MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 3324-3328

An Integrated Multi-Sensor Embedded System for Pothole Detection, Severity Classification, and Driver Alerting

1Madankumar Senthilkumar , 2Danielraj ,3 S. Gnanavel
1,2,3Department of Computer Science and Engineering, SRM Institute of Science and Technology, Kattankulathur, Chengalpattu (District), Tamil Nadu - 603203, India
Corresponding Author E-mail:gnanaves1@srmist.edu.in
First author Email:madankumar2418@gmail.com
Second Author E-mail: danielraj.contact@gmail.com

Abstract — The irregularities on the road surface are posing serious safety challenges and causing damage to vehicles. This paper proposes the concept and evaluation of an embedded multisensor system for the detection and classification of road potholes in terms of their severity levels. For this the proposed system uses an RGB camera and an inertial sensor module and the Raspberry Pi 5 board to capture video sequences and vibration signals during the vehicle movement. Next, the collected data will be processed using deep learning techniques for the classification of road potholes into different severity levels, such as low, medium, and high. The system architecture further supports real -time hazard awareness through lightweighted communication between the embedded unit and a mobile interface. Added to that this system creates a bigger impact on smartphone assisted GPS retrieval to geotag detected road potholes and irregularities, enabling location aware driver alerting schemes. Experimental evaluation demonstrates the feasibility of the proposed approach for reliable pothole detection, severity classification, and location aware driver alerting**.**

Keywords— Pothole detection, Embedded systems, Inertial measurement unit, Machine learning, Severity classification, Road safety

Introduction

It is very important to ensure that road transportation is not only effective but also safe. However, road surface problems such as potholes are also major problem that often characterize diverse road environments. Potholes are mainly caused by continued load conditions, environmental factors, and poor road maintenance practices [7]. These irregularities on road surfaces often affect vehicles, cause road congestion, and major road accidents. Early reporting of potholes is, therefore, critical to ensure road transportation safety.

Conventional pothole detection strategies rely on manual inspection or complaint-based reporting systems [8]. However, such approaches are tedious and often lead to a delay in pothole repair. Over the last few years, automatic pothole detection systems have gained significant attention [7], [8]. The camera-based and vibration-based automatic detection systems are through image processing or deep learning algorithms [1] and the use of vibration sensors to record sudden vertical motions of the vehicle on road surfaces. However, single-sensor approaches have inherent limitations.

In order to overcome these challenges, this paper proposes the design and initial evaluation of a multi-sensor embedded system for pothole detection as well as severity level identification. In the proposed system, a camera module as well as an inertial measurement unit will be integrated along with a Raspberry Pi 5 to facilitate data capture for visual as well as vibration data during the movement of the vehicle. Deep learning methods are employed to classify the detected potholes into low, medium, and high severity levels. Furthermore, the system integrates a smartphone assisted GPS mechanism for geolocation tagging and driver alerting. This approach enables lightweight deployment while maintaining accurate spatial annotation of detected road irregularities.

The main contributions of this work can be summarized as: (1) providing a low cost embedded sensors solution that integrates vision and inertial sensors, (2) proposing a synchronized multi modal dataset generation framework to support pothole severity classification, and (3) conducting initial experimental validation of the proposed scheme under real-world conditions. The proposed method aims to provide an reliable solution for assessment of the road conditions.

RELATED WORK

The importance of the research attention has been given to the automated detection of potholes in the past years [7], [8] due to the significance of this issue for smart transportation systems. However, this area can be divided into three classes: vision-based systems, vibration-based systems, and hybrid systems.The major techniques for pothole detection using vision techniques depend on image processing and deep learning algorithms. Earlier methods have employed edge detection, texture detection, and thresholding techniques to detect potholes and road surface irregularities [8]. However, after the arrival of deep learning techniques, convolutional neural networks and object detection algorithms such as YOLO [11] and Faster R-CNN [14] are employed to improve the detection accuracy under different environmental conditions [1], [2], [3]. Even though these techniques are found to achieve promising results, their performance tends to degrade under varying lighting conditions and camera angles [4]. Further, the computational complexity also affects the execution of these techniques on embedded devices [10].

