Road Surface Damage Detection with Frame-Level Synchronized Localization on a Jetson Embedded Platform using YOLO

Pham Van Khoa*, Le Quang Nhat Nam, Pham Tran Minh

Author's address: Ho Chi Minh City University of Technology and Engineering, Vietnam

**Corresponding author. Email: khoapv@hcmute.edu.vn*

Received: dd/mm/yyyy; Revised: dd/mm/yyyy;

Accepted: dd/mm/yyyy; Published: dd/mm/yyyy

ABSTRACT

Real-time pothole detection is important for road safety and preventive maintenance, yet practical deployment remains challenging when inspection systems must operate continuously on resource-constrained embedded hardware. Existing studies have reported promising detection accuracy, but fewer works address the combined issues of real-time edge inference, reliable geo-referencing, and train-deploy mismatch in fixed dashcam settings. This study presents a deployment-oriented pothole detection framework for the NVIDIA Jetson Orin Nano using YOLOv8n as a lightweight detector. Rather than introducing a new detection architecture, the proposed system focuses on deployment efficiency through TensorRT FP16 acceleration, on-frame geo-tagging, and perspective-aligned data construction. A customized dataset of 1,410 dashcam images with consistent camera-to-road geometry was used for training. During inference, GNSS coordinates embedded directly in video frames were extracted using OCR, enabling frame-synchronized geo-tagging without separate sensor synchronization. Experimental results showed that the model achieved 83% mAP@0.5, 43% mAP@0.5:0.95, 88% precision, and 70% recall. On the Jetson Orin Nano, the optimized pipeline sustained 102.34 FPS with 82.1% GPU usage, 63.1% CPU usage, 10.4 W power consumption. These results indicate that a deployment-aligned pipeline can provide a practical balance between detection performance, computational efficiency, and geo-referenced road monitoring.

Keywords: *Pothole detection, NVIDIA Jetson Orin Nano, real-time road monitoring, intelligent transportation systems, TensorRT optimization.*

1. INTRODUCTION

Road transportation infrastructure plays a critical role in socio-economic development, urban mobility, and public safety. However, pavement surfaces are continuously exposed to repeated traffic loading, water infiltration, and environmental stress, which accelerate deterioration and lead to defects such as cracks and potholes. In traffic environments dominated by motorcycles and other two-wheeled vehicles, even relatively small potholes can pose substantial safety risks by destabilizing vehicles and increasing accident likelihood. Therefore, timely and reliable pothole detection is an important prerequisite for preventive maintenance, road safety improvement, and intelligent road asset management [1], [2].

Recent advances in deep learning and computer vision have significantly improved the feasibility of automated road inspection. Deep neural networks have demonstrated strong representation capability in visual recognition, while single-stage detectors have enabled real-time object detection in practical environments [1], [3]-[6]. Among these approaches, the YOLO family has become particularly influential because it offers a favorable speed–accuracy trade-off, making it attractive for transportation-related vision systems that must operate outside controlled laboratory settings [3]-[6]. YOLOv8n, in particular, is a compact variant designed for efficient inference, and is therefore a practical candidate for embedded deployment [6].

These advances have stimulated extensive research on pothole and road-damage detection. Earlier studies showed that low-cost vision-based inspection can support scalable road-defect monitoring. In particular, iWatchRoad demonstrated that a vehicle-mounted dashcam-based pipeline can support large-scale pothole detection, geotagging, and geospatial visualization for smart-city applications [2]. More recent studies have explored lightweight detectors for mobile and embedded platforms, including compact road-damage networks, YOLO-based pothole detectors, and improved one-stage models designed to reduce computational complexity while preserving real-time capability [7]-[10], [14].

Despite these developments, the transition from benchmark-oriented model development to stable field deployment remains challenging. High offline detection accuracy alone is insufficient when a system must operate continuously on resource-constrained edge hardware under realistic thermal, power, and latency constraints. For this reason, model compression and deployment-oriented optimization strategies, including pruning, quantization, and reduced-precision inference, remain highly relevant for practical embedded vision systems [11]-[12].

Another unresolved issue concerns localization reliability. In many road-monitoring pipelines, visual data and GNSS data are acquired through separate asynchronous channels, which may introduce temporal drift and spatial misalignment between detected potholes and recorded coordinates. Prior OCR studies on numerical character recognition suggest a possible basis for extracting embedded numeric information directly from image frames, although direct evidence for dashcam GNSS-overlay extraction remains limited [13]. This observation motivates the use of on-frame telemetry extraction as a practical alternative to external sensor synchronization when location data are already embedded in the video stream.

A further challenge lies in train–deploy mismatch. Many public pothole datasets contain images captured from highly heterogeneous viewpoints, camera heights, and environmental conditions. Although such diversity is useful for benchmarking, it may reduce deployment stability when the target system is intended for a fixed forward-facing dashcam mounted on a moving vehicle. Under these conditions, large viewpoint variation between training images and the target deployment setting may reduce detection stability in real-world operation [2], [14].

Motivated by these limitations, this study proposes a deployment-oriented pothole detection framework for the NVIDIA Jetson Orin Nano. Rather than introducing a new detection architecture, the study emphasizes practical deployment through three coordinated design choices: a lightweight YOLOv8n detector, TensorRT FP16 acceleration on embedded hardware, and an on-frame geo-tagging mechanism in which GNSS telemetry embedded directly in each video frame is extracted through OCR. In addition, the study adopts a perspective-aligned dataset construction strategy based on a consistent dashcam geometry in order to reduce train–deploy mismatch. Collectively, these components are intended to support efficient embedded inference, frame-level association between detections and coordinates, and improved deployment relevance in real-world road-monitoring scenarios.

The main contributions of this work are threefold. First, it develops a real-time pothole detection pipeline centered on YOLOv8n and optimized for embedded inference on the Jetson Orin Nano. Second, it proposes an on-frame geo-tagging strategy that preserves frame-level correspondence between visual detections and location data through OCR-based extraction of embedded GNSS telemetry. Third, it adopts a deployment-oriented dataset design with perspective-consistent dashcam imagery to better match the intended operational setting. These contributions aim to narrow the gap between pothole-detection research and practical edge-based road-monitoring systems for intelligent transportation applications.

