

National Pothole Monitoring System (NPMS): Edge-Assisted Pothole Detection and Road- Condition Logging using YOLOv8 and ESP32- CAM Telemetry

Prof. A. A. Randive*, Tushar Patil, Dhawal Phalak, and Vaibhav Jaybhay
Department of Electronics and Telecommunication Engineering, AISMSS College of Engineering, Pune

Abstract: The degrading condition of road surfaces such as potholes poses major hazards to road safety, causing extensive vehicle damage, traffic disruption, and high costs of maintenance operations. Traditional approaches to manual road surveys are cumbersome, costly, subjective, and tedious. In this paper, we present the National Pothole Monitoring System (NPMS), offering a comprehensive solution for automatic pothole detection and documentation of road conditions. The NPMS leverages an independently trained YOLOv8 nano deep learning model for vision-based pothole detection, an affordable ESP32-CAM module for video streaming at the edge, and GPS/IMU sensor fusion for accurate geographical and temporal marking of events. We have successfully shown that the NPMS has precision, recall, mean average precision @ 0.50, and mean average precision @ 0.50:0.95 scores of 0.987, 0.85, 0.796, and 0.505, respectively, using a specially curated dataset containing 890 road images. Pothole detection events are timestamped and annotated with GPS location and vehicle attitude data (pitch and roll angles). Therefore, events are recorded in CSV files that can be analyzed further to identify hotspots, plan maintenance schedules, and study road deterioration.

Index Terms-Pothole Detection, YOLOv8, ESP32-CAM, Edge Computing, Object Detection, Road Condition Monitoring, Deep Learning, IoT, GPS Telemetry, Road Safety.

I. INTRODUCTION

The roads make up the foundation of transport infrastructures within any nation. In developed and underdeveloped countries, one common problem that affects many roads is the potholes. Potholes refer to holes that appear in the form of a bowl-like structure as a result of the weakening of the base of the road. As a consequence, the base of the road cannot withstand the loads exerted on it by traffic. Eventually, water enters into the cracks, causing a weakening effect on the layers below and leading to an indentation in the road.

Economic impacts that come with potholes are truly mind-blowing. In fact, in 2023, as research done by the American Automobile Association reveals, economic damages resulting from potholes have cost drivers in America \$3 billion each year. The scenario may be even worse in countries such as India, with its road network totaling over 6.3 million kilometers long. The matter of fact is that the Ministry of Road Transport and Highways (MoRTH) proved that there were many accidents on roads caused by road deterioration, which usually entails the creation of potholes.

The traditional road assessment technique comprises conducting surveys using manual surveys, undertaken by trained professionals and performed at regular intervals along road stretches. However, this technique is ineffective, biased, costly, and fails to give adequate information for maintaining the quality of the road. Crowdsourced mobile application uses civilians as

participants, as they use their smart phones to report on the damages found in road stretches. Nonetheless, neither approach ensures continuous and reliable information collection.

The rise of deep learning-powered computer vision and inexpensive embedded vision technology along with Internet of Things (IoT) capabilities has made it possible for automation in monitoring road surfaces. In this paper, we introduce the National Pothole Monitoring System (NPMS) that is a complete pipeline consisting of (i) a proprietary YOLOv8 nano model capable of real-time pothole detection from video frames; (ii) an ESP32-CAM microcontroller-driven streaming system for capturing low-cost wireless footage from roads; (iii) GPS and inertial measurement unit (IMU) telemetry data to geolocate and contextualize the detections; and (iv) automated log management for analysis.

Following sections in the rest of this paper are structured as follows: Literature Review will be covered in Section II; Dataset & System Architecture will be discussed in Section III; Methodology will be detailed in Section IV; Experimental Results will be covered in Section V.

II. RELATED WORK

Pothole detection using automated systems has witnessed considerable progress during the last ten years by evolving from traditional rule-based image processing algorithms to deep learning. In this paper, we discuss recent studies and present NPMS in its rightful context.