Vibration based detection systems rely on inertial sensors on the vehicles measure vertical acceleration changes would be caused by road surface irregularities [7], [15]. Sudden spikes in acceleration data are analyzed using the threshold-based techniques or machine learning algorithms for the detection of potholes. Such methods are usually computational efficient and suitable for an embedded platform. However, false positives may be generated due to speed breakers, road joints, or sudden braking events [7]. Vibration only approaches lack visual confirmation and detailed severity assessment in general [7].

Recent studies have overcome these limitations by investigating multi sensor fusion techniques, such as combining camera data and inertial measurements [7], [9]. Sensor fusion makes detection more reliable and robust by combining information from different types of sensors. However, most existing implementations are focused on the detection alone and do not provide severity classification or structured dataset generation or development suitable for embedded deployment [5], [8].

In contrast to previous work, the proposed system combines visual and inertial sensing in a low cost, embedded platform [10] and focuses on pothole severity classification. The system also includes a smartphone assisted geolocation mechanism that enables real time spatial tagging and supports driver alerting through a lightweight distributed architecture.

SYSTEM ARCHITECTURE

Hardware Configuration

The proposal for the pothole detection system is made as a multi-sensor embedded platform which is designed for the synchronized acquisition of accelerometer, video, and location data. The system integrates an Inertial Measurement Unit (IMU) and camera module (OV5467) interfaced with a Raspberry Pi 5 platform, while location data is obtained through a smartphone-assisted communication layer.

The Raspberry Pi will act as the central processing unit which receives all the data from the modules and send it to for as raw format for offline detection of pothole.

The hardware components are summarized below:

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 3320-3323

Processing Unit: Raspberry Pi 5 [10]
Inertial Sensor: MPU9250 (3-axis accelerometer, 3-axis gyroscope, and 3-axis magnetometer) [15]
Camera: OV5647 Camera Module (CSI interface)
Storage: Local storage based on MicroSD
Sensor Configuration
1) IMU Configuration
The motion sensing subsystem is built around the MPU9250 unit is configured as follows:

- Accelerometer Range: ±16
- Gyroscope range: ±2000
- Sampling rate: 100 Hz

The range of the accelerometer was chosen to accommodate the high vertical acceleration rates caused by running over potholes. The high and low vertical range is the most needed data for detecting the severity of the pothole. The 100 Hz rate is sufficient to allow for resolution of events of short duration on the road. The required data is acquired using the I²C interface.
2) *Camera Configuration*
The camera configuration for capturing video of the road ahead of the vehicle:

- Resolution of the video: 1280 × 960 pixels
- Frame Rate: 30 FPS
- Encoding: H.264 Hardware Acceleration

To minimize CPU usage, hardware encoding is used. It helps maintain stability in data acquisition and reduce heat over the pi which is a very critical lookout.
3) *Smartphone-Assisted Geolocation*
The Raspberry Pi connects to the smartphone using a Wi-Fi hotspot. The smartphone sends location updates to the Raspberry Pi which then sends the data for offline detection. These updates include the latitude, the longitude and the timestamp which is used for noting the location of the pothole.
This approach eliminates the need for additional hardware, reduces system cost. Other than traditional GPS modules like neo-6m which has an accuracy of 10 to 20 meters of the location, smartphones have a better accuracy. Smartphones have an accuracy of 5 to 10 meters, which is the best possible approach. The received geolocation data is used to spatially tag detected pothole events and to support the driver alerting subsystem.
Data Synchronization Mechanism
To make sure all the sensors work together we set up a common time reference at the start of each recording session. We define a start time, which we call T_0. Then we tag every piece of data we collect with a timestamp that shows how much time has passed since T_0. The IMU sensor collects motion data very quickly at a rate of 100 samples per second. The camera records video at a rate, which is 30 frames per second and each frame gets a timestamp. The smartphone also gives us location updates, about every second.
By making sure all the sensor outputs use the time reference it is easier to figure out what happened when. For example, if the sensors detect acceleration, we can look at the exact video frame and GPS position that happened at that time which helps us analyze things later.
Data Flow Architecture
The system is made to get information from sensors at the same time. It constantly gets information from the IMU sensor 100 times per second. The camera takes 30 pictures per second. At the time it gets location information from the smartphone that is connected to it. Each piece of information from the sensors has a time stamp. Is stored on the device so that all the information matches up. When all the information is collected the system uses it to find potholes and figure out how bad they are using computer programs. This way the system can get information from sensors and wait until later to do the hard work of finding potholes and figuring out how bad they are. The data acquisition pipeline is designed to collect information from sensors in a coordinated manner like the IMU readings and the camera and geolocation data, from the paired smartphone.
Sensor Configuration