2. RELATED WORK

Prior studies on pothole and road-damage detection can be broadly organized into four related directions: scalable visual road inspection, lightweight detection models, model compression and embedded acceleration, and practical deployment pipelines integrating localization or temporal association. This structure is more appropriate for deployment-oriented pothole monitoring than a purely model-centric taxonomy, because the main challenges extend beyond detection accuracy alone.

The first direction concerns scalable road inspection using low-cost visual sensing. iWatchRoad showed that dashcam-based pothole monitoring can support large-scale detection and geospatial visualization in urban environments, thereby illustrating the practical value of vehicle-mounted vision systems for smart-city infrastructure monitoring [2]. This line of work established the feasibility of combining visual inspection with spatial reporting, but it did not by itself resolve the broader challenges of efficient embedded inference, deployment stability, or frame-synchronized localization under fixed dashcam conditions.

The second direction focuses on lightweight road-damage detection models. YOLO-LWNet demonstrated that reduced-complexity architectures can still achieve competitive road-damage detection performance on resource-limited platforms [7]. Subsequent studies further explored pothole-specific or improved YOLO-based detectors, including real-time YOLOv8 pothole detection, OBC-YOLOv8, and edge-oriented road-damage detection variants [8]-[10]. In addition, enhanced lightweight road-damage networks have been proposed to improve feature extraction while maintaining compactness [14]. Collectively, these studies support the suitability of one-stage and lightweight detectors for real-time road-defect analysis, especially when deployment efficiency is a primary concern.

A third direction addresses acceleration strategies for embedded deployment. The broader literature on deep-network compression has established pruning, quantization, and reduced-precision inference as effective approaches for reducing computational cost and improving execution efficiency on resource-constrained hardware [11], [12]. These methods are highly relevant to pothole detection systems intended for continuous edge operation, even though the compression studies themselves are not specific to road defects. Related work has also discussed knowledge distillation as a potential route toward lighter pothole-detection models [15]. However, this citation is better interpreted as a conceptual or theoretical direction rather than conclusive experimental evidence of deployment-ready performance. Overall, these studies highlight the importance of system efficiency, but many remain focused on computational savings or model design rather than full deployment behavior under realistic operating conditions.

A fourth direction concerns deployment realism, including sensing geometry, localization, and temporal consistency. OCR-based numerical recognition methods indicate that numeric metadata embedded in image content can be extracted under suitable imaging conditions [13]. Although this does not directly validate dashcam GNSS-overlay extraction, it provides a methodological basis for using OCR to recover frame-level telemetry when coordinates are rendered directly in the video stream. Beyond OCR, some studies have investigated UAV-based road-defect detection and transformer-enhanced detection-and-tracking frameworks [16], [17]. These approaches are technically valuable, but their sensing geometry, viewpoint diversity, and operational assumptions differ substantially from those of a fixed forward-facing dashcam mounted on a moving vehicle. As a result, they do not directly resolve the deployment mismatch encountered in perspective-consistent edge-monitoring scenarios.

Temporal association and embedded video analytics have also received growing attention. Deep association tracking methods such as Deep SORT provide an efficient mechanism for maintaining object continuity across frames [18]. Related video-analytics studies combining improved YOLO-based detection with BoT-SORT-style tracking further illustrate the feasibility of integrating detection and temporal association in practical pipelines [19]. In addition, prior work on NVIDIA Jetson-based object tracking suggests that lightweight vision models can be deployed on embedded GPU platforms for real-time operation [20]. Nevertheless, these studies generally prioritize tracking continuity or throughput and do not explicitly address precise pothole geo-localization through frame-synchronized metadata extraction.

Overall, the literature supports the feasibility of scalable road inspection[2], lightweight road-damage detection [7]-[10], [14], model compression and acceleration for embedded inference [11], [12] and embedded video analytics with temporal association [18]-[20]. It also suggests that OCR-based extraction of embedded numeric metadata may provide a useful basis for tighter coupling between visual detections and location records [13]. However, few studies combine real-time embedded pothole detection, frame-synchronized geo-tagging, and perspective-aligned training data within a single unified framework tailored to a fixed dashcam deployment environment. The present study addresses this gap by integrating TensorRT-accelerated YOLOv8n inference on the Jetson Orin Nano with OCR-based on-frame geo-tagging and a deployment-oriented dataset designed to better match the target operating geometry.

3. DESIGN METHODOLOGY

3.1 System Architecture

The proposed pothole detection system adopts a two-stage architecture consisting of model training and edge-based inference, as illustrated in Figure 1. This separation between model development and deployment enhances system modularity and facilitates efficient adaptation to resource-constrained edge platforms.

In the training stage, a customized dataset representing real-world dashcam driving conditions is collected and annotated with bounding boxes for pothole instances. The dataset is used to fine-tune the YOLOv8n object detection model to accurately recognize potholes under diverse environmental conditions such as varying illumination, motion blur, and road textures. After training, the model is converted into a hardware-optimized inference engine using TensorRT to accelerate execution on the target edge device.

In the inference stage, the optimized model is deployed on the NVIDIA Jetson Orin Nano platform. Dashcam video streams containing GPS information overlaid directly on the frames are processed in near real time. The system performs pothole detection, generates bounding boxes with associated confidence scores, and links each detection to the corresponding GPS coordinates extracted from the same frame, thereby enabling geo-tagged reporting of road defects. The detailed runtime control logic, including temporal subsampling, visualization persistence, and OCR-based metadata extraction, is presented in Figure 2 and Section 3.4.