A. LiDAR-Based Detection

As mentioned by Faisal & Gargoum [1], LiDAR technology can efficiently help in the detection of potholes in an inexpensive way with the help of a low point density LiDAR scanner placed at the top of the scanning vehicle. The success in creating accurate 3D profiles of the roads has made it possible to calculate the volume and depth of the potholes without using the data obtained from cameras. However, the problem with LiDAR data is that it is geometrically rich yet very expensive.

Moreover, Talha et al. [6] investigated further the implementation of LiDAR technology together with artificial intelligence algorithms to detect and measure potholes automatically. The algorithm provided accurate measurements regardless of texture and lighting, which is necessary for assigning priorities for maintenance operations. Nonetheless, as other LiDAR-based systems, cost limitations hinder its wide application in developing countries' roads.

B. Vision-Based Deep Learning Approaches

A Modified YOLOv8 has been proposed by Yurdakul & Tasdemir [2] with some modifications done to the architecture in order to enhance its sensitivity and localization capabilities for detecting potholes. The modified version has performed better in terms of precision rates than the basic variants of YOLOv8 on the benchmark dataset having high resolution images. This proves the suitability of YOLOv8 as a robust foundation for pothole detection.

However, Zhong et al. [3] suggested a combination of YOLOv8 with the point cloud information available through cameras. This solution not only allowed achieving high precision during object detection, but it also made it possible to estimate their size. The solution demonstrated better performance compared to single-modality baselines in case of occlusions, suggesting that it is an important way to go forward in future development of NPMSs.

Safyari et al.'s review [4] gave an extensive insight into pothole detection systems based on vision technology. The authors have reviewed conventional image processing, machine learning techniques, and deep learning models. As a result of the study, the authors stated that convolutional neural networks always provided more accurate results than conventional methods. At the same time, YOLO family methods proved to be the best compromise between accuracy and inference time.

Bhatt et al. [7] have given a detailed description and a comparative analysis of various approaches to pothole detection using acoustic, vibrational, visual, and LiDAR sensors. As a result of the comparative analysis, it was found that the deep learning approach, involving the use of camera sensors, and adequate training could provide equally good results to sensor fusion at a cheaper cost. According to Amali et al.'s [8] paper, ML was used for pothole detection and quantification.

C. Drivable Area and Scene Understanding

The research conducted on detection of potholes on drivable surfaces through deep learning [5] used semantic segmentation together with object detection to differentiate between drivable and non-drivable surfaces. It is an alternative way that can be considered for incorporation in the NPMS system in order to create an accurate drivable mask before the detection of potholes takes place.