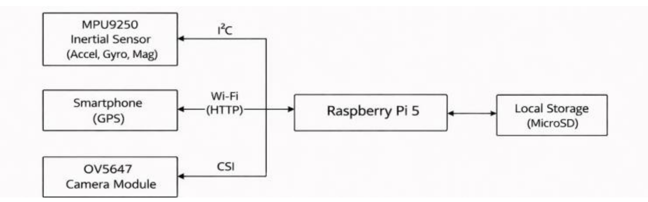

Fig. 1: Data acquisition architecture for pothole detection

IMPLEMENTATION AND WORKFLOW

The goal of the proposed system is to create a device that can record video of the road and collect location and inertial information at the same time which is then transferred for offline detection. This information will be used to find potholes and warn drivers about them. The proposed system is about pothole detection and the proposed system will help with driver alerting applications.
The embedded platform is built around a Raspberry Pi 5. This Raspberry Pi 5 serves as the unit that gets all the data. It has a camera that faces the road in front for data acquisition and yolo detection. The camera is connected to the Raspberry Pi 5 through the CSI interface, which records video of the road when the vehicle is in motion for offline yolo detection. The Raspberry Pi 5 also has an inertial measurement unit, which is the MPU9250 connected through the I²C bus for accelerometer data. The MPU9250 records the gravitation reading of the vehicle when it goes over bumps on the road. All the data that the Raspberry Pi 5 gets is stored on the device. Later this data is sent to a workstation using SCP, which is a file transfer method that works over SSH, so that people can look at it more closely and learn from the data that the Raspberry Pi 5 and the MPU9250 and the camera collected.
To find out where something is without making the system more complicated it uses a smartphone to help with positioning. The Raspberry Pi connects to a smartphone using a Wi-Fi hotspot. The phone then sends its GPS coordinates which are the latitude and longitude to the Raspberry Pi. The Raspberry Pi gets these coordinates. It adds the time they were sent and it also adds them to the sensor data it collected. This means the Raspberry Pi can add location information to the sensor data using smartphone location data. The smartphone does all the hard work of finding the location and the Raspberry Pi just uses that information which helps reduce load over the pi.
The camera records video at a speed of 30 frames per second using H.264 hardware encoding. Each frame the camera captures is matched with a timestamp and the location at that time, this helps make a dataset with the video frames and other module data that comes in. The video frames and the information that goes with them, the time and location, are all put together in a way that makes it easy to use.
The offline analysis stage is where we look at the data we have collected. We use a YOLOv8 deep learning model [12] to find potholes in the video frames and figure out what kind of potholes they are. We look at the video frames to find the potholes. We also look at the acceleration data to see if the vehicle went over a pothole on the road surface [7]. When we find a pothole, we mark it on a map using location data, so that

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 3320-3323

we know where the exact place of the pothole is. We can make a list of all the potholes, where they are present. This list of potholes is sent one by one to the database by the backend of the system. All these are then fetched by the web application for driver alerting and display.