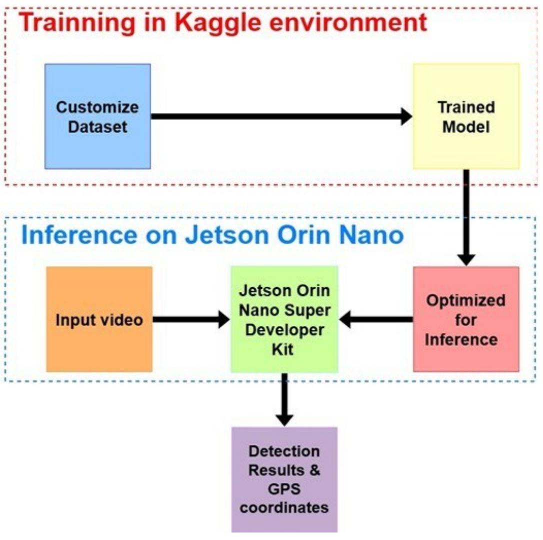

Trainning in Kaggle environment
Customize Dataset
Trained Model
Inference on Jetson Orin Nano
Input video
Jetson Orin Nano Super Developer Kit
Optimized for Inference
Detection Results & GPS coordinates

Figure 1. Block diagram of the proposed system

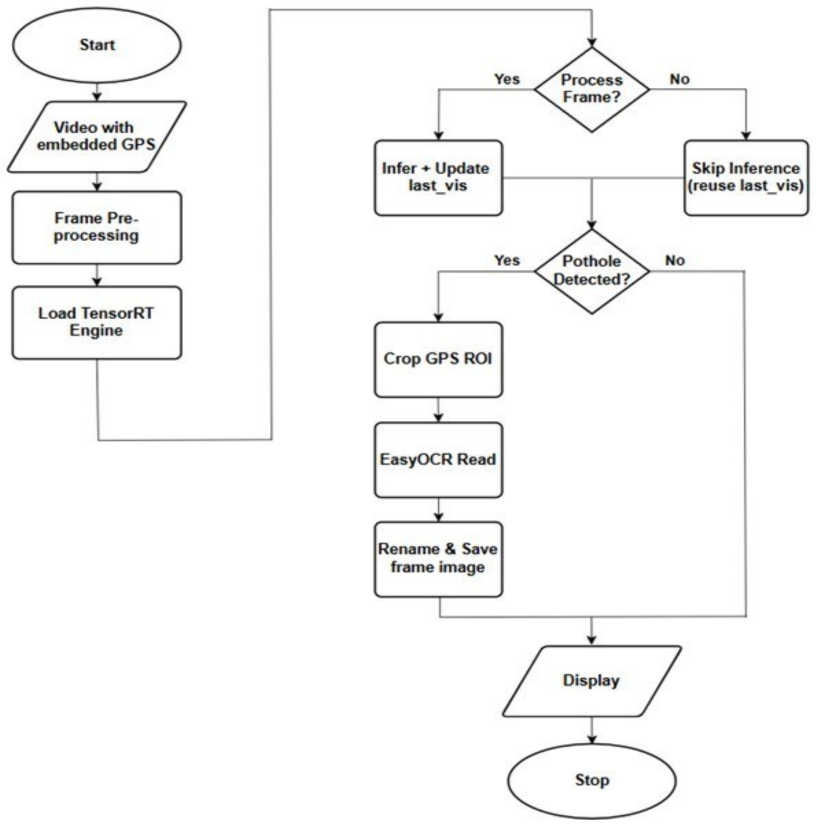


Figure 2. Proposed GPS-enabled pothole detection framework on the Jetson Orin Nano with frame skipping and EasyOCR-based evidence extraction.

Figure 2 presents the end-to-end deployment workflow for GPS-enabled pothole detection on the Jetson Orin Nano. In contrast to the high-level architecture, this workflow specifies the runtime control logic used to balance detection accuracy, throughput, and automated evidence archiving. The system first ingests a video stream containing GPS telemetry overlaid on each frame, after which the frames are resized and normalized to 640×640 for model inference. The TensorRT-optimized YOLOv8n-FP16 engine is then initialized. To improve throughput, inference is governed by a temporal subsampling mechanism implemented through the "Process Frame?" decision step. When the condition is met, the system performs full inference; otherwise, it bypasses heavy computation and reuses the previous visualization to preserve smooth playback. A notable component of the workflow is the conditional data-extraction branch activated when a pothole is detected. In the absence of a detection, the system proceeds directly to display, thereby minimizing latency. Upon a positive detection, however, the pipeline crops the GPS text region, extracts the coordinate strings using EasyOCR, and uses the recovered latitude and longitude values to generate a structured filename for evidence storage. The workflow then displays the annotated frame containing both bounding boxes and embedded GPS information before proceeding to the next cycle.

3.2 Dataset Preparation

To mitigate the domain shift often encountered when transitioning from training to edge deployment, this study constructs a deployment-oriented dataset tailored to real-world dashcam conditions in Vietnam. Unlike public datasets characterized by highly heterogeneous camera viewpoints, our customized dataset comprises 1,410 road-surface images with a consistent camera-to-road geometry, approximately a 45° viewing angle as shown in Figure 3.

To assess dataset suitability, we compared publicly available pothole datasets with our customized deployment-oriented dataset. Many open-source pothole datasets, including the AndrewMVD Pothole Detection dataset and the Angga DwiSunarto Potholes-Detection-YOLOv8 dataset [21], [22] exhibit substantial variation in pothole size, road-scene appearance, and camera viewpoint, ranging from elevated or near-ground views to vehicle-mounted perspectives. Although such diversity is useful for general benchmarking, it may introduce a pronounced domain gap when the model is ultimately deployed in a fixed dashcam setting on a moving vehicle. In this context, viewpoint inconsistency may reduce detection stability and increase missed detections.

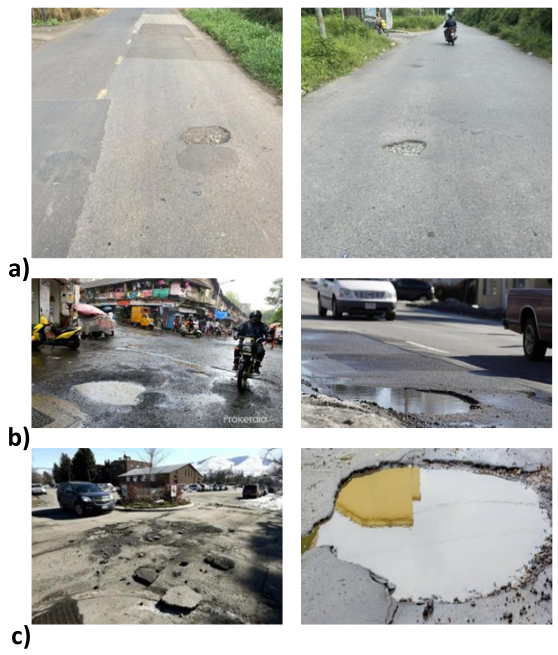


Figure 3. Representative pothole image samples from the datasets used in this study: (a) the customized dataset, collected under a fixed dashcam configuration with relatively consistent pothole scale; (b) the AndrewMVD Pothole Detection dataset, showing greater variation in pothole size and road-scene appearance [21]; and (c) the Angga DwiSunarto Potholes-Detection-YOLOv8 dataset, showing further variation in viewpoint, acquisition conditions, and object scale [22].

