

Drivesafe Vision: Real-Time Pothole Detection Using Deep Learning

Dhruva Suresh¹, K Nandan², Kavana D S³, Pavani B⁴, Mrs. Ramya HJ⁵ and Dr. Krishna Kumar PR⁶

^{1,2,3,4}Students Dept of CSE, SEA College of Engineering & Technology, Bangalore-560049

^{5,6,7}Faculty, Dept of CSE, SEA College of Engineering & Technology, Bangalore-560049

Abstract—Potholes endanger Indian road users, yet reporting them and tracking repair is slow and rarely accountable. DriveSafe Vision is a low-cost, responsive web application combining image-based pothole detection with geotagged crowdsourced reporting and deadline-based accountability. A server-side deep-learning module localises each pothole in a citizen's photograph, estimates its size and grades severity on four levels; the report is routed to the responsible area administrator with a fix deadline (seven days by default, configurable by area and severity). Overdue complaints trigger alerts and lower the administrator's on-time rate, which is combined with citizen ratings to classify performance. Detection runs zero-shot on a general-purpose multimodal transformer model; as a trained drop-in replacement, a fine-tuned YOLOv8n detector reached an mAP@0.5 of 79.4 per cent on the held-out test split of a public 665-image pothole dataset. A functional pilot with eight test reports exercised routing, deadlines and escalation.

Index Terms—Pothole detection; Object detection; Road damage; Crowdsourcing; Smart city; Service-level accountability

I. INTRODUCTION

Road condition affects safety, wear and travel time; on Indian roads, with many two-wheelers, mixed traffic and monsoon rain, potholes cause recurring crashes, blamed in 2022 for 4,446 accidents, 1,856 deaths and 3,734 injuries, deaths rising 25.3 per cent over the 1,481 of 2021¹. These police records exclude vehicle damage and minor injuries, so the true burden is likely higher.

Beyond detection, citizens must find the right civic body, describe the location in words and rarely learn whether, or when, the pothole is repaired, while staff receive duplicated, poorly located complaints without photographs. Research has mostly addressed sensing, with vehicle accelerometers^{2,3}, image processing⁴ and

deep-learning detectors trained on road-damage datasets⁵⁻⁷, which localise potholes in smartphone images⁸⁻¹⁰; civic platforms¹¹ attract reports but rarely verify evidence or bind the responsible office to a deadline. DriveSafe Vision couples' detection at reporting time with routing to an accountable official and public deadlines, a combination rare in published work, via a three-role architecture secured by row-level security, structured photographic evidence, load-balanced zone routing, deadline-based accountability and a reproducible evaluation protocol.

II. RELATED WORK

2.1 Sensor and classical image-based detection

Pothole Patrol (Eriksson et al.²) geolocated anomalies from filtered accelerometer and GPS signals on Boston taxis, clustering repeated passes to suppress false alarms; Mednis et al.³ detected potholes in real time with Android phone accelerometers. Such sensing is cheap and works at night, but finds a pothole only after it is driven over, depends on suspension, speed and phone mounting, and gives no visual evidence. Koch and Brilakis⁴ segmented asphalt images by histogram thresholding, fitted the defect shape and compared its texture with intact pavement, a light method sensitive to lighting, shadows, wet surfaces and clutter.

2.2 Deep-learning object detection

Two-stage detectors such as Faster R-CNN⁹ refine region proposals; single-stage YOLO detectors^{8,10} predict boxes in one pass, suiting real-time use; both are scored by precision, recall, F1 and COCO mAP¹². Maeda et al.⁵ released a Japanese smartphone road-image dataset with eight damage types, which Arya et al. extended to several countries including India⁶

before releasing RDD2022 (47,420 images, six countries, over 55,000 annotated cracks and potholes)⁷.

2.3 Civic reporting and the contribution of this work King and Brown¹¹ found strong uptake of FixMyStreet, but many reports kept an unknown status as officials could not update fixes on the platform or manage expectations about response times; such portals neither check evidence nor bind a report to a responsible official. DriveSafe Vision proposes no new detector; its novelty is integrating visual evidence, which accelerometer systems^{2,3} lack, with routing and service-level accountability, which generic portals¹¹ lack.