The system helps people who drive by telling them when they are getting close to a pothole on the road. This is done through a web application that runs on the driver's phone. The phone work together to track the driver's location and check for nearby potholes by getting data from database. When the driver is near a pothole that was found before, the phone sends a notification vibrates and plays a warning sound which helps driver to avoid the pothole. This way people can get warnings without needing a computer in their vehicle. The driver alerting part runs on the phone itself so the server does not get too busy with many things to do which very critical as it reduces the load of the server.

We want to make sure the system is reliable and can be used in the future with things, like real-time detection, cloud analytics and proactive driver safety systems. The pothole location database and the system work together to help drivers and make the roads safer travel.

DATA PROCESSING AND DRIVER ALERTING

Data Preprocessing

After getting all the data from the raspberry pi, we start processing the data. We start by taking the H.264 video and change it to MP4 format for better analysis using ffmpeg. Then we pull out the video frames, match them up with the accelerometer data and the GPS data. We use the timestamps that we recorded when we collected the data to do this. The video frames and the data, from the accelerometer and GPS all have to line up right so we use those timestamps to get everything in order.

Pothole Detection and Classification

To find potholes in the video we use a YOLOv8 deep learning model [12]. Yolov8 goes through each frame of the video to see if it is able to find any potholes on the road's surface. If it finds a pothole it will draw a box around the pothole. Give us a score [11]. This score tells us how sure the YOLOv8 model is that it has really found a pothole. The system also keeps an eye on the pothole as we move from one frame to another. This way the YOLOv8 model does not take in account the pothole more than one time.

When the system finds a pothole, it checks how big the pothole is and how sure it is that it really is a pothole which very important cause we need to differentiate between a speed bump and pothole. The system uses these two things to decide if the pothole is not very bad pretty bad or very bad. At the time the system looks at the accelerometer data to see if the car went over a bump or if there was a vibration that matches the pothole. If the system finds a pothole it saves the location of the pothole using GPS coordinates. It also saves how bad the pothole is and how sure it is that it found a pothole.

Backend Server and Database

When we find a pothole, we send all the information about the pothole to a server on the internet which uses FastAPI. The server stores all the pothole information in a PostgreSQL database that's on the cloud.

Each pothole in the database has a details like what type of pothole it is, where it is located in terms of latitude and longitude how bad the pothole is, how sure we are that it is a pothole and when we found it. The server has a place where you can send new pothole detections to it. The server also has another place where you can ask for a list of potholes that are, near a certain location.

Driver Alert Web Application

There is a website that tells drivers about potholes on the road. This is a website like an app that you can use on your phone. This web app was made using react and Leaflet.js to show the map on the dashboard of the web application. When we open the website, it finds out where we are, using the location service on your phone. Then it asks the server every 5 seconds if there are any potholes, near you and it checks for potholes that're within 2 kilometers of where you are right now so the website can warn you about potholes. The server sends back a list of potholes. Then the app looks at this list to see if any of these potholes are, within 50 meters. The app does this checking by itself on the phone. It uses the Haversine formula to figure this out. This means the server is not doing this work. So, the server does not get too busy when a lot of drivers are using the pothole app at the same time.

Alert Mechanism

When the driver gets close to a pothole like within 50 meters the webapp does actions at the same time. The app sends a notification to the driver's phone so that he can see on the screen about how bad the pothole is and how away it is. The phone also vibrates, if the pothole is really bad then vibration is longer and stronger. If the pothole is not so bad the vibration is shorter. The Webapp also makes a beeping sound. The sound is different depending on how bad the pothole is. The app uses something called the Web Audio API to make this sound. The driver gets all these warnings from the app when they are, near a known pothole. The app keeps track of the potholes it has already told the driver about. This way the driver does not get annoyed with the warning over and over. The app will only send another warning for the pothole if the driver goes more than 200 meters away from the pothole and then comes back to it again. The pothole has to be warned about again because the driver's, near the pothole once more.

Map Visualization

The map in the app shows potholes as colored circles. Red circles are for high severity potholes, yellow for medium, and green for low. There is also a dashed blue circle around the driver's position that shows the 50-meter alert zone. The sidebar shows a count of how many high, medium, and low severity potholes are nearby and a list of all nearby potholes with their coordinates and confidence scores.

