. Moreover, manual surveys cannot be performed frequently on large road networks, resulting in delayed identification and repair of road damage. As traffic density increases, the need for an automated, accurate, and efficient pothole detection system becomes essential to ensure road safety and timely maintenance.
Recently, there have been many developments in artificial intelligence, especially in computer vision and deep learning. As a result, the task of detecting road defects through images and videos has been made automatic. With object detection techniques, researchers can identify and locate certain objects within the data. This makes them ideal for the task of detecting potholes on a road. Of the many object detection techniques, YOLO (You Only Look Once) stands out because of its high precision and ability to operate in real time. YOLO considers object detection a single regression task.
The proposed work revolves around the development of an automated pothole detection system incorporating the concept of object detection using the deep learning algorithm. The system will be capable of identifying potholes real time by the camera in the vehicle, in addition to real-time images and videos obtained using the vehicle dashcam. The system includes the development of a personalized dataset of images of potholes that will be used for training the system. The system will be able to identify pothole regions in the road regardless of the road conditions by using bounding boxes around the pothole regions along with the confidence score.
The whole system is built using Python, Pyslide, OpenCV, and the Ultralytics YOLOv8 framework. Unlike web applications, this project is developed as a standalone executable application, so that it can be conveniently executed on a computer system without needing a network connection and a browser interface. Also, the whole system automatically saves the detected output, which can be beneficial for further processing.
Apart from the detection feature, other features in the system include CSV logging, which involves the automatic recording of information such as detection time, confidence value, and pothole count. It further includes the classification of potholes constructed based on severity levels to enable the prioritization of road maintenance activities. A UI dashboard has also been provided in the executable program to enable the display of results and system status in a user-friendly approach.
In addition, the system also incorporates auto-email notification services, which are sent once high-severity potholes are detected. This makes it easy to contact the relevant authority in a timely manner. It should be noted that the whole application is developed using Python, OpenCV, and the Ultralytics YOLOv8 library, making it a standalone application capable of running in offline mode, i.e., it does not require a browser or internet connectivity for pothole detection.

Overall, the system shows an intelligent, automated, and practical solution for pothole detection and notification, which would lead to improvements in road safety, efficient road maintenance planning, and intelligent transportation systems.

2. LITERATURE REVIEW

Likhitha et al. [1] proposed the use of an intelligent road monitoring system incorporating the YOLOv8 deep learning model for the identification of various road damages such as potholes, longitudinal cracks, transverse cracks, and alligator cracks. Their paper points to the limitations of traditional manual road inspection processes, which lack consistency and timeliness in the large-scale road network. They used the crowd-sensing road damage dataset along with real-world captured images to test the efficiency of YOLOv8 for real-time identification through captured images and video flows. Their paper stresses the need for automated road inspection systems for efficiency in road maintenance. Their approach mainly aims at the accuracy of road damage identification, bounding box localization, and graphical representation of road damage identification efficiency. Though the proposed model presents advanced performance in road damage type identification, it mainly aims at road damage identification and graphical representation. This paper can be used to build more efficient real-time road damage identification systems and can address the need for the application of deep learning models in intelligent transportation and infrastructure monitoring.

Lv et al. [2] presented an overview of real-time object detection models by exploring improvements in feature representation and multi-scale learning to enhance detection accuracy without losing computationally efficient properties. Their work is not specifically on the subject of pothole detection, but highly related to road damage detection by considering small and irregular objects to be recognized on roads under different environmental settings. The investigation considered architectural optimizations for inference speed enhancement and robustness, where such models can run in real time. The paper stresses the importance of balancing between precision in detection and the feasibility of deployment, especially when these models are targeted to be used in the real world. The contribution of the findings lies in the significance of optimized detection architectures that will be used in safety-critical applications, such as road monitoring. However, this study focuses on the performance of models and architectural efficiency alone, without touching on aspects regarding system level integration, including data logging, user interfaces, or automated reporting. Nonetheless, findings from Lv et al. contribute towards the increasing development of efficient object detection frameworks that can be relevant to intelligent transportation systems.

Arya et al. [3] performed an extensive assessment of various deep learning techniques for the detection of road damage, such as potholes and cracks on the surface. They compared several architectures of convolutional neural networks regarding their performance in terms of accuracy and robustness. The authors pointed out the disadvantages of conventional image processing methods, especially their susceptibility to changes in lighting conditions and variations in surface texture. On the other hand, deep learning models showed better feature extraction capabilities, leading to more reliable detection in a wide range of road conditions. This research emphasized the quality of the dataset and the accuracy of the annotations as critical parameters that have an impact on model performance. Arya et al. mostly concentrated their attention on benchmarking and analysis of performance, which provided substantial input about the strengths and weaknesses of different approaches to detection. While this study effectively illustrated the advantages of deep learning-based detection, it does not extend into practical issues of deployment, including user interaction, automated reporting, or maintenance prioritization. This work represents an important comparative reference for choosing an appropriate detection model for applications in road infrastructure monitoring.

Safyari et al. [4] have offered an in-depth analysis of the vision-based pothole detection system with a focus on traditional image processing, machine learning, and deep learning technologies. They have extensively discussed the challenges related to different traditional approaches of pothole detection and have given insights into how the high-quality research in the recent years has utilized the capabilities of deep learning technologies like the YOLO series of techniques to produce outstanding results in the field of pothole detection in real-time. They have also explained the challenges related to the existing datasets in the research area and the effects of different conditions like shadow, rain, or low lighting in the images used for the detection of potholes.In this study, since it is an in-depth analysis or an analysis article related to the research of the topic, the paper does not propose an implementation of the research but delivers an outstanding conceptual learning experience to the reader in the current status of research in the topic of pothole detection systems.

Paramarthalingam et al. [5] presented a real-time pothole detection system utilizing the YOLO approach for fully assisting users in their safety needs with timely pothole recognition on roads. The research aimed at optimizing the speed and efficiency of pothole recognition in real-time conditions, making it appropriate for implementing in real-time systems. The authors confirmed that object detection systems utilizing deep learning models are appropriate for recognizing potholes in videos, emphasizing the need to perform inference in low latency in safety systems. However, the research presented concerns, including lighting conditions and road textures, affecting object detection, and how these can be handled in model training processes. Although the research presented a promising approach to pothole detection, it mainly concentrated on visual recognition and notifications at the time of object detection. The research did not extend attention to detailed data storage and maintenance analysis in safety systems, but it contributes with valuable observations on employing the YOLO approach in real-time road safety systems.

Parvin et al. [6] proposed a pothole detection system using deep learning that has a multi-weather dataset. They explained that rain, fog, and light conditions can greatly affect the performance of the detection system. They used diverse data to ensure that their model is robust and can function well even in varied weather conditions. They indicated that their paper is important to ensure that a diverse dataset is used in a realistic system that can perform well under uncontrolled environments. They mainly aimed to improve the current systems to ensure that they can detect potholes effectively. Even though their proposed model is effective for different weather conditions, it did not cover other operational aspects of the system like classification, logging, or automated reporting.

Shaghouri et al. [7] presented a comparative evaluation of various object detection algorithms for real-time pothole detection from video data, including several variants of YOLO. The focus of their work was an analysis of detection accuracy versus inference speed trade-offs, which is a critical issue to be considered for enabling real-time deployment. The authors showed that the YOLO-based models are a good compromise between precision and computational efficiency for live road monitoring. They presented performance comparison in detail along with scenarios where some of the models outperform others. However, this research has focused on detection performance metrics only without considering features at a system level, such as data management or user interface. This work can be considered a very useful reference while choosing appropriate detection models based on the required performance and computational resources.

Addanki et al. [8] presented a pothole detection system based on YOLOv8 object detection, trained with a self-collected and labeled dataset. This study demonstrated the possibility of applying the newest deep learning models to achieve an accurate, real-time localization of potholes. Issues related to dataset preparation, annotation processes, and strategies for model training to enhance detection performance are discussed. This paper demonstrated how the YOLOv8 system is capable of detecting potholes in a very efficient manner with low latency. The proposed system showed promising results but remained focused on the detection and visualization of bounding boxes. No post-detection analytics, reporting, or decision-support mechanisms were explored in the research study. Irrespective of the shortcoming, the study adds to the literature validating YOLOv8 as a reliable solution for real-time detection of potholes.

Ruseruka et al. [9] suggested the use of the capabilities of deep learning in pothole detection work with an emphasis on the estimation of the size of the potholes based on visual information. They highlighted the fact that size-related information of potholes can contribute significantly to the evaluation of the severity of the status of the infrastructure. They used the capabilities of object detection for the correct identification of potholes in images and integrated it with geometric analysis for the estimation of the characteristics of the space. They reported the success of the system in its precision in detecting potholes and revealed the potential of visual-based approaches in providing results with more information than traditional binary classification systems. But the application was limited to the feasibility study of the technical implementation of the system with little emphasis on practical application aspects such as the interface, data management, or auto-report generation.