We developed a localized dataset of 1,410 images for three main reasons. First, perspective consistency was maintained through an approximately fixed 45° viewing angle, allowing the model to learn features that better reflect the visual geometry of a standard dashcam. Second, the dataset provides environmental relevance by capturing characteristics of Vietnamese road infrastructure, including local asphalt textures, repair patterns, and lighting conditions that are often underrepresented in public datasets. Third, the dataset was designed to better match the intended deployment environment, thereby reducing train–deploy mismatch and improving the practical relevance of the trained model for fixed dashcam operation.

3.3 Pothole Detection Model

As illustrated in Figure 4, the pothole detection model is trained through a structured pipeline consisting of dataset preparation, preprocessing, fine-tuning, validation, and best model selection. This workflow is intended to obtain a detector that remains sufficiently accurate for pothole recognition while preserving practical suitability for deployment on resource-constrained edge devices.

The process begins with the customized dataset, which is designed to reflect real-world dashcam conditions while maintaining consistent camera viewpoints, image resolution, and road-surface context. During the preprocessing stage, all annotations are converted into the YOLO format and organized into a standardized directory structure with a fixed input resolution, reducing unnecessary variability in the training data.

Next, the YOLOv8n model is fine-tuned from pretrained weights to adapt the network to pothole detection. Leveraging pretrained representations allows the model to inherit general visual knowledge while accelerating convergence on the road-surface domain. Throughout training, periodic validation is performed on a separate validation set to monitor convergence trends and detect potential overfitting.

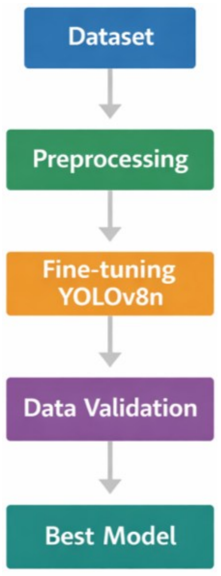


Figure 4. Workflow for training the YOLOv8n pothole detection model.

Finally, the model achieving the best validation performance across all training epochs is selected as the final model for evaluation and deployment. This pipeline is intended to obtain a lightweight detector that preserves practical pothole-detection accuracy while remaining suitable for embedded execution.

3.4 Edge Optimization and Geo-tagging Pipeline

Efficient deployment on resource-constrained edge devices requires inference optimization, temporal subsampling, and a practical geo-tagging mechanism. In this study, the trained YOLOv8n model was exported from PyTorch to ONNX and compiled into a TensorRT inference engine optimized for the Jetson Orin Nano's Ampere GPU architecture. FP32 parameters were converted to FP16 to improve Tensor Core utilization and reduce memory bandwidth consumption, while TensorRT applied kernel-level optimizations such as layer fusion to minimize execution overhead and increase inference throughput. Since adjacent frames in high-frame-rate driving videos contain considerable temporal redundancy, processing every frame is inefficient. Accordingly, the system employs a temporal subsampling strategy that performs inference selectively when a pothole is expected to remain visible within the camera field of view.

We define the detection opportunity window as the number of frames during which a pothole remains visible within the effective detection distance, D. If the vehicle travels at speed v (km/h) and the input video frame rate is FPS, the visibility duration is given by

$$T_{vis} = \frac{D}{V_{m/s}}, V_{m/s} = \frac{v}{3.6}$$

Thus, the number of available frames in which detection can occur is

$$N_{frame} = FPS * T_{vis} = \frac{3.6 * D * FPS}{v} \quad (1)$$

where v is the vehicle speed (Km/h), FPS is the input video frame rate, and D is the effective detection distance determined by the camera field of view (FOV).

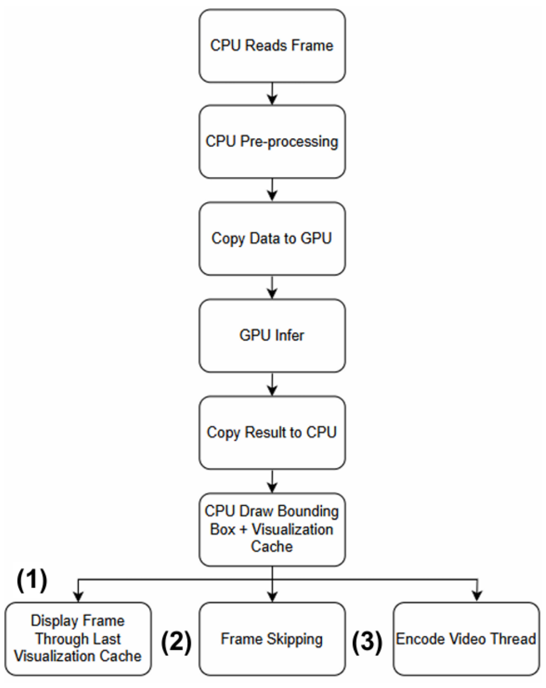


Figure 5. Parallel inference pipeline with temporal subsampling and visualization persistence.

Based on this formulation, the system performs inference only on selected frames while reusing detection results during skipped frames through a Visualization Persistence mechanism. As illustrated in Figure 5, bounding box coordinates are temporarily stored and reused to maintain stable visual output without flickering. Meanwhile, an asynchronous thread performs video encoding to prevent I/O operations from blocking the main inference loop. To eliminate synchronization errors commonly found in external GPS logging systems, the proposed method directly extracts location information embedded within the video frames. When a pothole is detected with confidence above a predefined threshold, the system crops a Region of Interest (ROI) containing the GNSS telemetry overlay. The ROI undergoes binarization to suppress background noise and enhance character clarity before being processed by the EasyOCR engine. The extracted latitude and longitude values are then used to generate a structured filename, automatically linking each detected pothole with its corresponding geospatial coordinates for subsequent analysis.