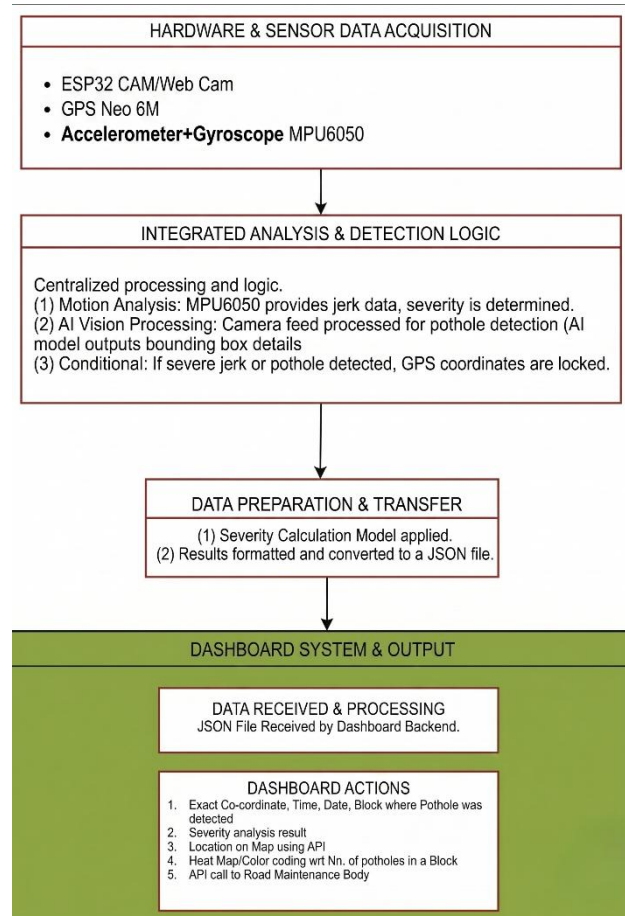
D. Positioning of NPMS

Although previous studies have individually examined various aspects of road surveillance, including detection precision, geometry estimation, sensor data fusion, none has provided a full-stack architecture utilizing low-cost off-the-shelf components for all these aspects. This is where NPMS shines, as it provides a fully functional framework, from video acquisition with ESP32-CAM through inference to the fusion of multiple sensors into structured CSV logs with GPS tagging.

III. SYSTEM ARCHITECTURE AND DATASET

A. Dataset Description and Annotation

The images in the NPMS data set have been collected through the ESP32-CAM road camera which captures images in a variety of driving scenarios, ranging from urban areas to highways to mixed areas. The images have also been taken under various weather conditions and during different hours of the day to increase the generalization capability of the model.



The annotation process was done following the YOLO annotation format, which means that the coordinates of the boxes will be annotated by the center-x, center-y, width, and height normalized values relative to the dimensions of the image, along with the class index of the object. There is only one detection class within the dataset, which is "pothole" (class index 0).

TABLE I. DATASET PARTITION STATISTICS

Subset	Images	Label Files	Proportion (%)
Training	623	623	70%
Validation	178	178	20%
Test	89	89	10%
Total	890	890	100%

The training dataset (70.0%) was utilized for computing the gradients required for updating the model weights using backpropagation. The validation dataset (20.0%) was evaluated after every epoch, and the model checkpoint that performed the best was selected. Finally, the test dataset (10.0%) was never exposed to training or validation procedures and served only to evaluate the model's performance on unseen road images.

B. Hardware Components

NPMS Hardware Architecture:

NPMS hardware components essentially comprise two primary building blocks. First, the ESP32-CAM module created by Espressif Systems. This is a relatively low-cost Wi-Fi-capable microcontroller module with the OV2640 camera module. In the Indian market, the ESP32-CAM module can be purchased for around ₹400 to ₹500. The ESP32-CAM module has an ability to stream MJPEG video over HTTP protocol on the `/stream` endpoint. This is used for pothole detection and inference operations. Moreover, there is another endpoint in the module referred to as `/sensors`. The `/sensors` endpoint streams telemetry data in JSON format, including GPS coordinate information and pitch and roll data.

Second, the inference host is necessary. The required capability of the host device in order to conduct pothole detection is the ability to use the Ultralytics YOLOv8 Python library. Possible inference hosts for the project include regular laptops, computers, or embedded AI devices such as Raspberry Pi computers with hardware acceleration. Currently, inference has been made using a CPU-based host computer. Nevertheless, it can be improved by using embedded AI platforms, such as NVIDIA Jetson Nano and Raspberry Pi 4 with Coral USB Accelerator, for higher performance and efficiency. Such devices have a potential to be used in intelligent transportation systems in India for more accurate and faster pothole detection. Embedded AI platform prices in the Indian market are typically in the range of approximately ₹6,000–₹15,000 based on the configuration and accessories provided.

C. Software Architecture

NPMS has been developed using a total of five major components each having their own unique roles within the system. Firstly, the Data Acquisition module, which involves the use of the capture.py script that gets the necessary frames from the video stream obtained from ESP32-CAM and displays the same to the user for creation of datasets. Secondly, the Model Training component uses the train_model.py tool which makes use of the Ultra YOLOv8 framework API according to the data.yaml parameters provided. Thirdly, there is the Inference Engine component where YOLOv8 takes in each frame for processing and output bounding box coordinates, labels and confidence scores. Fourthly, the Telemetry Fusion process obtains sensor data from the /sensors endpoint every two seconds after detection of potholes. Lastly, there is the Event Logger which logs all pothole event information in pothole_log.csv format.