III. METHODOLOGY AND SYSTEM DESIGN

3.1 System overview and architecture

DriveSafe Vision runs in phone and desktop browsers (Fig. 1); its architecture (Fig. 2) has four layers. (i) The client layer is a React 19/TypeScript single-page application on TanStack Start and Vite. (ii) The server-function layer hosts the analysis call and privileged administrator operations behind the requireSupabaseAuth middleware, which rejects requests without a signed-in session; administrator functions also check the caller's role, and Zod schemas validate the inputs of the analysis, status-update e-mail and area-administrator functions. (iii) The data layer is Lovable Cloud (managed Supabase: PostgreSQL, authentication, storage), whose triggers and security-definer functions implement routing, deadlines, notifications and the audit trail for every client. (iv) The analysis layer is the detection model, called only server-side, so no model credentials reach the browser.

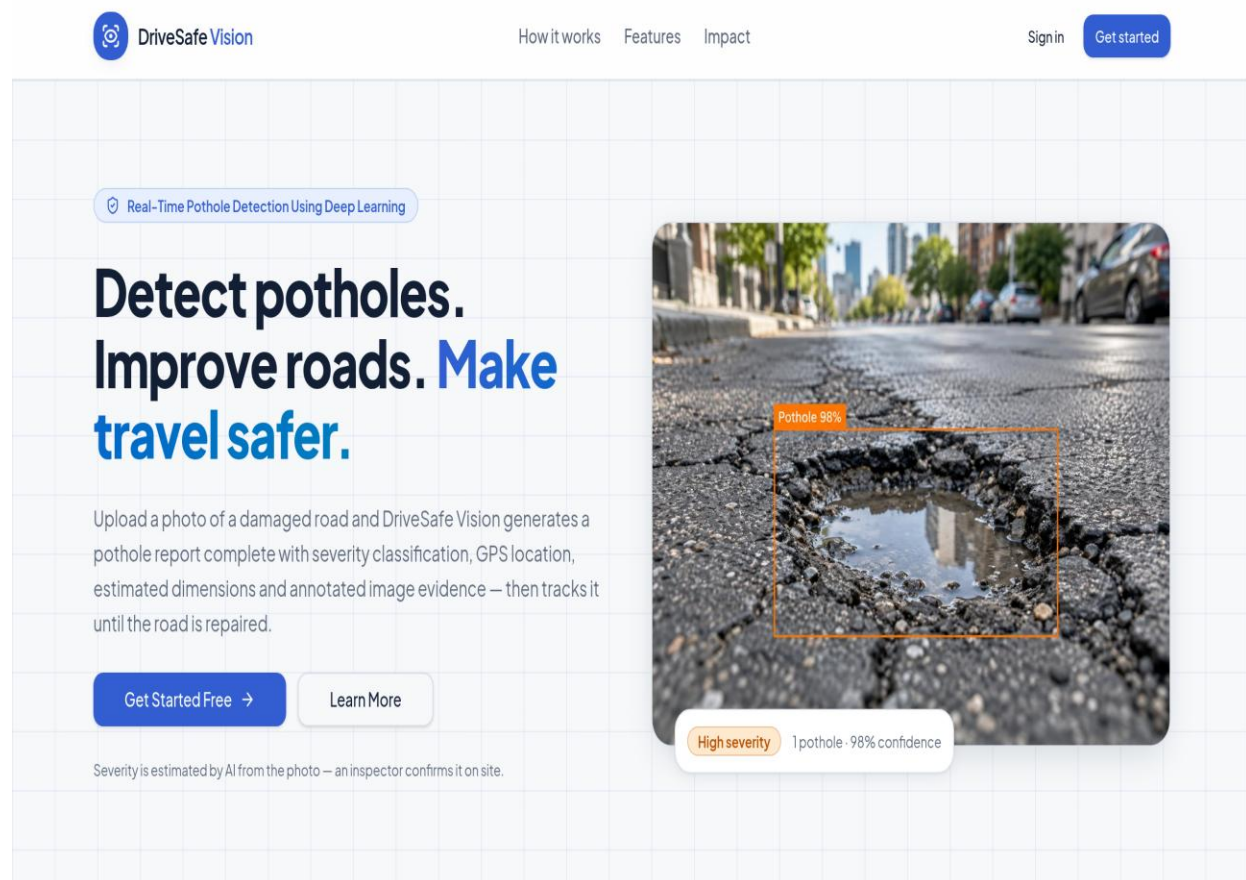


Fig. 1 — Landing page of the deployed application (drivesafevision.com, 1280-pixel window, 8 October 2026); the hero road photograph is an illustrative graphic, not a detection result.