4. RESULTS AND DISCUSSION

To assess the effectiveness of the proposed pothole detection system, comprehensive experiments were performed on a customized deployment-oriented dataset. The training process was evaluated using multiple performance indicators, including loss functions, precision–recall metrics, and mean Average Precision (mAP). These metrics provide valuable insight into both the model's convergence behavior and its final detection performance. Figure 6 presents the evolution of the key evaluation metrics during training. The mAP@0.5 curve increases steadily and stabilizes after approximately 250 epochs, indicating progressive improvement in the model's localization capability. Likewise, the mAP@0.5:0.95 metric shows gradual convergence, reflecting enhanced bounding-box regression accuracy across multiple IoU thresholds. The precision and recall curves also improve consistently throughout training, eventually reaching approximately 88% and 70%, respectively. These results suggest that the model achieves a balanced trade-off between accurate pothole detection and the suppression of false positives.

The convergence behavior of the loss functions is illustrated in Figure 7. The training and validation losses for bounding-box regression (Box Loss), classification (Cls Loss), and distribution focal loss (DFL Loss) all decrease consistently as training progresses. The close alignment between the training and validation curves indicates that the model generalizes well without significant overfitting. Notably, the validation losses stabilize after approximately 300 epochs, suggesting that the model has reached a stable point of optimization.

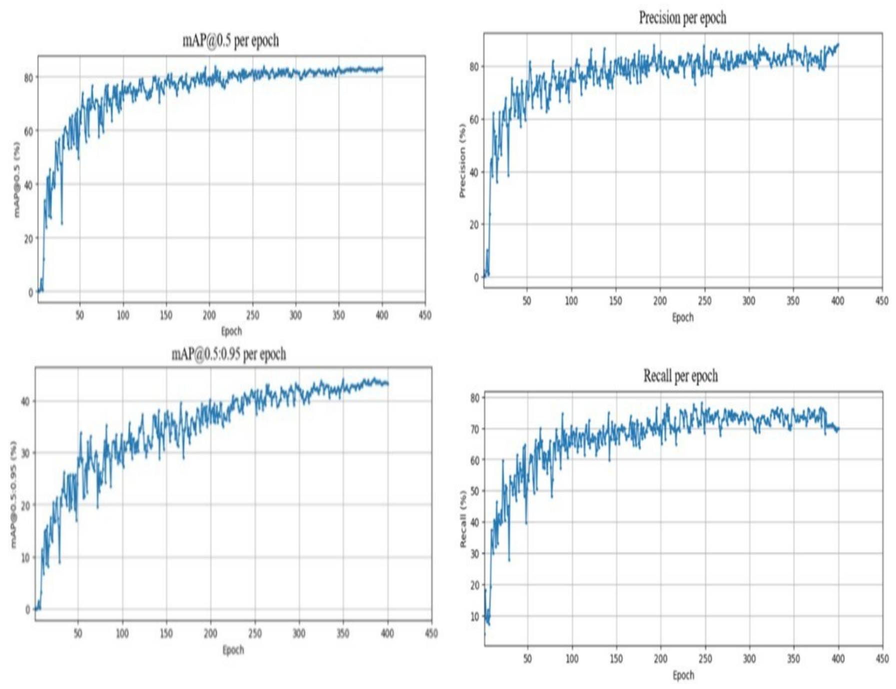


Figure 6. Model performance metrics across training epochs, including mAP, precision, and recall.

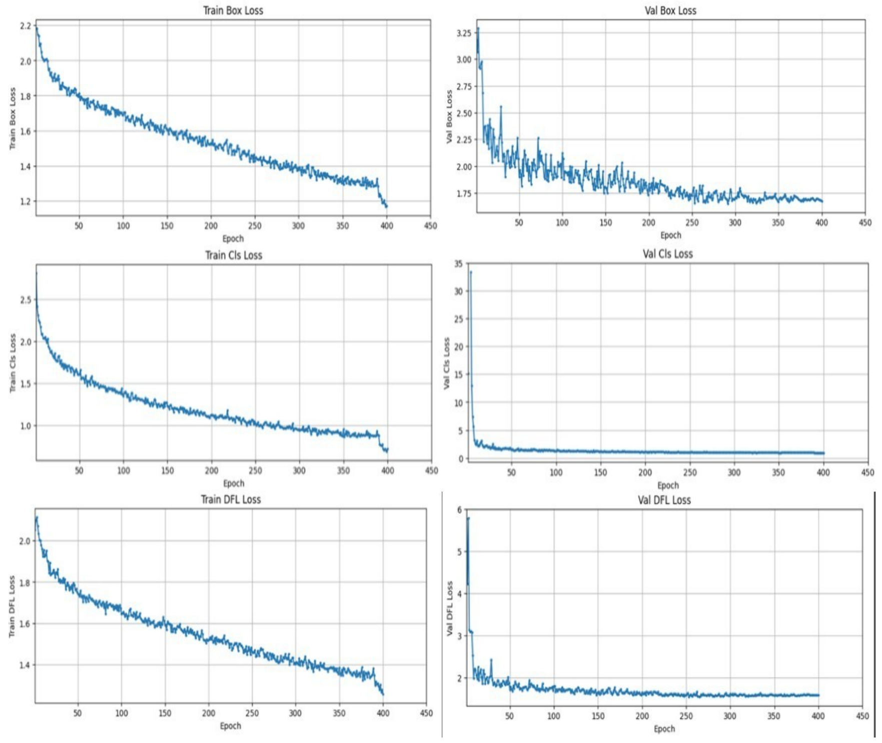


Figure 7. Training and validation losses for bounding box regression, classification, and distribution focal loss.

Table 1. Quantitative performance of the YOLOv8n pothole detector on the customized dataset.

Metric	Value
mAP@0.5	83%
mAP@0.5:0.95	43%
Precision	88%
Recall	70%

Based on the best-performing checkpoint, Table 1 summarizes the final detection performance of the YOLOv8n model on the customized pothole dataset. The model achieved 83% mAP@0.5, 43% mAP@0.5:0.95, 88% precision, and 70% recall, indicating relatively high precision with acceptable recall under the tested conditions.