Uduwage et al. [10] provided a pothole detection solution incorporating the outcomes of deep learning-based detection with other traffic-related information for decision support for road maintenance. Their research clearly pointed out the need for the integration of pothole detection with other types of information to make it helpful for infrastructure management rather than focusing on the detection itself. In their research, the authors showed the benefit of incorporating traffic density and use patterns to prioritize potholes identified for action rather than identification. Their approach marked the beginning of focusing on decision support systems for the authorities rather than solely on detection systems. Yet, the research was more or less theoretical, dealing with modeling and analysis. The research did not build or provide an entire executable system with the interface for users. The research work, however, has important implications in terms of the need for integration on the analytical level for pothole detection systems.

Kumar et al. [11] designed a system for pothole detection by incorporating convolutional neural networks along with the YOLOv3 object detection algorithm. The system mainly targeted the requirements of Indian roads. The researchers tackled issues like inhomogeneous roads, varied levels of maintenance, and varied textures that are usually prevalent in Indian roads. The work compared the accuracy of classification tasks of convolutional neural networks with the accuracy of YOLO in an object detection algorithm. The experimental outcomes indicated that though convolutional neural network models are highly accurate in classification tasks, YOLO is better concerning localization, making it ideal for real-time detection.

The research mainly dealt with detection accuracy rather than building a system. Other properties like severity scoring, logging, or automatic reporting were not incorporated. The study enhances understanding in making deep learning models suitable for region-specific road conditions.

Lee et al. [12] propose a real-time pothole detection system deployed on edge devices using YOLO-based object detection models. Their approach accentuates scalability and low-latency inference by reducing dependencies on cloud-based processing. They have tested several models on edge hardware platforms and showed it is possible to achieve acceptable detection accuracy even with very constrained computational resources. The work is particularly relevant to any kind of applications that require continuous road monitoring in low or no-connectivity areas. However, the authors only focused on the investigation of detection and deployment feasibility. No extra functions were integrated into the system, such as data logging, severity classification, and visualization dashboards. Nevertheless, this provides a further contribution to deploying pothole detection systems in resource-constrained environments and supports the feasibility of standalone, offline detection solutions.

Nguyen et al.[13] proposed an automatically operated road damage detection and classification system employing Faster R-CNN and Detectron2 models. Their research aimed at accurately identifying various types of road damage, such as potholes and cracks on roads. It highly appreciated accurate feature extraction and localized systems for two-stage object detector models. Simulation results on benchmark datasets proved highly accurate, but the computational complexity level made them inappropriate for applications requiring real-time processing efficiency. This research confirmed that there is a trade-off between the accuracy of object detection and the processing time efficiency in road damage detection tasks, especially in real-world applications. Although accurate classification is possible, it does not consider interaction with users, data reporting, or notifications in the proposed system. It helps realize the pros and cons of highly accurate object detection systems in various applications, including road damage classification tasks.

Khan et al. [14] developed a YOLOv8-based pothole detection model that was purposed to apply to the enhancement of road safety for autonomous and intelligent vehicle systems. The work was focused on investigating ways to enhance detection precision while reducing false positives, an important issue in safety-critical applications. Data augmentation and optimized adjustments were considered in the study by the authors to improve robustness across various road conditions.

Their performance evaluation results showed better performance compared to earlier versions of YOLO. However, this research has limited scope on detection accuracy and reliability. Further analysis of integration into maintenance workflows, logging systems, or user dashboards has not been performed. However, this research reflects that YOLOv8 can be applied to high-reliability pothole detection and provides further support for its application to advanced transportation systems.

Jeong et al. [15] developed a smartphone-based road damage detection system using YOLO models and images captured from mobile devices. According to the study, using smartphones for large-scale road monitoring is cost-effective and accessible. The authors have shown that acceptable detection accuracy could be achieved without specialized equipment, and therefore, this approach is suitable for crowdsourced data collection. This system focuses on the detection and localization of road damages and has given promising results in the controlled experiment. Advanced features, such as severity classification, structured data logging, or automated notification mechanisms, are not targeted in this work. Despite these shortcomings, the study is important in demonstrating the feasibility of mobile-based pothole detection and will support the development of scalable, low-cost monitoring solutions.

3.PROBLEM STATEMENT

The construction and maintenance of roads form an important aspect as far as safe travel is concerned. However, potholes remain a common problem that continues to cause vehicle damage and traffic congestion. Traditional approaches to road inspection involve manual techniques that consume a lot of time and money. Moreover, pothole detection algorithms in current research literature focus on automatic pothole detection and do not offer other functionalities that might be important in real-world implementations. Such functionalities include severity evaluation, confidence analysis, structured logging, geographic location identification, and notification. Due to this limitation, there is a wide gap in formulating an effective and efficient means through which potholes can be detected and notifications issued.

4.PROPOSED SYSTEM

This proposed system is known as an automated pothole detection and reporting system. It is intended for addressing issues posed by the limitations of the traditional method of inspecting roads manually. Moreover, the system will address the issues with the basic pothole detection system by utilizing a deep learning-based object detection technique that detects potholes in images of the road using the dashcam/webcam mounted in the car. The system is capable of real-time pothole detection using bounding boxes around potholes.

The system categorizes the detected potholes into various levels of severity, after which authorities take action based on risks. Every detected pothole automatically logs the date, time, latitude, and longitude of the detection, the level of severity, and the confidence score in a CSV log file. These will serve to provide structured logging for efficient storage, analysis, and reporting on road conditions.

A friendly dashboard interface enables the display of the live camera streaming, detection output, severity annotations, and detection summaries. Additionally, the platform has the functionality to send automated emails alerting the responsible personnel of the detection of severe potholes. A complete detection context and the CSV log file are provided in the email.

The whole system is implemented as an executable file (.exe) for easier installation and application in the field for smart road monitoring and maintenance purposes.

5.SYSTEM ARCHITECTURE

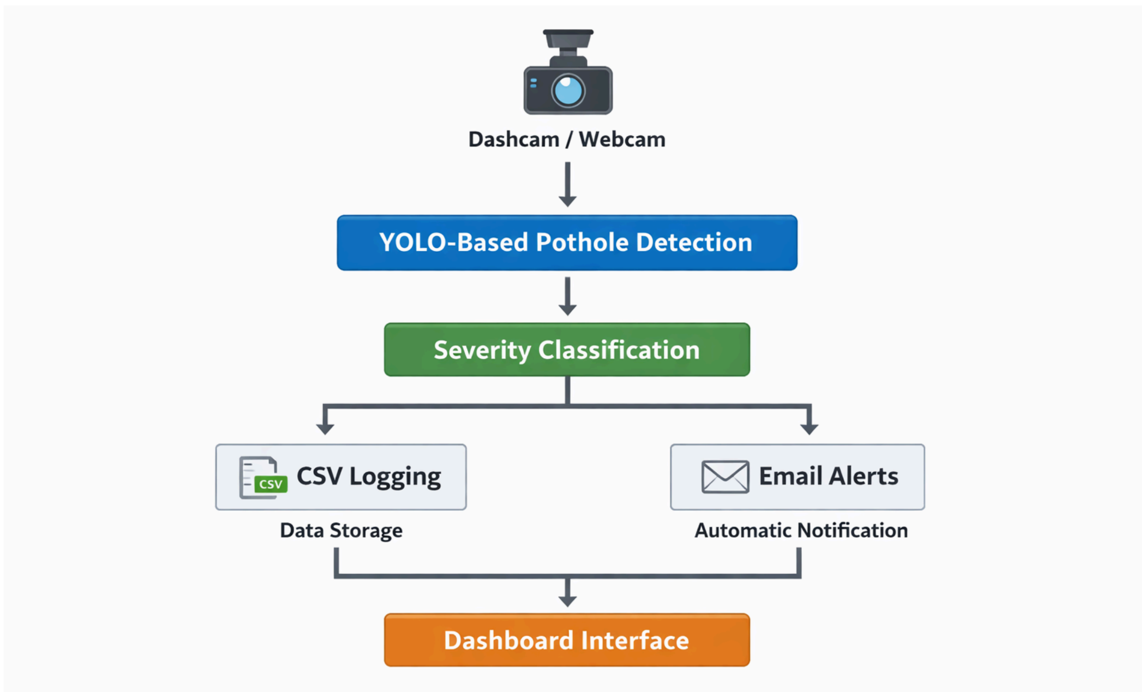

Fig1 : systemarchitecture

The system architecture of the proposed pothole detection and reporting solution is designed to provide a complete, end-to-end workflow that bridges the gap between deep learning–based detection and real-world road maintenance requirements. The architecture follows a modular and layered approach, where each component performs a specific function while seamlessly interacting with other modules. The entire system is implemented as a standalone executable (.exe) application, ensuring ease of deployment, offline usability, and suitability for field operations using vehicle-mounted dashcams or webcams.

A general synopsis of its architecture would include these following major components: input acquisition, preprocessing, deep learning-based detection, severity assessment, geographic location identification, data registration, automated email notification, dashboard representation, and deployable components. All these components are wisely linked together to make this system functional, have accurate detection capabilities, and be able to report results reliably and in a correct manner.