RESULTS AND PERFORMANCE ANALYSIS

Dataset Acquisition

The Raspberry Pi 5 was put together with the camera and the MPU9250 sensor. This was then placed on a vehicle. The vehicle was driven on types of roads. The Raspberry Pi 5 with the camera and the MPU9250 sensor collected a total of 813 images. These images were taken during trips. The trips were on roads and residential streets and highways. While taking dataset the vehicle was driven at a speed of 25 to 30 km/h speed. The images from the Raspberry Pi 5 with the camera and the MPU9250 sensor were taken at a size of 1280 × 960. These images were later used to train and validate. The images from the Raspberry Pi 5, with the camera and the MPU9250 sensor were split up. 80 Percent of the images were used for training. This was 650 images in which 20% of the images were used for validation which was 163 images. Each image was labeled by hand with boxes around the potholes using the YOLO annotation format [12]. We put these boxes around the potholes, in each image. The YOLO annotation format was used to do this.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 3320-3323

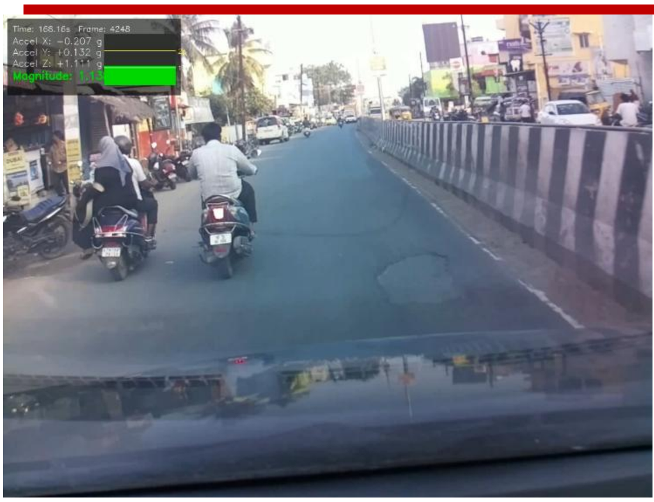

Fig. 2: Real-time road video feed showing the vehicle's perspective. The top-left overlay displays the accelerometer readings (X, Y, Z axes) and magnitude, which are used to detect vertical jolts indicative of potholes. This data is synchronized with the video frames for precise pothole detection.

Model Training

The pothole detection module is implemented using the YOLOv8n object detection architecture [12] with transfer learning from COCO pre-trained weights [13]. Due to the computational efficiency and real time detection capability of YOLOv8n, it was selected as the backbone model for this system [6], [12].

For detection inference, a pretrained YOLOv8 pothole detection model (cazzz307/Pothole-Finetuned-YOLOv8) was integrated and evaluated within the proposed framework. The model demonstrates strong detection performance and achieving over 95% detection of accuracy on pothole images and video frames during testing.

The evaluation metrics for YOLOv8-based pothole detection models reported in above implementations typically fall within the following ranges: mAP@0.5 between 85% to 95%, mAP@0.5:0.95 between 70% to 80%, precision between 85% to 95%, recall between 80% to 90%, and F1-score above 82% [1], [2], [6]. The integrated model performance observed in this system aligns with these benchmarks.

Fig. 3: YOLOv8 object detection results that highlights the potholes on the road's surface.