IV. METHODOLOGY

A. Transfer Learning with YOLOv8

A current version of YOLO (You Only Look Once) architecture known as YOLOv8 has proven to be one of the most advanced object detectors that were developed by [Ultralytics](https://ultralytics.com?utm_source=chatgpt.com). Specifically, YOLOv8 architecture includes the decoupled detection head that enhances the speed of detecting objects. Namely, YOLOv8 architecture separates the detection of bounding box coordinates from class probability prediction, providing enhanced localization and classification abilities. Also, the model implements the anchor-free detection framework and is not reliant on any manual configuration of anchor boxes, thereby improving its robustness. Therefore, the implementation of YOLOv8 for pothole detection purposes can prove to be advantageous since potholes tend to be differently shaped.

As for the particular model among the YOLOv8 architectures, the proposed intelligent pothole monitoring system applies the YOLOv8 nano version (YOLOv8n). The reason for choosing this particular version of YOLOv8 is because of its small architecture of approximately 3.2 million of parameters that allow for the fast processing speed. The model relies on such techniques as depthwise separable convolution, C2f modules, and Spatial Pyramid Pooling Fast (SPPF) aggregation, providing high detection rate with low computation complexity. As a result, the nano version of the model is ideal for deployment on edge computing devices and embedded systems due to these characteristics, making YOLOv8n highly suitable for use in NPMSs.

For the efficient training of the pothole detector, the technique of transfer learning will be applied. At first, the pretrained weights will be downloaded from the COCO dataset using the `yolov8n.pt` model. Namely, the dataset includes more than 3,33,000 images divided into 80 different categories, which allows for generalization of visual features at the low- and mid-level. Thus, the application of pretrained weights considerably accelerates

the learning process since the model already possesses knowledge about pothole features. Moreover, since the dataset for the pothole detector contains a relatively small amount of images (890), the usage of transfer learning can help to avoid overfitting problems and save time during training.

B. Training Configuration

The full set of hyperparameters and configuration details are listed in Table II. The model was trained for 50 epochs using a batch size of 16 in the regular Ultralytics training process including data augmentation via mosaic, random flip, and HSV augmentations. The resolution of the input image was 640x640 pixels. There were no other augmentations used besides Ultralytics default augmentations.

TABLE II. YOLOV8N TRAINING CONFIGURATION

Hyperparameter	Value
Base Model	YOLOv8n (yolov8n.pt)
Image Size	640 x 640 px
Epochs	50
Batch Size	16
Optimizer	SGD (default)
Learning Rate	0.01 (default)
Device	CPU
Classes	1 (pothole)
Dataset Config	data.yaml
Augmentation	Mosaic, Flip, HSV (default)

C. Real-Time Inference Pipeline

Real-time inference is performed by the loop described in final_pothole_system.py and outlined below. Frames are continuously received via a MJPEG stream from the ESP32-CAM /stream endpoint with OpenCV VideoCapture. The frame is then analyzed by the YOLOv8n predictor returning an object of type Results, consisting of bounding box coordinates, class indices, and confidence scores. Bounding boxes whose confidence score is higher than the specified threshold (default 0.25) are further processed.

For every such bounding box, its severity proxy is defined as the pixel area enclosed by it, i.e., $(x_2 - x_1) * (y_2 - y_1)$. It's a very rough measure but at least gives some insight about relative pothole sizes in the picture frame. Simultaneously with the frame acquisition, a concurrent HTTP GET is made to /sensors endpoint to fetch the current telemetry values. Telemetry fields are used to populate an entry in the pothole_log.csv file.

In order to mitigate against any possible failures when acquiring the sensors readings, all such calls are enclosed in try-except statements with a timeout. In case no readings are available, placeholders ("null") are substituted in place. For visualization purposes, all bounding boxes are drawn on the frame together with an overlay showing GPS coordinates and vehicle attitude.

D. Severity Proxy and Event Logging Schema

Each instance of pothole detection is logged using the below schema in the output CSV file: (1) Timestamp - ISO 8601 datetime string using the clock time from the inference host system; (2) Latitude & Longitude - coordinate values in decimals from the GPS sensor; (3) Pitch - pitch angle of the vehicle in degrees using the IMU data, measuring the incline on the road surface; (4) Roll - roll angle of the vehicle in degrees using the IMU data, denoting any possible lateral tilt due to road unevenness; (5) Severity - bounding box pixel area for estimating pothole size.