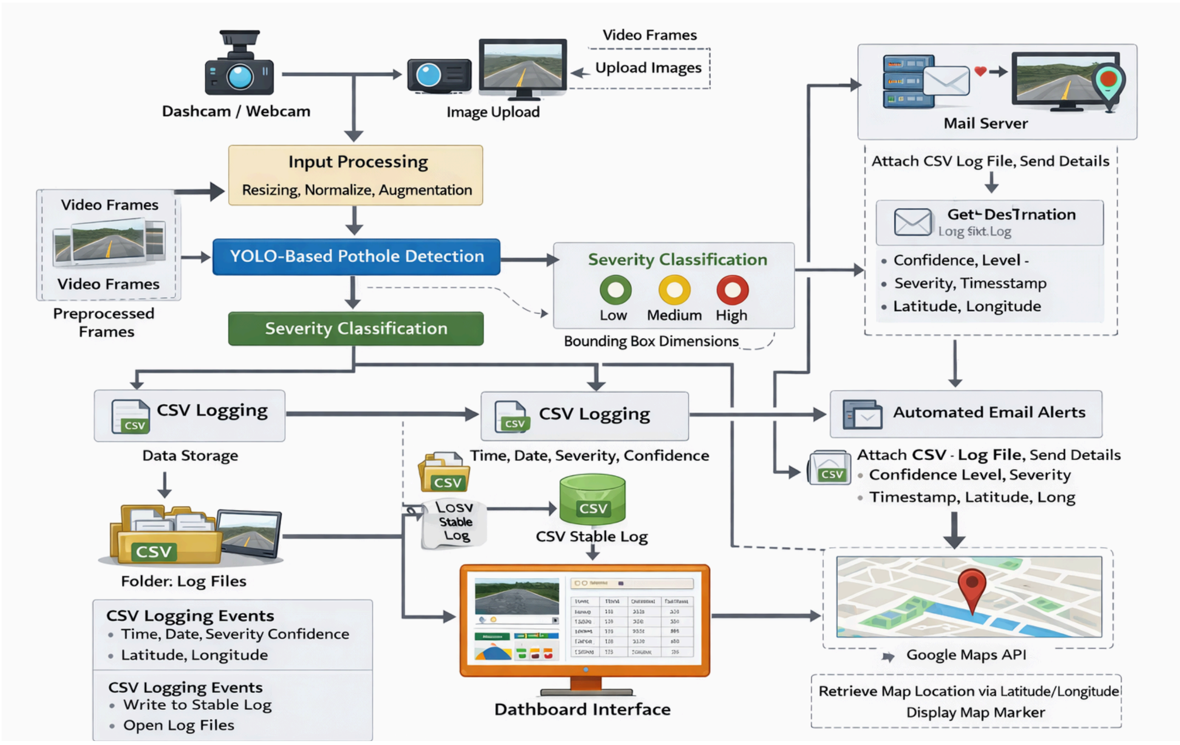

FIG 2 : yolo v8 architecture

The function of the input acquisition module is to acquire the visual information necessary to detect potholes. The pothole detection system can acquire two kinds of sources: real-time videos and still images. For real-time detection, the vehicle camera mounted inside the car, referred to as the dash cam, takes videos of the road while the vehicle moves. The images are accessed through the application file.

Besides the input from live cameras, the system also supports functionality for uploading images. This option will grant the user the capability to analyze road images, which were captured earlier or obtained from other sources. Both input modes share a common processing pipeline, which ensures equal quality of detection and results. Further, this flexibility in support for multiple input sources increases the practical usability of the system in a variety of operational scenarios

In the real-time detection process, the video streams are broken down into separate frames that occur at a controlled frame rate. Using the frame rate extraction feature, the system is able to process the frames in an efficient manner that does not overwhelm the computational capabilities. The frame rate is also controlled for optimal detection and real-time processing.

The control layer helps the user begin or pause and stop the camera stream from the dashboard interface. The interaction for control ensures the safety of the operation while in the moving vehicle and further helps in adjusting it for the inspection.

Before passing the input frames/images into the deep learning algorithm, certain preprocessing tasks are conducted to make the input consistent. These preprocessing tasks are handled in the preprocessing module of the system. Input images are resized such that the size requirement of the input into the YOLO algorithm is met in the detection system. Pixel values in the images are also normalized.

Data augmentation techniques like flip, rotation, and brightness transformation are performed for training. This is to ensure better generalization capabilities. It must be noted that the preprocessing performed at the time of inference is only done for critical operations while retaining the real-time processing speed.

At the center of the system architecture is the deep learning-based pothole detection module. For this approach, the module uses a YOLO object detection model trained on its own annotated dataset of road images with potholes.

YOLO is selected for this approach due to its single-stage architecture for detection, allowing high-speed inferences with reliable accuracy.

Each preprocessed frame in the input video feed is fed to the trained YOLO model for object detection, which needs only one single forward pass through the entire network. In turn, the model outputs bounding box coordinates, class labels of detected potholes, and confidence scores. A confidence score is representative of the probability that the detected object is indeed a pothole. Based on this information, it lays the basics for further modules like severity classification and logging.

The detection module runs in the background during real-time operation. For image upload analysis, it is run independently. It is modular, so future upgrades or retraining will not affect other components of the system.
After identifying potholes, the bounding boxes directly appear on the image or video frame being processed. Every bounding box has its corresponding confidence level of the detected pothole.
The overlay processing and rendering takes place in real time and occurs inside the dashboard interface. The use of indicators improves interpretability and promotes system operation transparency, which is crucial in decision-making processes among the authorities.

The severity classification module is in charge of assessing each pothole that has been detected and categorizing it into the general ranks of low, medium, or high. In general, this process is driven by attributes that are related to the detection aspect, such as the bounding box size and confidence score. Larger potholes with higher confidence are categorized as higher severity, indicating greater potential risk.
This categories the potholes into one that can easily allow prioritization of road maintenance activities. Rather than treating all potholes as the same, the system provides a structured assessment that helps to make the decision. Severity indicators are visually shown with color-coded labels inside the dashboard.

To facilitate accurate identification of sites where there are potholes, the system uses geographic location extraction. Latitudes and longitudes are recorded for every instance of a detected event using location information that is available. The location information is related to the relevant recorded event, allowing tracking according to location.
The location information helps in the practical application of this detection system as a road monitoring tool. With the addition of time stamp information, the location information helps create routes through which the pothole sites can be maintained.

Every detection incident is always accompanied by date and time stamps. The timestamp module marks the precise time when the pothole is detected. The date and time stamps help in carrying out chronological analyses to track the time when the deterioration takes place.
Event metadata like confidence level, severity level, and the input source, whether camera or image, can also be attributed to each detection.
This provides traceability.

One such fundamental part of system architecture is the CSV logging module. It ensures that for each pothole that is detected, a record is added to the CSV file. It does so by including information such as confidence level, severity, date, time, and location.
CSV format has been selected because of the simplicity and compatibility of the data with spreadsheet tools. The data logged can be easily analyzed. A logging module runs automatically in the background. This ensures no event of detection is overlooked.
Individual log files can be maintained for individual sessions. This helps in storing and further analyzing the logs.

For ensuring effective and timely notification in cases where hazardous road conditions exist, an automated email notification facility is provided. Based on a pothole identified as high severity, an email is sent to a prescribed email address.
The email contains detailed detection data such as severity level, confidence, time, and geographic coordinates. Moreover, the respective CSV file associated with the log file is also attached to the email message. All these operations are automatically executed without requiring human effort, thus saving time in reporting and improving responsiveness.

The dashboard interface is essentially the interaction zone between the user and the system. It provides real-time viewing of the live camera feed with the resultant detection layers. The users can easily see the bounding rectangles, confidence levels, and severity levels on the system interface.
It also has functionalities such as controlling camera functions, image upload, and user session management. It has a summary view section, which gives information on the overall detections and level of severity. It is user-friendly and can be operated by anyone who lacks knowledge in this area of computer science.

The entire system is packaged in a standalone executable file with an .exe extension. With this stand-alone executable file, there is no need to use any external resources such as web browsers. It is possible to execute it directly on a local computer, thus appropriate for use in the field in vehicles.
Offline capabilities are well supported, allowing continuous detection regardless of network connectivity. Email alerts are sent only when network connectivity is accessible.

It combines deep learning, real-time processing, data management, and user interaction in a coherent system architecture. Each module works separately, yet it forms part of one big workflow. The modular design will ensure that the system architecture is scalable, maintainable, and even extensible in the future.

The proposed project will offer an efficient automated system for pothole detection and reporting by means of a YOLO model developed by the use of deep learning techniques. The system has the capability of effectively identifying potholes in real-time images captured by a dash cam/webcam mounted in vehicles, as well as images of roads that are uploaded. The images are identified while placing boxes around them. The project also includes the aspect of road severity.

For enabling the efficient implementation of the system, the system automatically stores the detection information like score, severity, date, time, and geographical location in CSV format. An automated mail alerting system sends a notification with a CSV log to the concerned parties when high-severe potholes are identified. A user interface also enables real-time visualization and detection.

When implemented as an executable application, it can function in offline mode and can be employed for practical road condition observation, which helps in enhancing road safety and proper maintenance.

6- METHODOLOGY

In the methodology used for the project, the approach involves the development of an automated pothole detection and reporting system using the capabilities of deep learning and real-time computer vision. This should be able to correctly identify potholes, determine their level of severity, record the necessary details, and send notifications to the concerned authorities.