Figure 8 presents representative qualitative detection results under varied illumination, pavement texture, and traffic conditions. The examples show that the detector can localize potholes across different road environments, including both single-defect and multi-defect scenes. In most cases, the predicted bounding boxes are reasonably aligned with the damaged regions, while the confidence scores vary according to defect visibility, contrast, and boundary clarity. More challenging cases, such as shadowed surfaces, irregular textures, and low-contrast potholes, tend to produce lower confidence values, although the system remains capable of detecting the relevant defects. Overall, these observations indicate that the proposed framework remains effective under realistic operating conditions.

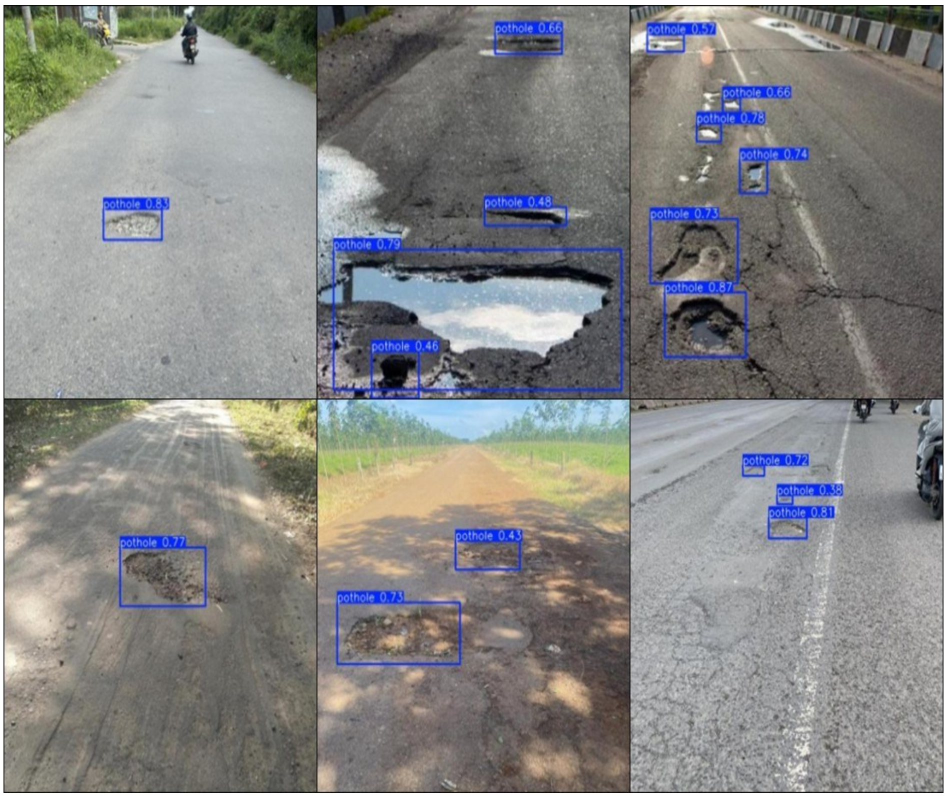
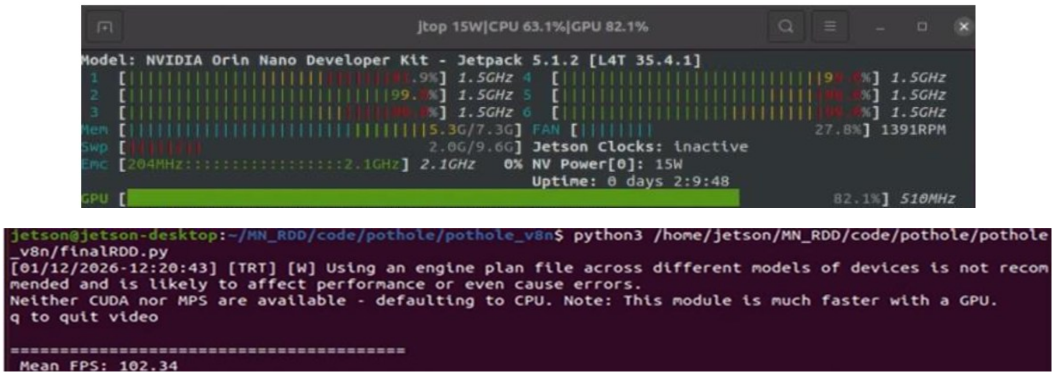


Figure 8. Qualitative pothole detection results under diverse traffic and illumination conditions.

The operational efficiency of the proposed system was assessed under continuous inference on the NVIDIA Jetson Orin Nano. Under the tested deployment setting, the optimized pipeline sustained a throughput of 102.34 FPS. Table 2 summarizes the measured resource utilization, including 82.1% GPU load, 63.1% CPU load, 10.4 W power consumption, and an average temperature of 49°C. Figure 9 presents the corresponding runtime evidence collected from the deployed system.



```
jtop 15W|CPU 63.1%|GPU 82.1%
Model: NVIDIA Orin Nano Developer Kit - Jetpack 5.1.2 [L4T 35.4.1]
 1  [||||||||||||||||||||||||||||||||[illegible].9%] 1.5GHz 4  [|||||||||||||||||||||||||||||||9[illegible]%] 1.5GHz
 2  [|||||||||||||||||||||||||||||||99.[illegible]%] 1.5GHz 5  [||||||||||||||||||||||||||||||[illegible]%] 1.5GHz
 3  [||||||||||||||||||||||||||||||[illegible]%] 1.5GHz 6  [||||||||||||||||||||||||||||||[illegible]%] 1.5GHz
Mem [|||||||||||||||||||||||||||||||||||5.3G/7.3G] FAN [||||||||            27.8%] 1391RPM
Swp [||||||||                   2.0G/9.6G] Jetson Clocks: inactive
Emc [204MHz:::::::::::::::::::2.1GHz] 2.1GHz    0% NV Power[0]: 15W
                                                 Uptime: 0 days 2:9:48
GPU [                                                           82.1%] 510MHz
```

```
jetson@jetson-desktop:~/MN_RDD/code/pothole/pothole_v8n$ python3 /home/jetson/MN_RDD/code/pothole/pothole
_v8n/finalRDD.py
[01/12/2026-12:20:43] [TRT] [W] Using an engine plan file across different models of devices is not recom
mended and is likely to affect performance or even cause errors.
Neither CUDA nor MPS are available - defaulting to CPU. Note: This module is much faster with a GPU.
q to quit video

========================================
 Mean FPS: 102.34
```

Figure 9. Runtime console output illustrating the measured inference throughput on the Jetson Orin Nano.