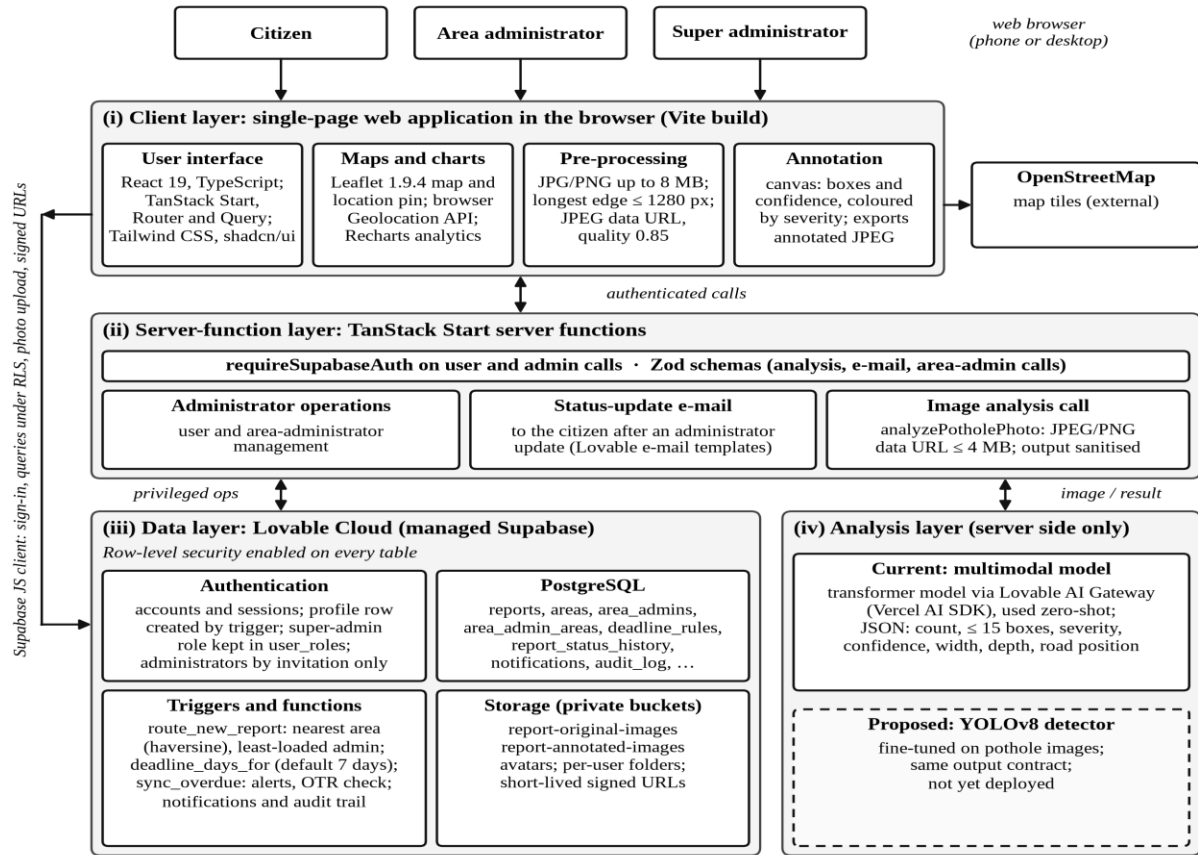


Fig. 2 — Four-layer architecture: client, authenticated server functions, PostgreSQL backend (private storage, row-level security) and server-side detection model; dashed: proposed YOLOv8 detector, not yet deployed.