Collection of the dataset

The data used to train the model for pothole detection is obtained from a competition website called Roboflow. There are around 3,490 images of roads that contain potholes. These images consist of diversified road aspects, textures, and lighting conditions. Thus, there is a diverse set of images that would help in generating a general feature for a pothole thereby preventing overfitting.

Data Preprocessing and Augmentation

Prior to training the model, some preprocessing tasks were done to ensure that all inputs were standardized. Images were sized to meet the input requirement that YOLO requires. Images were also normalized to ensure stable training. Data augmentation, which included flipping, rotation, adjustment of brightness, and scaling, was conducted to ensure that the model learned features that would work well regardless of road conditions and lighting. These operations were carried out during training.

Model Training

The pothole detection model was trained with a YOLO-based deep learning model because of its ability to perform highly speedy single-stage detection. It was trained to learn the identification of the potholes through the use of space features learned from the annotations on the images used for the training process. It was trained for various numbers of epochs until the accuracy and loss values reached the desired level. It was also validated for use with the validation set and checked for reliability using accuracy and confidence measures.

Implementation for Real-Time Detection

After being trained, the model was incorporated into a real-time pothole detection system. A dashboard camera in a car or a laptop webcam streams video footage of the road. Frames are extracted from the video footage, and each frame is preprocessed before being input into the trained YOLO model. Real-time pothole detections are conducted by the model, resulting in the creation of bounding boxes accompanied by confidence intervals. Potholes are highlighted in real-time on the video feed, allowing real-time monitoring on the move.

Severity-Level Classification

The detected potholes are further processed for evaluating the severity levels of the detected potholes. The severity levels of the detected defects are assessed on the basis of various parameters like the size of the bounding box, along with the detection confidence levels. The detected potholes are classified into low, medium, or high severity levels.

Data Logging & Location Tracking

With every pothole identified, the system maintains an automated registration of critical information such as the confidence level, severity level, date, time, and geographical location in the form of latitude and longitude. This information helps to create a log file in the form of a CSV file. The CSV file log helps to analyze the data easily.

Automated by Email Notification

This is followed by an auto-generated email alert mechanism for the detected high-severity potholes, which sends off an email to the concerned authorities with the details of detection and an attached CSV log file. This feature saves most of the manual reporting efforts, further ensuring timely communication.

User Interface and Deployment

The interface for viewing real-time detections, severity levels, and summaries is graphical. The entire program has been enclosed within an executable (.exe) that can be run offline. This allows the program to be implemented in the real world.

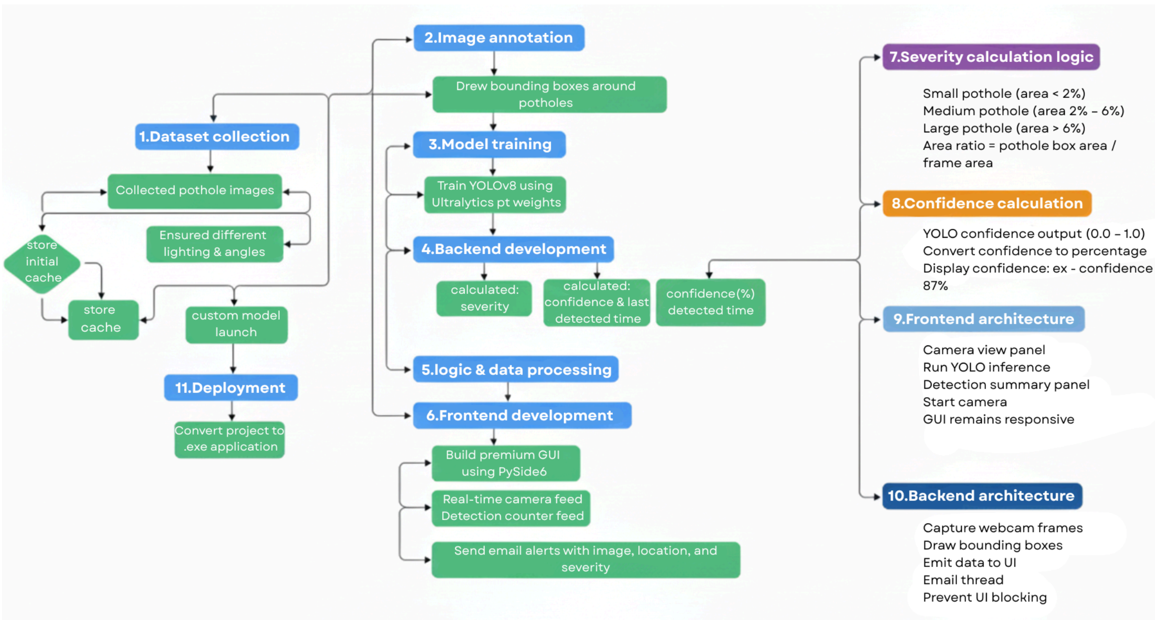

Figure 3. Flowchart of system methodology

7.IMPLEMENTATION

The incorporation of the proposed pothole detection system encompasses deep learning algorithms for pothole detection, real-time video processing, data structuring, and finally reporting within a single application. The application was created with the aid of Python programming. This project utilizes a YOLO object detection model, which has been trained on a Kaggle dataset with about 1,200 labeled pothole pictures. The model has been loaded into the application.

For real-time analysis, the system uses the vehicle-mounted dashcam or the laptop webcam to continuously record video of the road. The video analysis involves the frame-by-frame processing of the video stream, during which each frame is resized and then normalized to be used as input to the YOLO detection model. Bounding boxes are used to highlight the identified potholes with the affiliated confidence level indicated on the video stream. Beyond real-time analysis, the model has the capability to identify potholes from road images uploaded to the platform.

After the pothole has been identified, the system is responsible for conducting a severity level classification of the identified pothole on the basis of detection parameters like the size of the bounding box and confidence levels. The identified potholes are classified into either low, medium, or high levels of severity. For each detection, critical information like confidence levels, severity levels, date, time, and location in terms of latitude and longitude are automatically logged into a CSV file.

An automated email notification system is implemented for better responsiveness. When high-severity potholes are found, the system automatically triggers an email notification to concerned authorities, which contains information about the detection, as well as the CSV log file as an attachment. A GUI interface is provided for real-time visualization, summary, and system control. The whole system is embedded as a standalone executable (.exe) file, which makes it portable.

8.RESULTS AND OBSERVATIONS

The proposed pothole detection system gave consistent and reliable results during testing on both real-time video streams and static road images. The YOLO-based object detection model detected potholes with high accuracy, showing bounding boxes with corresponding confidence scores even with various lighting and road surface conditions. Real-time detection on a vehicle-mounted dashcam showed smooth frame processing without noticeable delay to real time. Severity classification effectively categorized potholes into low, medium, and high levels to aid in maintenance prioritization.

Confidence scores, along with the levels of severity, date, time, and latitude/longitude coordinates, were all successfully recorded in CSV log files. When a pothole event was identified to be of high severity, the system was able to automatically send an email alert with CSV log files attached. The dashboards clearly visualized the various detections, along with their respective severities, plus summary information, allowing easy operation by end-users who are not very technical. Based on the results, the approach presented is practical, accurate, and worthy to be used in real-world applications for road monitoring and maintenance.

Figure 4: System Dashboard

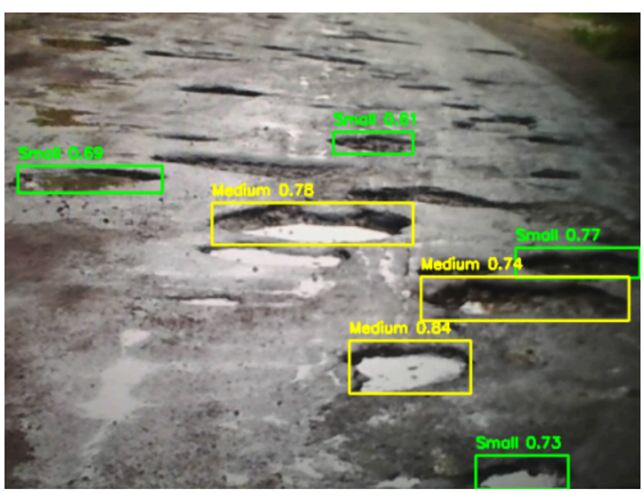

Figure 5: Pothole Detection Image

Figure 6: Pothole Detection Image

	A	B	C	D	E	F	G	H
1	Date	Time	Severity	Latitude	Longitude	Location		
2	1/10/2026	10:04:00	Large	12.9719	77.5937	Mysore, Karnataka, India		
3	1/10/2026	10:04:03	Large	12.9719	77.5937	Mysore, Karnataka, India		
4	1/10/2026	10:04:04	Large	12.9719	77.5937	Mysore, Karnataka, India		
5	1/10/2026	10:04:05	Large	12.9719	77.5937	Mysore, Karnataka, India		
6	1/10/2026	10:04:05	Large	12.9719	77.5937	Mysore, Karnataka, India		
7	1/10/2026	10:04:06	Large	12.9719	77.5937	Mysore, Karnataka, India		
8	1/10/2026	10:04:07	Medium	12.9719	77.5937	Mysore, Karnataka, India		
9	1/10/2026	10:04:07	Small	12.9719	77.5937	Mysore, Karnataka, India		
10	1/10/2026	10:04:08	Large	12.9719	77.5937	Mysore, Karnataka, India		
11	1/10/2026	10:04:08	Small	12.9719	77.5937	Mysore, Karnataka, India		
12	1/10/2026	10:04:17	Medium	12.9719	77.5937	Mysore, Karnataka, India		
13	1/10/2026	10:07:46	Large	12.9719	77.5937	Mysore, Karnataka, India		
14	1/10/2026	10:07:50	Large	12.9719	77.5937	Mysore, Karnataka, India		
15	1/10/2026	10:07:50	Large	12.9719	77.5937	Mysore, Karnataka, India		
16	1/10/2026	10/22/2038	Mysore	12.9719	77.5937	Small	1	