Table 2. System resource utilization during continuous operation

Resource Metric	Measured Value
Inference Throughput (FPS)	102.34
GPU Load (%)	82.1
CPU Load (%)	63.1
Power Consumption (W)	10.4
Average Temperature (◦C)	49

As shown in Table 2, the proposed system sustained a throughput of 102.34 FPS during continuous operation, with GPU and CPU loads of 82.1% and 63.1%, respectively. The observed resource distribution is consistent with a hybrid workload that includes video decoding, inference, OCR-related processing, and file management. In addition, the system operated at a power consumption of 10.4 W and an average temperature of 49°C under the tested setting. These results indicate that the proposed pipeline is practically suitable for real-time edge-based pothole monitoring on the Jetson Orin Nano.

Figure 10 presents the final on-device output, where potholes are marked with bounding boxes and the GPS telemetry embedded in the frame is retained for geo-tagging. In the illustrated example, the deployed system detected two potholes within a single field of view, with confidence scores of 0.74 and 0.66, both exceeding the predefined threshold of 0.60. The higher confidence of the larger pothole is consistent with its clearer boundaries and stronger visual features, whereas the smaller defect remained detectable despite its lower contrast. This result indicates that the proposed framework can perform reliable multi-instance pothole detection while maintaining frame-level location information for geo-referenced reporting.

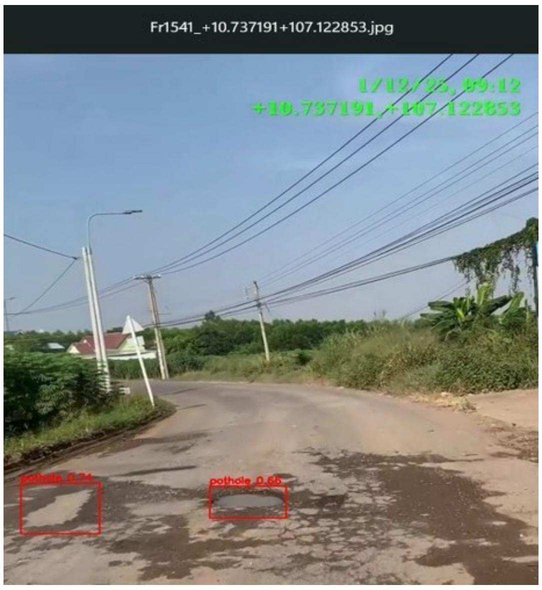


Figure 10. On-device visual output showing pothole detections together with embedded GPS telemetry used for frame-synchronized geo-tagging.

To further assess the practical utility of the geo-tagging mechanism, a field experiment was conducted along a predefined road trajectory. The extracted GPS coordinates were automatically recorded and mapped onto a geographic information system (GIS) interface. As illustrated in Figure 11, the resulting spatial distribution map visualizes the locations of detected potholes along the route. The mapped markers suggest that the proposed pipeline can transform visual detection results into structured geographic information that may support road maintenance and infrastructure monitoring.

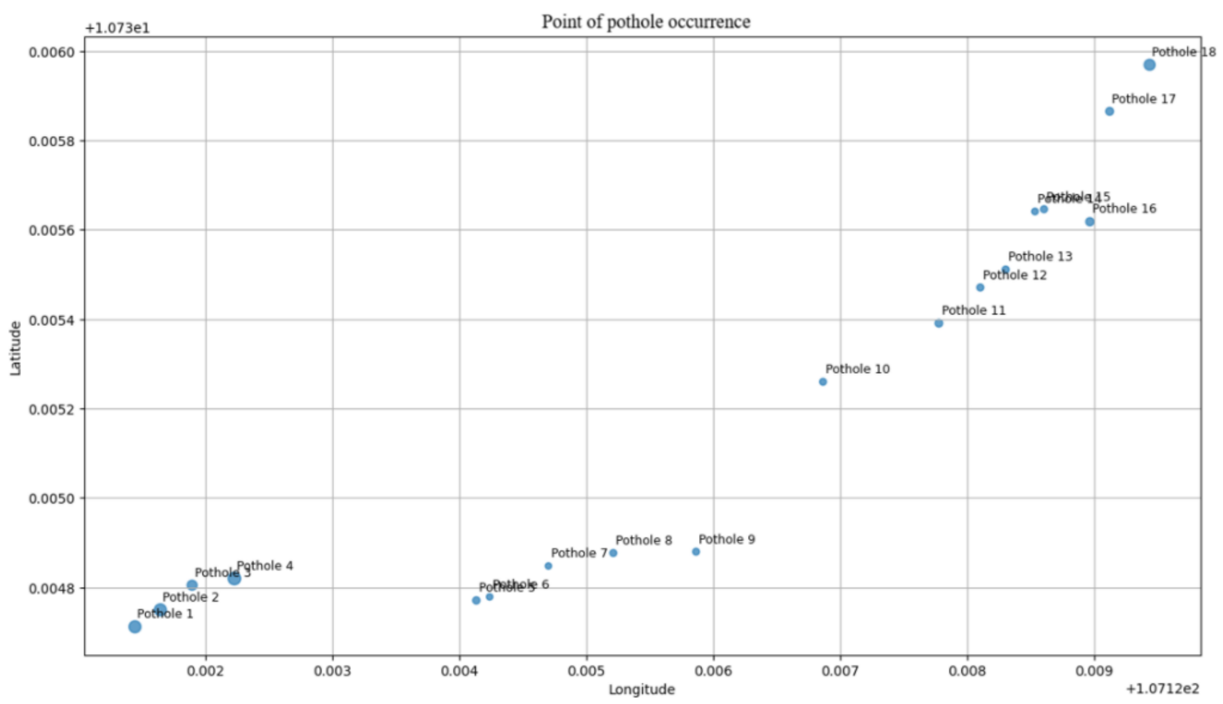


Figure 11. Geospatial distribution of detected potholes along the test trajectory.