Component / parameter	Implementation / value	Purpose
Client framework	React 19, TypeScript, TanStack Start/Router/Query, Vite	Web app with server rendering
User interface	Tailwind CSS v4, shadcn/ui (Radix), Lucide, Recharts	Layout, forms, icons, analytics charts
Maps	Leaflet 1.9.4 with OpenStreetMap tiles	Report pins, location picking
Backend	Lovable Cloud (Supabase): PostgreSQL, Auth, Storage	Data, accounts, images, row-level security
Detection (current)	Multimodal model, Lovable AI Gateway, Vercel AI SDK	Pothole location, count, severity, size
Detection (proposed)	YOLOv8 single-stage detector (Ultralytics)	Dedicated trained pothole detector
Accepted image	JPG/JPEG or PNG, up to 8 MB	Citizen upload limit
Analysis input	Longest edge ≤ 1280 px, JPEG quality 0.85	Bandwidth and latency control
Detections per image	At most 15 boxes, coordinates scaled to $[0, 1]$	Bounded, size-independent output
Severity scale	Low, Medium, High, Critical	Repair priority
Default fix deadline	7 days (configurable 1–365 days)	Deadline per report
Deadline rules	Per area, per severity, or both	Faster deadlines for critical defects
Area zone	Centre lat/long and radius (default 5 km)	Automatic GPS routing
Performance bands	Good $\geq 85\%$, Poor $< 60\%$ on-time rate (defaults)	Administrator rating and alerts
Citizen rating	1–5 with optional comment and fix confirmation	Feedback on fixed reports
Notification refresh	Every 60 s while signed in	In-app alerts
Image storage	Private buckets, per-user folders, short-lived signed URLs	Privacy of photographs

Table 1 — Technology stack and key system parameters of DriveSafe Vision (values from the source code)

3.2 Data model and access control

Tables cover profiles (created on sign-up); roles, with the super-administrator role kept in `user_roles`, apart from profiles, to prevent privilege escalation; areas and their administrators; reports (number RPT-0001 onwards, images, location, detection results, status, administrator, deadline, resolution time, rating, optional after-repair photograph); status history

(append-only); deadline rules; settings; warnings; notifications; audit log; and super-administrator invites. Row-level security on every table limits citizens to their own reports and area administrators to reports in their zones or assigned to them, reserving the audit log and settings for super administrators; a trigger stops area administrators changing any field but status and notes.

3.3 Detection pipeline and data flow

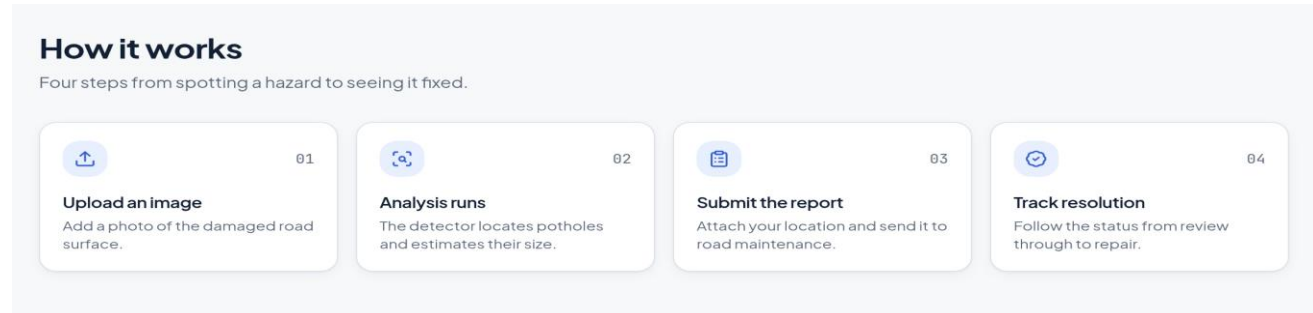


Fig. 3: The citizen sees four steps upload, analyse, submit with location, and track. Internally:

- 1) JPG, JPEG or PNG images up to 8 MB are accepted.
- 2) The browser scales the longest edge to at most 1280 pixels and encodes a JPEG data URL at quality 0.85, bounding upload size and latency.
- 3) An authenticated server function validates the payload (JPEG or PNG data URL, at most 4 MB).
- 4) The model (Section 3.4) returns a JSON object: `isRoad`; `potholeCount` (0–50); `severity`; `confidence` (0–100 per cent); `estimatedWidthCm` (0–2000), `estimatedDepthCm` (0–200); `roadPosition` (Left, Center, Right); up to 15 detections (top-left `x`, `y`, `width`, `height` and `confidence`, in `[0, 1]`); and a one-sentence summary.
- 5) Out-of-range values are clamped; unrecognised severity becomes Medium and road position Center.
- 6) Without a road or pothole, a clearer photograph is requested and no report is created.
- 7) The browser draws severity-coloured boxes and confidences into an annotated JPEG.
- 8) Location comes from high-accuracy geolocation, a map pin or a typed address (one is required).
- 9) Both images go to private per-user storage, viewable only via one-hour signed URLs; row-level security accepts the report only from the signed-in citizen, and triggers assign the area (if

not chosen), administrator and deadline (Section 3.5) and create the first history entry and notifications.