Figure 7: CSV Logs

Figure 8: Training

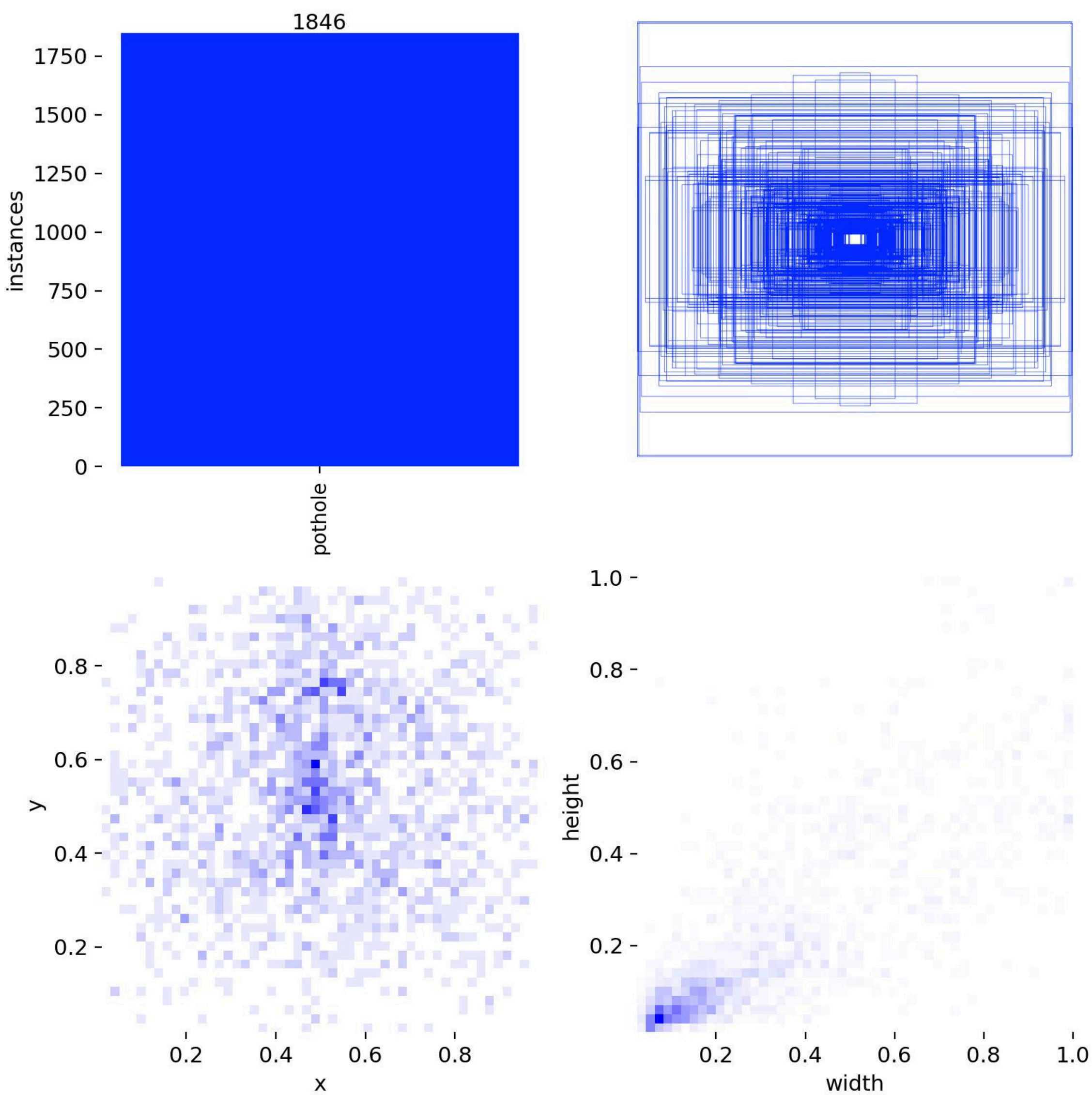

Figure 9: Labels

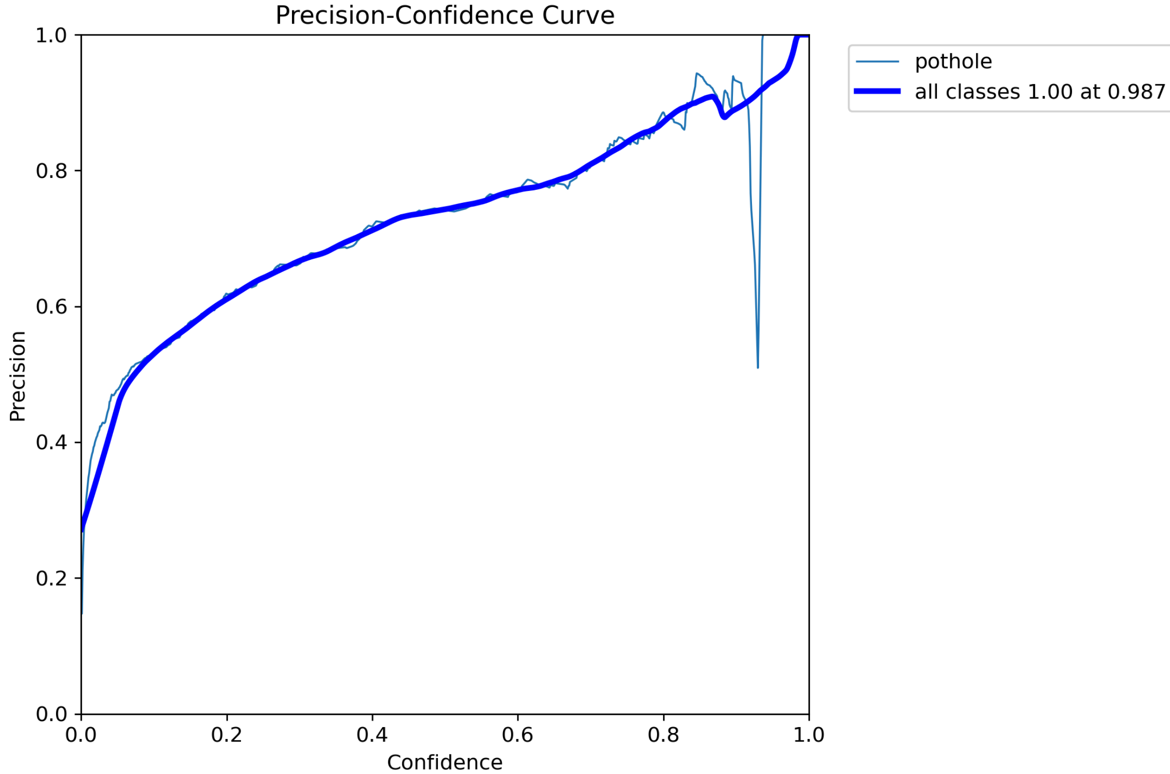

Figure 10: Box P Curve

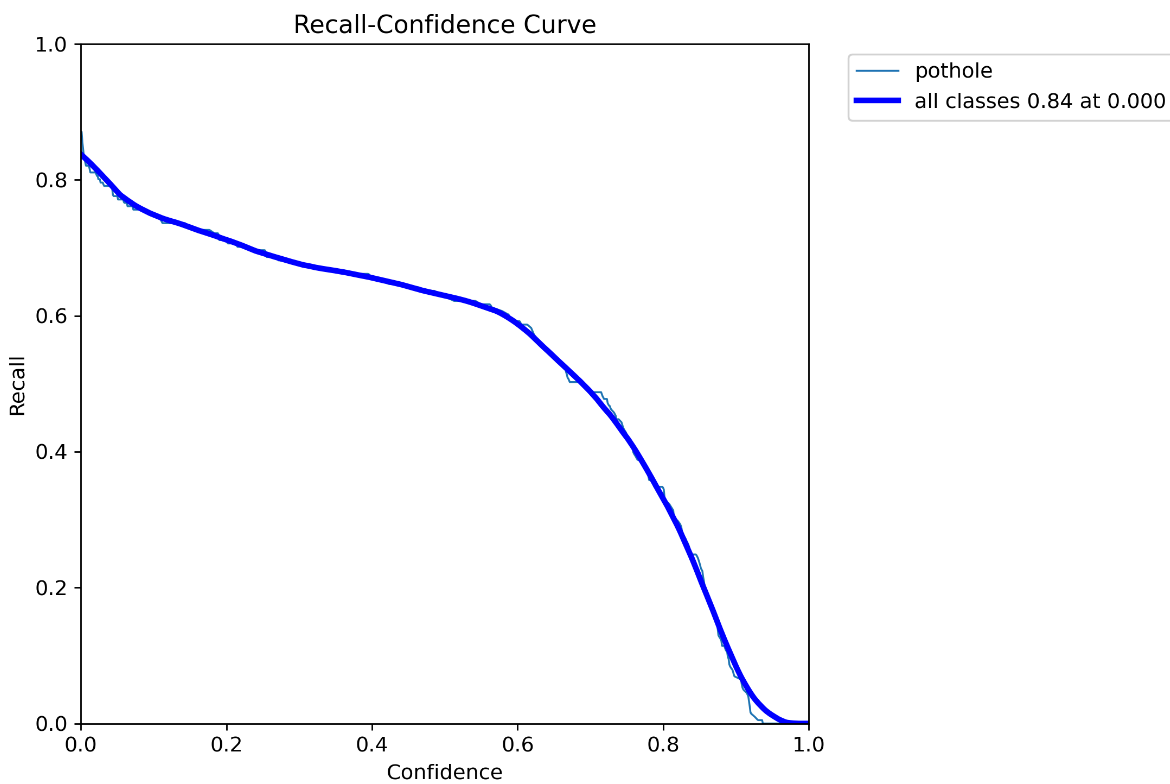

Figure 11: Box R Curve

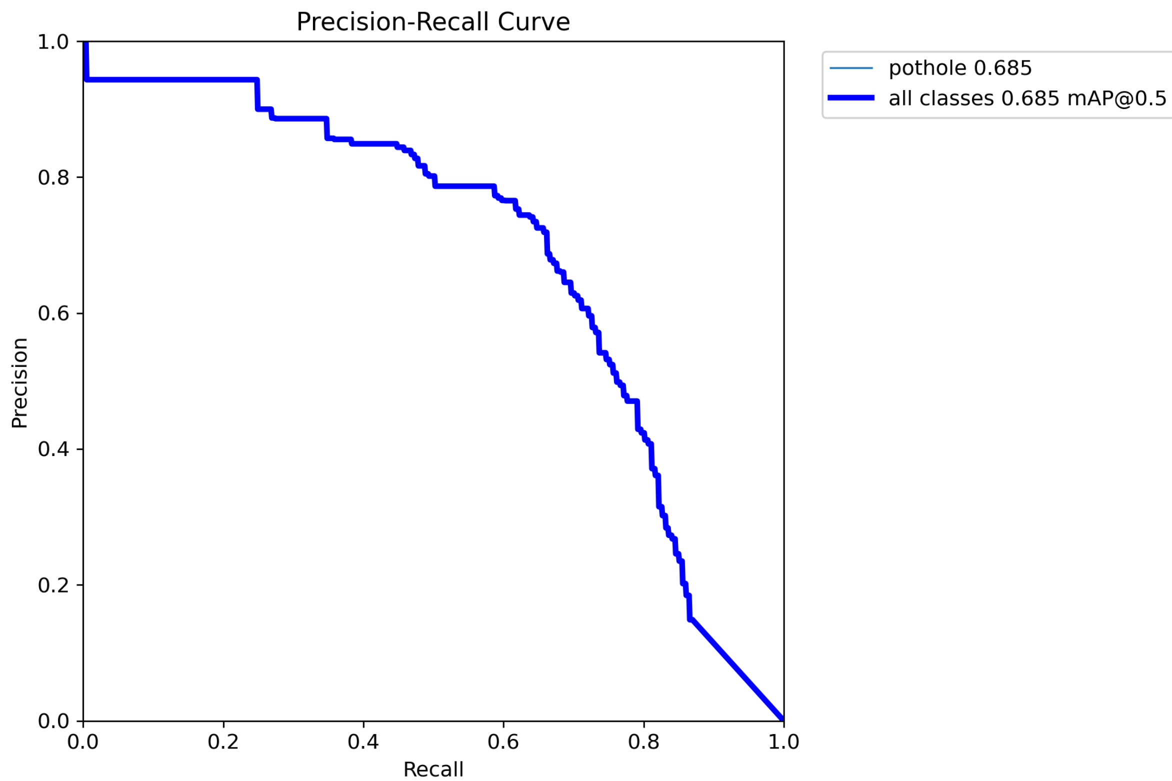

Figure 12: Box PR Curve

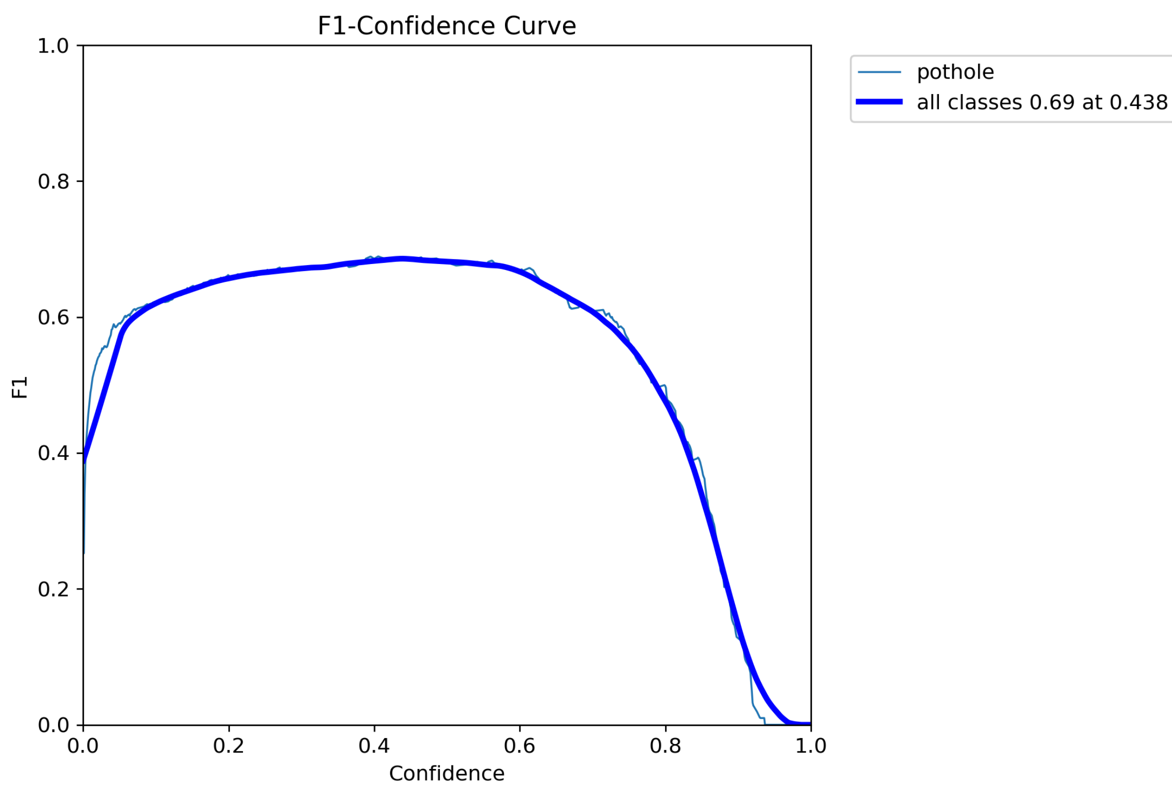

Figure 13: Box F1 Curve

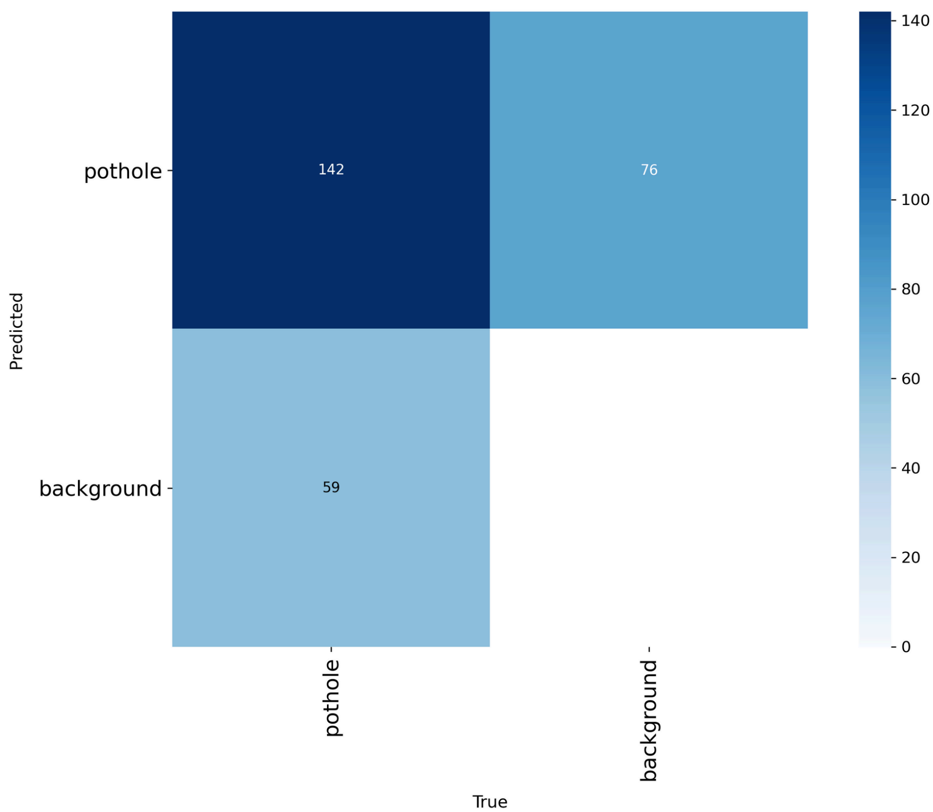

Figure 14: Confusion Matrix

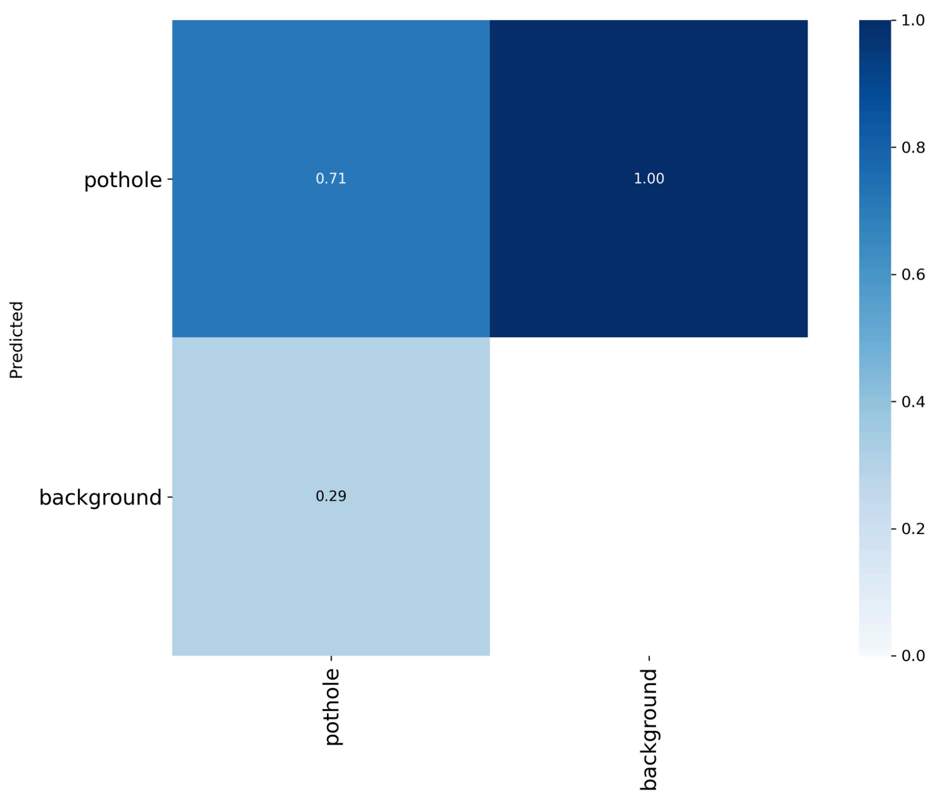

Figure 15: Confusion Matrix Normalized

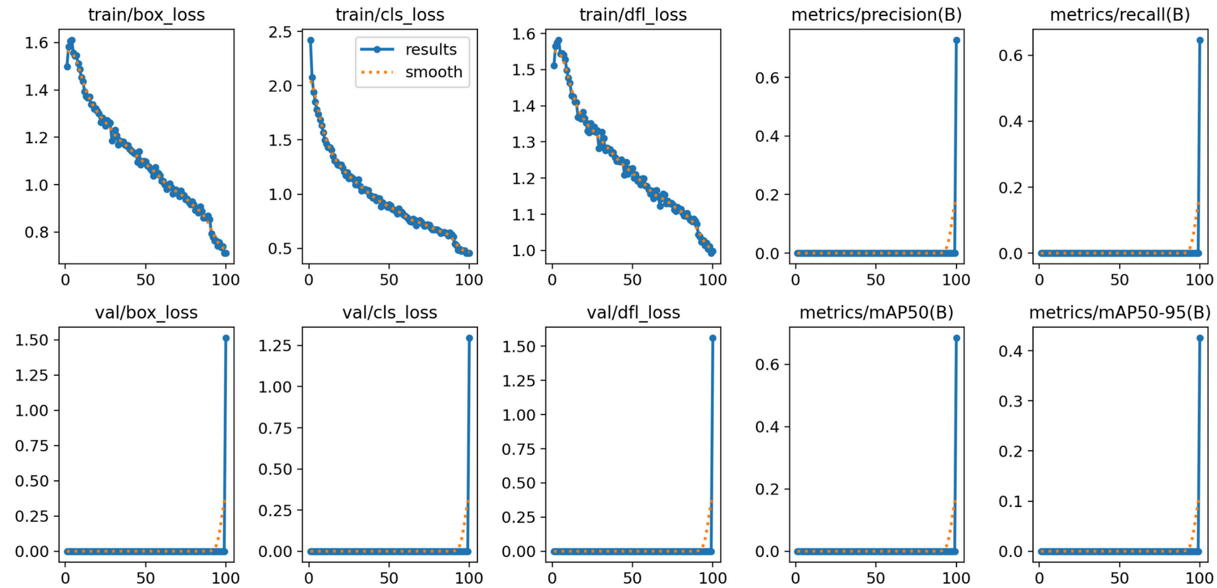

Figure 16: Results

L86 fx 0 ✦ Summarize this data

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	epoch	time	train/box_loss	train/cls_loss	train/dfl_loss	metrics/precisior	metrics/recall(B)	metrics/mAP50(I	metrics/mAP50-!	val/box_loss	val/cls_loss	val/dfl_loss	lr/pg0	lr/pg1	lr/pg2
2	1	20.9257	1.4991	2.42211	1.51106	0	0	0	0	0	0	0	0.000662963	0.000662963	0.00066
3	2	38.6859	1.58272	2.074	1.56432	0	0	0	0	0	0	0	0.00131647	0.00131647	0.0013
4	3	56.1466	1.60663	1.94169	1.57501	0	0	0	0	0	0	0	0.00195677	0.00195677	0.0019
5	4	73.6724	1.61173	1.84913	1.58233	0	0	0	0	0	0	0	0.0019406	0.0019406	0.001
6	5	91.1402	1.55726	1.77872	1.54317	0	0	0	0	0	0	0	0.0019208	0.0019208	0.001