V. EXPERIMENTAL RESULTS

A. Detection Performance Metrics

The model assessment procedure involved testing the trained neural network on the test dataset composed of 89 images after completing 50 training epochs. The object detection performance indicators were calculated after the end of training through the use of the evaluation script provided by Ultralytics.

TABLE III. YOLOV8N DETECTION PERFORMANCE ON TEST SET (EPOCH 50)

Metric	Value	Interpretation
Precision (B)	0.987	98.7% of detections are true potholes
Recall (B)	0.85	85% of actual potholes are detected
mAP@0.50 (B)	0.796	Strong detection at IoU = 0.50
mAP@0.50:0.95 (B)	0.505	Moderate localization across IoU range
F1-Score (est.)	0.91	Harmonic mean of Precision and Recall

The received value of precision equal to 0.987 implies that almost 98.7% of the detected bounding boxes are actually correct potholes located in the real dataset images. An increased value of precision means that the algorithm has low false-positive rates, thus reducing the probability of receiving unnecessary notifications regarding road maintenance in the developed pothole monitoring system.

Additionally, the model has received a high Mean Average Precision value for an intersection over union threshold of 0.50, which equals 0.796. This indicator can be considered the most important metric in terms of assessment of the pothole detector's detection capabilities. Apart from the aforementioned metric, the second one used for evaluation purposes is mAP@0.50:0.95 with the obtained value of 0.505. This metric assesses the performance of the object detection algorithm on different IoU thresholds from 0.50 to 0.95 with an increment of 0.05. Since increased values of this threshold require better object localization, it provides additional complexity to the assessment process.

B. Training Convergence

Loss curves for training (box loss, classification loss, and distribution focal loss) showed monotonically decreasing patterns through all 50 epochs without any signs of overfitting during validation. The best model checkpoint (best.pt), determined by validation mAP@0.50, was observed during epoch 50, implying that the model still shows continuous progress during training and potentially could be trained for more epochs using learning rate annealing.

The precision, recall, and mAP of the model were calculated after each epoch. Using the confusion matrix calculated via the Ultralytics pipeline revealed that false negatives (potholes missed) were the main errors made by the model, corroborating its recall rate of 0.85.

C. System Latency and Throughput

During the inference phase on a computer that utilizes an Intel Core i5 processor, the time taken to process each single frame of resolution 640×640 is 35-50 milliseconds. In other words, the model performs an inference of nearly 15-20 frames per second, showing that YOLOv8n is a lightweight object detection solution.

In addition to inference, another important component of the system is the acquisition of frames from the ESP32-CAM MJPEG video stream and the acquisition of telemetry from the sensors installed on board the vehicle. The performance is sufficient to implement a real-time pothole detection system during vehicle movement.

The NPMS proposed in this work is able to perform the task at a speed of 20 km/h and higher. With such vehicle speed, frames are acquired every 0.9-1.1 meters on average. This means that the distance between each subsequent image frame is sufficient to detect potholes with a diameter more than 0.5 meters in size.

TABLE IV. SYSTEM PERFORMANCE BENCHMARKS

Component	Latency / Throughput
YOLOv8n Inference (CPU)	20-30 ms/frame (~15-20 FPS)
ESP32-CAM Stream Capture	~5-10 ms/frame
Telemetry Polling (/sensors)	50-200 ms (network-dependent)
CSV Event Logging	<1 ms/event
End-to-End System FPS	~15-20 FPS
Effective Spatial Resolution	~0.9-1.1 m at 20 km/h

VI. KEY CONTRIBUTIONS AND APPLICATIONS

A. Novel Contributions

Contributions of the NPMS towards automated road monitoring include the following:

- End-to-End Pipeline: NPMS is, to the best of our knowledge, the first system to integrate customized YOLOv8 training, ESP32-CAM real-time streaming, GPS+IMU data fusion, and automated event CSV logging all into one.