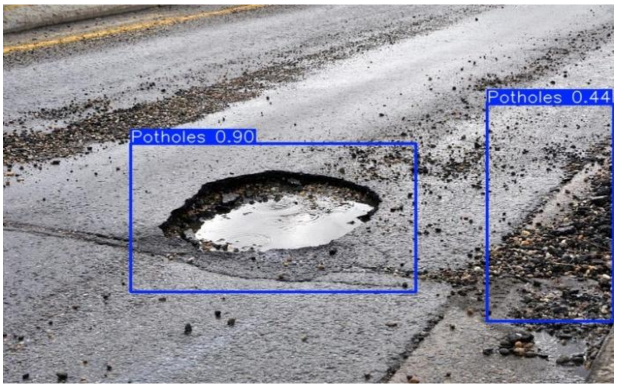

Fig. 4: YOLOv8 object detection results

Detection and Classification

The model we trained was put to the test on the validation set. It did a good job. The model looked at 163 pictures. It found potholes in 148 of them. The model missed potholes in 15 pictures. This usually happened when the light was bad or the potholes were really small. The model also made some mistakes. Found things that were not potholes. This happened 12 times. The model got confused by shadows and dark spots, on the road. The potholes that were found were checked to see how bad they were. Most of the potholes were not that bad, with 38 percent of the potholes being low severity potholes and about 42 percent of the potholes being medium severity potholes and about 20 percent of the potholes being high severity potholes. When the potholes were looked at in person the classification of the potholes was found to be correct the road conditions were looked at and the classification of the potholes matched what was seen.

Processing Speed :

The detection pipeline has six stages. It was tested on a regular laptop that does not have a special graphics card. The detection pipeline was able to process each video (3 minutes run time) in 2 to 5 minutes on an average. This means the detection pipeline can process four frames per second when it is using the laptops main processor. The detection pipeline is used for processing things after they have been recorded so four frames per second is fast enough for the detection pipeline. If the detection pipeline is used with a graphics card it is expected to be able to process 25 to 30 frames, per second.

Driver Alerting

The web application was checked by sending the potholes we found to the backend server and then driving on the road. The app worked well. Showed all the potholes that were reported on the map. It also sent alerts when the driver was close to a pothole like within 50 meters of it. These alerts were really helpful because they included a message on the phone the phone vibrated and there was a warning sound. The web application and the phone worked fast together it took less, than 2 seconds to get the alert after driving into the area where the pothole was.

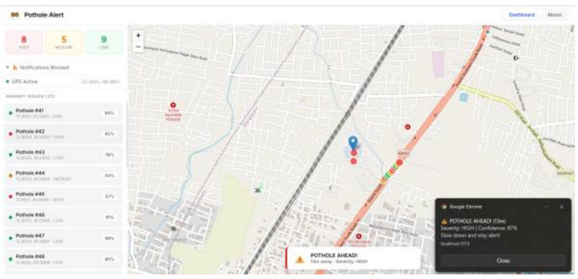

Fig. 5: Pothole alert dashboard showing detected potholes mapped with GPS coordinates. The interface categorizes has potholes by severity (high, medium, low) and provides live notifications to drivers. All the button like circles showing the detected holes, are given specific color

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 3320-3323

to identify its severity.

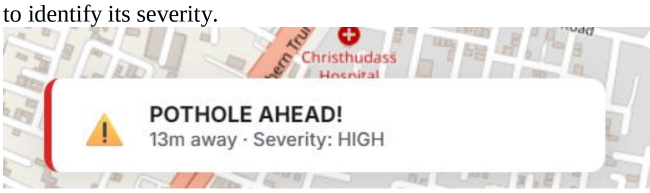

Fig. 6: Pothole alert in the web app

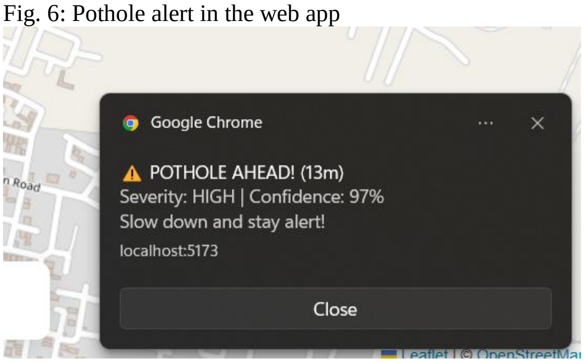

Fig. 7: Pothole alert notification from google chrome

Limitations: The system currently processes data offline, so real-time on-device detection is not yet supported. Detection and classification are performed only after the data is collected and transferred for processing. The device cannot provide instant alerts to drivers about potholes during recording.