7	6	108.619	1.54444	1.73705	1.54471	0	0	0	0	0	0	0	0.001901	0.001901	0.00
8	7	126.271	1.54513	1.68362	1.54014	0	0	0	0	0	0	0	0.0018812	0.0018812	0.001
9	8	144.391	1.51078	1.63148	1.52747	0	0	0	0	0	0	0	0.0018614	0.0018614	0.001
10	9	162.479	1.48772	1.56727	1.49819	0	0	0	0	0	0	0	0.0018416	0.0018416	0.001
11	10	180.317	1.45308	1.49722	1.47767	0	0	0	0	0	0	0	0.0018218	0.0018218	0.001
12	11	198.012	1.43607	1.46437	1.46238	0	0	0	0	0	0	0	0.001802	0.001802	0.00
13	12	215.385	1.3933	1.42946	1.42831	0	0	0	0	0	0	0	0.0017822	0.0017822	0.001
14	13	234.102	1.37317	1.43059	1.42466	0	0	0	0	0	0	0	0.0017624	0.0017624	0.001
15	14	254.01	1.36766	1.40672	1.41095	0	0	0	0	0	0	0	0.0017426	0.0017426	0.001
16	15	273.428	1.37069	1.34975	1.4089	0	0	0	0	0	0	0	0.0017228	0.0017228	0.001
17	16	293.146	1.33857	1.3108	1.36908	0	0	0	0	0	0	0	0.001703	0.001703	0.00
18	17	312.671	1.33895	1.29724	1.37053	0	0	0	0	0	0	0	0.0016832	0.0016832	0.001
19	18	332.08	1.31958	1.26999	1.36468	0	0	0	0	0	0	0	0.0016634	0.0016634	0.001
20	19	352.133	1.32085	1.2673	1.38261	0	0	0	0	0	0	0	0.0016436	0.0016436	0.001
21	20	371.962	1.31068	1.26949	1.36592	0	0	0	0	0	0	0	0.0016238	0.0016238	0.001