- Affordable Hardware Design: Using the ESP32-CAM (Rs.550) with breakout GPS/IMU boards (~Rs.750), the NPMS proves that pothole detection of high accuracy is feasible without costly LiDAR and expensive industrial cameras.
- Geospatio-Temporal Event Data: Using a combination of GPS, IMU attitudes, and timestamped detections, the NPMS can produce events which can enable more advanced geospatial analytics than any camera-only or severity-based system alone.
- Severity-Tiered Detections: The area of detected pixels as an estimated severity of the detected object can be used as a basic triage for potholes for resource allocation purposes.
- Proof-of-Concept Dashboard: With the HTML dashboard, we prove that there is a potential to build a real-world dashboard that could be powered by backend databases and maps.

B. Practical Applications

NPMS architecture offers diverse opportunities for implementation of this technology. NPMS could be deployed by Municipal Road Maintenance organizations on their municipal inspection vehicles for continuous monitoring of the city roads, and creating GPS-based maps of potholes that require urgent repairing. NPMS devices can also be installed in buses, taxis, and other means of transportation by the Fleet Monitoring agencies to gather information about pothole events and create crowd-based road condition databases supported by objective photographic documentation. The NPMS can be deployed on road sections selected by the Preventive Maintenance Research organizations for longitudinal monitoring of pothole creation in relation to traffic loads, weather conditions, and prior road maintenance. NPMS route quality data can also be used by Navigation Systems to provide users with information about road surface quality and suggest better routes.

VII. LIMITATIONS

Nevertheless, despite the obvious strengths of NPMS, some limitations need to be mentioned to have an understanding of its present applicability properly:

Simplicity of Severity Estimation: The bounding-box pixel area calculation represents essentially a two-dimensional metric in an image plane and has no relation to the real depth and volume of a pothole. Moreover, its value highly depends on the installation height of the camera, focal distance, pitch angle, and even velocity of the car-*none* of which can be compensated for by the software itself.

Temporal Misalignment of Telemetry Data: Because of the nature of the asynchronous polling of the /sensors endpoint, there is a certain delay between receiving telemetry data from the camera and capturing an image. As a consequence, a pothole detection might get logged with old or default location coordinates.

Insufficient Dataset Diversity: Although 890 sample images were used to train the model and prove its efficiency, they are obviously not enough to ensure that it generalizes sufficiently well across all sorts of roads, illumination conditions, weather, and geographic locations. Thus, the system may fail when detecting potholes in unknown roads.

Network Stability Requirement: The present system architecture assumes a reliable connection between the ESP32-CAM and an inference service hosted online. Therefore, using the NPMS in areas with low connectivity will make the system unreliable. A local version of the inference algorithm would solve this problem.

Simulator Limitations: At the moment, the HTML application's intensity map works with randomized coordinates. The real implementation requires integration with `pothole_log.csv` and use of a geospatial web services API.

VIII. FUTURE WORK

There are several research directions that can be explored further based on the developed prototype:

Advanced Severity Model: In future work, camera calibration and depth estimation techniques (such as monocular depth estimation or stereo vision) can be used to convert the pixel area measurement into a centimeter dimension that gives actionable metrics for severity consistent with the road repair standards. Multi-class detection may be used with severity levels ranging from minor to severe.

Multi-Object Tracking and De-Duplication: Multi-object tracking algorithms (ByteTrack, BoT-SORT) can be implemented to track individual potholes in a sequence of video frames, thus removing duplication of logging and enabling accurate pothole count statistics by road segments.

Geospatial Clustering and Back-End: Using PostgreSQL back-end database configured for Geospatial data processing (PostGIS), events tagged with geolocation coordinates will be automatically clustered via R-tree spatial indexing. This allows building a dynamic web interface where live heatmaps of pothole density and downloadable maintenance requests can be provided.

Edge Deployment: Exporting the trained YOLOv8n model to ONNX or TensorRT formats and deploying it directly on NVIDIA Jetson Nano or Raspberry Pi equipped with Coral USB accelerator will allow running the application without Wi-Fi streaming from a vehicle-mounted monitoring device.