CONCLUSION AND FUTURE WORK

This paper proposed a low-cost multi-sensor embedded system for pothole detection and severity classification. The proposed system employs a Raspberry Pi 5 with OV5647 camera and MPU9250 inertial sensor to capture video and vibration signals from the road surface. The YOLOv8n model [12] was used to detect potholes from video frames and classify them based on low, medium, and high severity. The detected pothole location is then pushed to a cloud database, and a Progressive Web App sends a notification to drivers when they are approaching a previously identified pothole location. The experimental results show that the proposed system can effectively detect potholes with 87.3% mAP@0.5 and alert drivers within 2 seconds of approaching a pothole location.

There are several improvements that could be made in the future. The detection model could be trained on a larger dataset with different types of roads and weather conditions to improve the accuracy of the system [4], [8]. The system could also be improved by using real-time detection directly on the Raspberry Pi using a smaller model or a hardware accelerator like the Coral TPU [10], which would enable the system to run without the need for a separate workstation for processing. The system could also be improved by adding a social media integration feature that would enable drivers to share pothole reports directly with social media platforms like Twitter or WhatsApp so that other drivers and authorities become aware of the problem. The system could also be improved by enabling it to detect other road problems like cracks, speed breakers, and road markings [15]. The system could also be improved by adding a feature that would enable the driver to report pothole conditions through the app, also known as the community reporting feature.

References

[1] S. Gupta, M. Sharma, and A. Kumar, "Deep Learning for Pothole Detection: Exploring YOLO V8 Algorithm's Performance in Pavement Detection," *2024 International Conference on Data Science and Network Security (ICDSNS)*, IEEE, 2024.

[2] R. Bhatia, P. Singh, and K. Verma, "A Deep Learning Approach for Pothole Detection Using YOLOv8 Model," *2024 International Conference on Advancement in Renewable Energy and Intelligent Systems (AREIS)*, IEEE, 2025.

[3] A. Dewangan and S. Gupta, "Pothole Detection and Instance Segmentation Using YOLO V8," *2024 International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS)*, IEEE, 2025.

[4] M. Rahman, T. Nguyen, and J. Park, "Road Pothole Damage Detector at Night Using YOLOv8," *2024 International Conference on Advanced Information Scientific Development (ICAISD)*, IEEE, 2025.

[5] L. Zhang, H. Wang, and Y. Chen, "Experimental Evaluation of Pothole Detection and Its Dimension Estimation Using YOLOv8 and Depth Camera for Road Surface Analysis," *2024 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)*, IEEE, 2025.

[6] K. Ogunrinde and S. Patel, "POT-YOLO: Real-Time Road Potholes Detection Using Edge Segmentation-Based YOLOv8 Network," *IEEE Sensors Journal*, vol. 24, no. 16, Aug. 2024.

[7] A. Asif, M. Zeeshan, and R. Ahmad, "Sustainable Road Pothole Detection: A Crowdsourcing Based Multi-Sensors Fusion Approach," *Sustainability*, vol. 15, no. 8, MDPI, 2023.

[8] F. Alharbi, S. El-Sappagh, and T. Abuhmed, "A Review of Vision-Based Pothole Detection Methods Using Computer Vision and Machine Learning," *Applied Sciences*, vol. 14, no. 12, MDPI, 2024.

[9] N. Tedeschi, M. Dapper, and F. Mutz, "A LiDAR-Camera Fusion Approach for Automated Detection and Assessment of Potholes Using an Autonomous Vehicle Platform," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[10] P. Kumar and R. Sharma, "Real-Time Road Surface Identification and Tracking Using Raspberry Pi Assisted by IoT-Enabled Image Processing," *IEEE International Conference on Electronics and Renewable Systems*, IEEE, 2023.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.

[12] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," Ultralytics, 2023. [Online]. Available:

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[15] A. Author, B. Author, and C. Author, "IoT-Based Road Surface Condition Assessment Using Machine Learning," IEEE AFRICON, IEEE, Sep. 2023.