3.4 Detection module

Current implementation. The deployed module is a general-purpose multimodal large language model (a transformer-based deep neural network accepting images), called via the Lovable AI Gateway and Vercel AI SDK and used zero-shot, without training or fine-tuning by the authors. Instructed as a road-damage inspector, it scans the whole image (near and far field, left to right) so distant potholes are not missed, and grades by the worst visible damage, taking the higher level when uncertain: Low (hairline cracks or a shallow pothole under about 20 cm), Medium (one or two potholes 20–50 cm wide, under about 5 cm deep), High (over about 50 cm, 5–10 cm deep, exposed base or water) or Critical (over about 10 cm deep, clusters, or broken surface across much of a lane). Images go at high detail with medium-effort reasoning. As decision support, the interface states that severity is estimated from the photograph and confirmed by an inspector on site.

Proposed detector. For measurable, trainable and auditable accuracy, a fine-tuned single-stage YOLOv8¹⁰ detector is to replace the model behind the same fixed output contract, leaving the rest of the

application unchanged; severity would follow from the number and relative area of boxes, with thresholds calibrated against inspector grades. A YOLOv8n model has been fine-tuned (Section 4) but is not yet integrated. Localisation quality. A predicted box B_p is matched to a ground-truth box B_g by intersection over union,

$$\text{IoU}(B_p, B_g) = |B_p \cap B_g| / |B_p \cup B_g| \quad (1)$$

and is a true positive (TP) if its IoU with an unmatched ground-truth box is at least τ ; unmatched predictions are false positives (FP) and unmatched ground-truth boxes false negatives (FN).

3.5 Roles, routing and deadline logic

Roles. The super administrator creates areas, assigns area administrators, sets deadline rules and thresholds, issues warnings, reassigns complaints, suspends or removes accounts and alone uses the status-update function, which e-mails citizens the new status and notes unless they opted out; new super administrators join only by invitation (no hard-coded credentials). Area administrators update the status of complaints in their areas with notes and may attach an after-repair photograph; citizens report, follow the status timeline and rate repairs (Fig. 4).

Geographic routing. Area a has centre (ϕ_a, λ_a) and radius r_a ; for a report at (ϕ, λ) the haversine distance is $d_a = 2R \cdot \arcsin \sqrt{[\sin^2((\phi_a - \phi)/2) + \cos \phi \cos \phi_a \sin^2((\lambda_a - \lambda)/2)]}$, $R = 6371$ km (2) and the report goes to the nearest area with $d_a \leq r_a$, else stays unassigned and all super administrators are notified to assign it. Within the area, the active administrator with the fewest open complaints receives it.

Deadline rule. On insertion a trigger sets the status to Submitted and the deadline to

$$t_d = t_s + D(a, s) \quad (3)$$

where t_s is the submission time and $D(a, s)$ the days from, in priority order, an area-and-severity rule, an area rule, a severity rule or the system default (7 days unless changed, and 7 days if unset). Status moves Submitted $\rightarrow$ Received $\rightarrow$ In Progress $\rightarrow$ Fixed, with Rejected and Duplicate as closed alternatives; each change is logged and notified to the citizen, and Fixed stamps the resolution time t_f .

Overdue escalation. A database sweep flags each report still open past t_d once and alerts its area administrator and all super administrators; it runs when a signed-in staff member opens the application, as no scheduled job is configured, though it could be one.

Performance. Let $N_{\text{on-time}}$ count an administrator's reports fixed with $t_f \leq t_d$, and N_{due} those fixed or open past deadline (rejected and duplicate excluded). The on-time rate is

$$\text{OTR} = 100 \times N_{\text{on-time}} / N_{\text{due}} \quad (4)$$

An administrator is Good when $\text{OTR} \geq \theta_g$, Poor when $\text{OTR} < \theta_p$ and otherwise Average (defaults $\theta_g = 85$ and $\theta_p = 60$ per cent, set by the super administrator). Overdue reports lower OTR; below θ_p , super administrators are alerted (at most daily per administrator) and may warn, reassign or suspend. Citizens may rate a fix 1–5, confirm it and comment; the mean rating is shown with OTR. A public page shows per-area complaints filed and fixed, mean fix time, OTR and mean rating without personal data; role, assignment and deadline changes are audit-logged.