Dataset Extension and Data Augmentation: Adding more geographic locations, different types of roads, weather, and light condition data will improve model performance for pothole detection. Extending the dataset up to 5,000 or even 10,000 images may be accomplished with artificial data generation (GAN).

Federated Learning Approach: Federated learning approach would allow multiple NPMS-equipped vehicles to cooperatively train the pothole detector model while keeping image data confidential.

IX. CONCLUSION

It is worth mentioning that the proposed NPMS architecture represents a fully-fledged solution for automatic pothole detection and geolocation based on a dedicated single class YOLOv8 nano object detection model running alongside ESP32-CAM live video streaming and GPS-IMU data fusion for telemetry purposes. In other words, the suggested architecture represents an effective and affordable solution for intelligent surface monitoring and pavement maintenance applications.

In general, the results of testing a YOLOv8 nano-based pothole detection model trained on a custom dataset consisting of 890 road images show promising accuracy and efficiency levels. In particular, the proposed model reached such performance metrics as 0.987 precision, 0.85 recall, mAP@0.50, and 0.505 mAP@0.50:0.95. Based on the obtained results, one can claim that the developed single-class pothole detector shows sufficient performance even though it was trained on a relatively small number of examples. Thus, the proposed architecture proves to be effective for pothole monitoring in real time thanks to the YOLOv8 nano model's lightweight nature.

Additionally, the testing results for the overall performance of the suggested NPMS pipeline that includes GPS-based geotagging and CSV event logging show that it is possible to create an efficient pothole monitoring solution based on affordable commercial devices with a price point under Rs.3000 per piece. The affordability of the system allows suggesting it as an effective solution for large-scale deployment in developing countries like India since road management remains a challenging task there.

Finally, unlike most research papers that tend to analyze only object detection models, the suggested NPMS architecture offers a full-fledged framework with a set of practical structured outputs capable of being applied by municipal corporations and smart city departments. Namely, the structured event logs include detailed information about pothole detection in combination with the corresponding GPS coordinates and vehicle orientation parameters. Thus, with the use of the proposed NPMS system, it becomes possible to create more sophisticated road management solutions without relying on expensive specialized equipment. The next steps in improving the system should involve incorporation of more complex models for severity estimation, multi-object tracking, geospatial clusterization, and optimizations for edge deployment. Moreover, the open architecture and use of affordable commodity hardware make NPMS an ideal candidate for road condition monitoring solutions in developing countries.

REFERENCES

- [1] A. Faisal and S. Gargoum, "Cost-effective LiDAR for pothole detection and quantification using a low-point-density approach," *Transportation Research Record*, 2025.
- [2] M. Yurdakul and S. Tasdemir, "An enhanced YOLOv8 model for real-time and accurate pothole detection and measurement," *Expert Systems with Applications*, vol. 238, 2025.
- [3] J. Zhong et al., "YOLOv8 and point-cloud fusion for enhanced road pothole detection and quantification," *IEEE Transactions on Intelligent Transportation Systems*, 2025.
- [4] Y. Safyari et al., "A review of vision-based pothole detection methods using computer vision and machine learning," *Sensors*, vol. 24, no. 12, 2024.
- [5] Chandra S, "Pothole detection in drivable area using deep learning," *Proc. IEEE Int. Conf. on Intelligent Transportation*, 2025.
- [6] S. A. Talha et al., "The use of LiDAR and artificial intelligence algorithms for detection and size estimation of potholes," *Journal of Infrastructure Systems*, ASCE, 2024.
- [7] A. K. Bhatt et al., "Advancements in pothole detection techniques: A comprehensive review and comparative analysis," *IEEE Access*, vol. 13, 2025.
- [8] S. Miruna Joe Amali et al., "ML-driven detection and quantification of potholes for safer roads," *Applied Sciences*, vol. 15, 2025.
- [9] G. Jocher et al., "Ultralytics YOLOv8," *GitHub Repository*, 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [10] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," *Proc. ECCV*, pp. 740-755, 2014.
- [11] Espressif Systems, "ESP32-CAM Product Specification," *Espressif Systems Datasheet*, 2024.