22	21	392.109	1.30052	1.24136	1.35331	0	0	0	0	0	0	0	0.001604	0.001604	0.00
23	22	411.566	1.26461	1.20371	1.33007	0	0	0	0	0	0	0	0.0015842	0.0015842	0.001
24	23	431.009	1.28197	1.17345	1.32545	0	0	0	0	0	0	0	0.0015644	0.0015644	0.001
25	24	452.954	1.27202	1.19839	1.3517	0	0	0	0	0	0	0	0.0015446	0.0015446	0.001
26	25	472.286	1.24825	1.14568	1.32984	0	0	0	0	0	0	0	0.0015248	0.0015248	0.001
27	26	491.527	1.27268	1.16393	1.34168	0	0	0	0	0	0	0	0.001505	0.001505	0.00
28	27	510.983	1.26555	1.14961	1.32785	0	0	0	0	0	0	0	0.0014852	0.0014852	0.001
29	28	530.277	1.26038	1.13269	1.33201	0	0	0	0	0	0	0	0.0014654	0.0014654	0.001
30	29	549.845	1.18634	1.08634	1.28224	0	0	0	0	0	0	0	0.0014456	0.0014456	0.001
31	30	571.039	1.20223	1.09397	1.28793	0	0	0	0	0	0	0	0.0014258	0.0014258	0.001
32	31	590.755	1.23209	1.13452	1.32685	0	0	0	0	0	0	0	0.001406	0.001406	0.00
33	32	610.313	1.2072	1.07579	1.3108	0	0	0	0	0	0	0	0.0013862	0.0013862	0.001
34	33	630.068	1.16816	1.03186	1.27622	0	0	0	0	0	0	0	0.0013664	0.0013664	0.001
35	34	650.874	1.18556	1.04241	1.28424	0	0	0	0	0	0	0	0.0013466	0.0013466	0.001
36	35	671.297	1.18197	1.04359	1.28129	0	0	0	0	0	0	0	0.0013268	0.0013268	0.001
37	36	690.79	1.18255	1.03133	1.27929	0	0	0	0	0	0	0	0.001307	0.001307	0.00