Experimental observations suggest that system performance is influenced by both pothole morphology and operating conditions. While the model achieved high precision, localization accuracy as measured by mAP@0.5:0.95 remained limited by the irregular shape and diffuse boundaries of potholes, especially relative to more structured defects such as cracks. Vehicle speed was also a critical factor. The proposed frame-skipping strategy performed most effectively in the range of 40–65 km/h, where sufficient temporal density was maintained to provide at least seven frames per defect. At speeds exceeding 80 km/h, motion blur and sparse sampling reduced the effective detection window, shifting the main limitation from computational capacity to optical sensing constraints. In such cases, disabling temporal subsampling may improve reliability by restoring temporal redundancy.

5. CONCLUSION

This study presented a deployment-oriented pothole detection framework for real-time road monitoring on the NVIDIA Jetson Orin Nano. By integrating YOLOv8n, TensorRT FP16 acceleration, OCR-based on-frame geo-tagging, and perspective-aligned dashcam data, the proposed system was designed to address practical deployment constraints rather than benchmark accuracy alone. On the customized dataset, the model achieved 83% mAP@0.5, 43% mAP@0.5:0.95, 88% precision, and 70% recall, while the deployment pipeline reported a throughput of 102.34 FPS with moderate power consumption and stable operating temperature under the tested setting. The results further suggest that embedding GPS telemetry directly into video frames can support frame-level association between detections and location records. The deployment-oriented dataset design may also help reduce train–deploy mismatch in fixed dashcam scenarios, although a dedicated comparative study would be needed to quantify that effect more rigorously. Nevertheless, performance remained sensitive to challenging road textures, irregular pothole boundaries, and OCR reliability under adverse imaging conditions. Overall, the proposed framework demonstrates the practical feasibility of integrating lightweight pothole detection, embedded acceleration, and frame-synchronized geo-tagging within a single edge-based road-monitoring pipeline.

DECLARATION OF CONFLICTS OF INTEREST

The authors declare that this research has no conflicts of interest.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] R. R. Sahoo, S. S. Mohanty, and S. Mishra, “iWatchRoad: Scalable detection and geospatial

visualization of potholes for smart cities," *arXiv* preprint arXiv:2508.10945, 2025.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR*), 2016, pp. 779–788.

[4] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," *in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR*), 2017, pp. 7263–7271.

[5] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv* preprint arXiv:2004.10934, 2020.

[6] G. Jocher, A. Chaurasia, and J. Qiu, YOLOv8 Documentation. *Ultralytics*, 2023.

[7] C. Wu, M. Ye, J. Zhang, and Y. Ma, "YOLO-LWNet: A lightweight road damage object detection network for mobile terminal devices," *Sensors*, vol. 23, no. 6, Art. no. 3268, 2023.

[8] S. P. Thadakamalla and S. Babu K., "Real-time pothole detection using convolutional neural networks and YOLOv8*," International Journal on Science and Technology*, vol. 16, no. 3, p. 6854, 2025.

[9] S. Zhang, Z. Liu, K. Wang, W. Huang, and P. Li, "OBC-YOLOv8: An improved road damage detection model based on YOLOv8," *PeerJ Computer Science*, vol. 11, Art. no. e2593, 2025.

[10] H. Mahmudah, A. S. Aisjah, S. Arifin, and C. A. Prastyantyo, "iYOLOV7-TPE-SS: Leveraging improved YOLO model with multilevel hyperparameter optimization for road damage detection on edge devices," *IEEE Access*, vol. 13, pp. 79247–79263, 2025.

[11] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," *in Proc. 4th International Conference on Learning Representations* (ICLR), 2016.

[12] T. Liang, J. Glossner, L. Wang, S. Shi, and X. Zhang, "Pruning and quantization for deep neural network acceleration: A survey," *Neurocomputing*, vol. 461, pp. 370–403, 2021.

[13] P. H. Jain, V. Kumar, J. Samuel, S. Singh, A. Mannepalli, et al., "Artificially intelligent readers: An adaptive framework for original handwritten numerical digits recognition with OCR methods," *Information*, vol. 14, no. 6, Art. no. 305, 2023.

[14] H. Luo, C. Li, M. Wu, and L. Cai, "An enhanced lightweight network for road damage detection based on deep learning," *Electronics*, vol. 12, no. 12, Art. no. 2583, 2023.

[15] A. Musa, M. Hamada, and M. Hassan, "A theoretical framework towards building a lightweight model for pothole detection using knowledge distillation approach," *in Proc. 4th ETLTC International Conference on ICT Integration in Technical Education* (ETLTC), 2022.

[16] W. Zhang, X. Li, L. Wang, D. Zhang, P. Lu, et al., "A lightweight method for road defect detection in UAV remote sensing images with complex backgrounds and cross-scale fusion," *Remote Sensing*, vol. 17, no. 13, Art. no. 2248, 2025.

[17] N. Wang, L. Shang, and X. Song, "A transformer-optimized deep learning network for road damage detection and tracking," *Sensors*, vol. 23, no. 17, Art. no. 7395, 2023.

[18] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," *in Proc. IEEE International Conference on Image Processing (ICIP*), 2017, pp. 3645–3649.

[19] Y. Yang, D. Pi, L. Wang, M. Bao, J. Ge, et al., "Based on improved YOLOv8 and Bot SORT surveillance video traffic statistics," *Research Square preprint*, rs-4161504/v1, 2024.

[20] A. Selberg, "Generic object tracking with NVIDIA Jetson Nano using Siamese convolutional neural networks," *M.S. thesis, Lund University*, Lund, Sweden, 2020.

[21] AndrewMVD, Pothole Detection, Kaggle dataset. [Online]. Available: Kaggle.

[22] A. D. Sunarto, Potholes-Detection-YOLOv8, Kaggle dataset. [Online]. Available: Kaggle.