results.csv

Figure 17: Results Logs

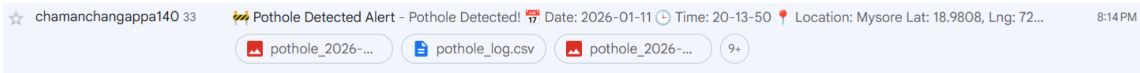

Figure 18:Alert Notification

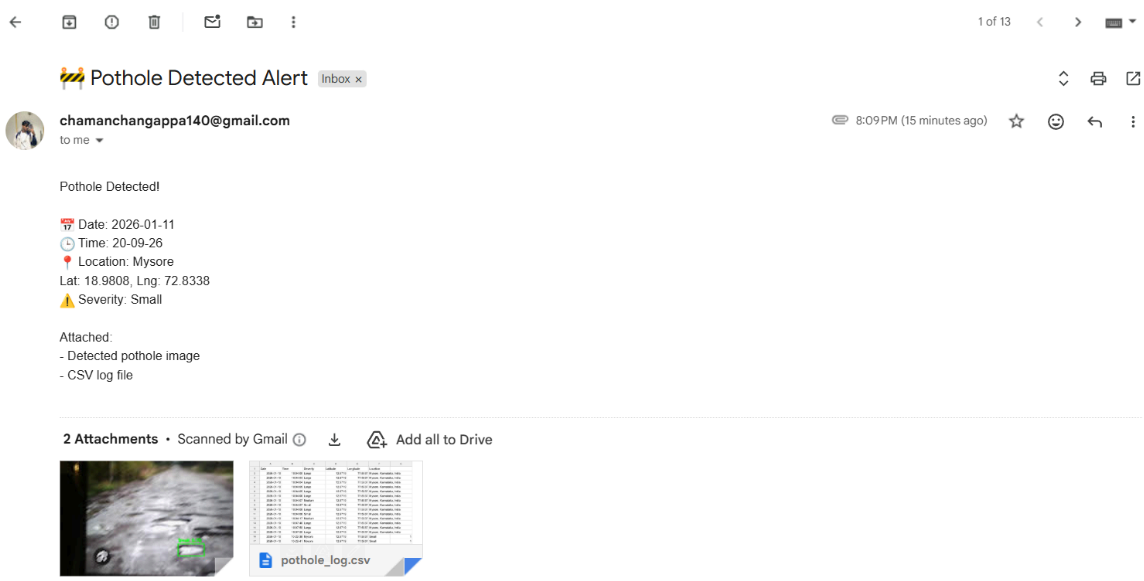

Figure 19:Alert Mail

CONCLUSION

This project proposes an automated pothole detection and notification system using the YOLO model. The system is capable of accurately identifying potholes in real-time dashcam recordings and uploaded images of the road, highlighting the pothole region along with the confidence levels. The severity levels of the identified potholes enable efficient prioritization of road maintenance work. Furthermore, the details of the pothole detection, including the confidence level, severity level, date of detection, time of detection, and geographical position, are recorded in the CSV form. The system includes automated notifications in the form of emails along with the CSV log when high-level potholes are identified. The system comes with an intuitive dashboard for real-time analysis and control of the system, while the standalone executable feature of the system enables the system to run in the offline environment. The system proposes an optimal solution for intelligent road management.

FUTURE SCOPE

The pothole detection system proposed here has many potential areas of development in the future. The pothole detection system can be developed further to detect other defects like cracks, surface, and patches in the roads using multi-class deep learning models.

The pothole detection system can be combined with the GPS map services provided in the modern world, which can help visualize the pothole detection information in real-time, helping the authority in path planning. The pothole severity level detection model can be improved further using the depth information of the pothole detection using sensors. The model can be developed further to be embedded in edge devices, mobile, and automotive systems. The pothole detection database can be further integrated with the smart city database, which can help develop an automated maintenance analysis database. The pothole detection model can be further developed to be retrained using the newly collected data.

REFERENCE

1. Roboflow, Pothole Detection Dataset – Used for training the model (approximately 3,490 images). Available at: https://roboflow.com
2. Ultralytics, YOLOv8 Documentation – Used for understanding YOLOv8 architecture, training, and inference. Available at: https://docs.ultralytics.com
3. Python Software Foundation, Python Programming Language – Core programming language used for system development. Available at: https://www.python.org

4. OpenCV, Open Source Computer Vision Library – Used for image processing, video capture, and real-time detection.
5. NumPy, Numerical Computing Library for Python – Used for array operations and numerical processing. Available at: https://numpy.org
6. Pandas, Data Analysis and CSV Handling Library – Used for CSV logging and data management. Available at: https://pandas.pydata.org
7. Matplotlib, Data Visualization Library – Used for plotting and visual summaries (if applicable). Available at: https://matplotlib.org
8. SMTP Email Protocol, Email Automation Service – Used for sending automated email alerts with CSV attachments.
9. LabelImg, Image Annotation Tool – Used for annotating pothole images in YOLO format. Available at: https://github.com/tzutalin/labelImg
10. PyInstaller, Python Executable Builder – Used to convert the Python application into a standalone .exe file. Available at: https://pyinstaller.org
10. Likitha et al. described a YOLOv8-based road damage assessment system for detecting potholes and various type-of-crack classifications.
11. Lv et al. surveyed the state-of-the-art models for real-time object detection, focusing on feature representation and multi-scale learning. Their research concentrated on efficiency and speed of inference for small and irregular road instances.
12. Arya et al. carried out a comparative study of deep learning algorithms to detect potholes and cracks. dataset quality and accuracy of data annotation emerged as important factors in this case.
13. Safyari et al. thorough survey on vision-based pothole detection methods was presented by .This research examined traditional methods, machine-learning methods, deep-learning methods, and YOLO.
14. Paramarthalingam et al. presented a system based on YOLO that detected potholes in a real-time manner Low latency during inference in real-time scenarios has been highlighted in this study.
15. Parvin et al. proposed pothole detection using deep learning from multi-weather images. The experiment showed robustness in rain, fog, and lighting conditions.The operation-related functionality like logging and reporting was not considered.
16. Shaghouri et al. analyzed a number of YOLO models for pothole detection in a live video feed. The studied tasks covered accuracy versus inference speed trade-offs in practical applications. System-level capabilities such as user interfaces and data processing were not taken into account.
17. Addanki et al. presented a pothole detection system using YOLOv8; however, it was trained on a self-made and labelled dataset.It then discussed dataset preparation and model training strategies for real-time detection.Analytics and decision-support mechanisms post-detection were not put in place.
18. Ruseruka et al. proposed a pothole detection system with visually-based estimation of the size. It coupled object detection with geometric analysis to estimate the severity in potholes. All other practical deployment aspects, like interfaces and reporting, were hardly addressed.
19. Uduwage et al. proposed a pothole detection method based on deep learning, integrated with traffic information.The research focused on the decision-making aspect of road maintenance prioritization. It was theoretical without an overall executable system.
20. Kumar et al. proposed a solution for detecting potholes using CNN and YOLOv3 models suitable for Indian roads. The accuracy in classification and localization for the models was compared. Severity analysis and automatic reporting were not included.
21. Lee et al. presented a YOLO-based system for pothole detection, which is implemented on edge devices. It presented a low latency inference with limited computations. Such advanced functionalities as dashboards were not added.
22. Nguyen et al. proposed a road damage detection system based on Faster R-CNN and Detectron2. The results of this research were highly accurate for detection. High complexity of calculations hindered real-time processing.
23. Khan et al. presented an improved YOLOv8-based pothole detection model for intelligent vehicles. The project aimed to enhance detection accuracy and eliminate false positives. System integration and maintaining processes were also not considered.
24. Jeong, et al. also designed a smartphone-based road damage detection system utilizing YOLO models. The system made it possible to accomplish road monitoring in a cost-effective and scalable manner through the use of mobile devices The classification process concerning severity levels and notifications through automation were not incorporated.