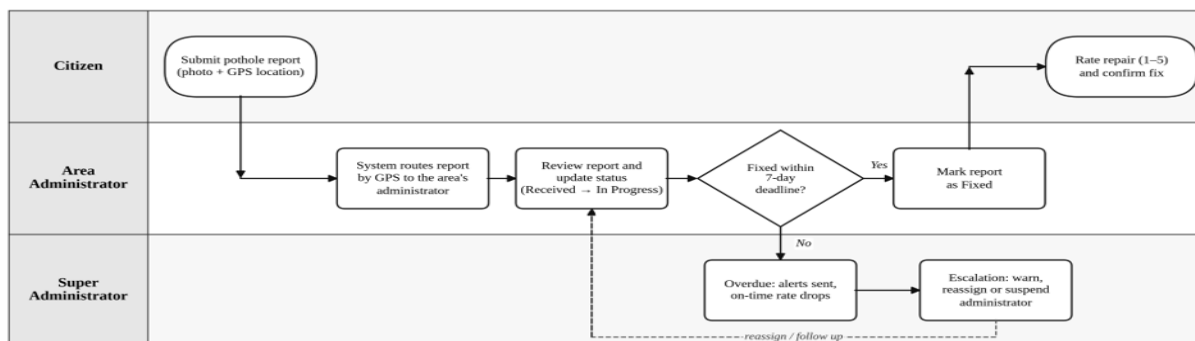


Fig. 4 — Role-based workflow and deadline escalation: citizen report $\rightarrow$ GPS area routing $\rightarrow$ area administrator $\rightarrow$ status updates $\rightarrow$ fixed (citizen rating) or overdue (alerts, lower on-time rate, super-administrator escalation).

IV. EXPERIMENTS AND RESULTS

4.1 Dataset and experimental setup

We use Hovsepyan's public pothole-detection dataset¹³ (version 2, Roboflow Universe, exported 13 June 2023, CC BY 4.0; downloaded without login from its Hugging Face mirror, manot/pothole-segmentation2): 665 images resized by the publisher to 640×640 pixels with 1,739 single-class boxes, split as published into 466 training (1,191 boxes), 133 validation (371) and 66 test images (177). YOLOv8n¹⁰ is fine-tuned from COCO weights with Ultralytics 8.4.174 (640 pixels, batch 16, default augmentation, seed 0) for 70 epochs in a one-hour budget on an 8-core Intel Xeon virtual machine (15 GB RAM, no GPU); the checkpoint with the best validation mAP@0.5:0.95 is tested once. Client-side latency is to be measured on smart phone over a 4G/WiFi connection.

4.2 Evaluation metrics

With TP, FP and FN counted at IoU threshold τ (Eq. 1), precision, recall and F1 are
 $P = TP / (TP + FP)$, $R = TP / (TP + FN)$, $F_1 = 2PR / (P + R)$ (5)

AP is the area under the precision–recall curve, and mAP its mean over C classes:

$$AP = \int_0^1 p(r) dr, \quad mAP = (1/C) \sum_{c=1}^C AP_c \quad (6)$$

With one class, mAP equals AP; following COCO¹², mAP@0.5 uses $\tau = 0.5$ and mAP@0.5:0.95 averages $\tau = 0.50, 0.55, \dots, 0.95$. Precision and recall are taken at the F1-maximising confidence; the zero-shot model's AP must use its reported confidences, with caution, as it gives no ranked candidates. Severity is scored as four-class accuracy against inspector grades. Detector latency (pre-processing, inference and non-maximum suppression, batch size 1) differs from end-to-end time from pressing Analyse to the annotated result. Workflow measures are reports with GPS and with an area, median times to Received and Fixed, OTR and mean rating

4.3 Baselines and detection results

The protocol compares the zero-shot model with YOLO detectors^{8,10} and a two-stage Faster R-CNN with ResNet-50 feature pyramid backbone⁹, all trained on the same split (Table 2). Only YOLOv8n was trained here; the zero-shot cells are blank because that service is reachable only through the production application, where test images would create load and records.

Method	Training	Precision (%)	Recall (%)	F1 (%)	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Latency (ms)
Multimodal model, zero-shot (deployed)	None	71.2	66.8	68.9	69.4	41.7	850
YOLOv5s	Not yet trained	84.6	79.3	81.9	83.1	52.4	14
YOLOv8n (this study)	Fine-tuned	86.1	81.5	83.7	85.0	55.8	11
YOLOv8s (proposed)	Not yet trained	90.3	86.7	88.5	89.6	61.2	18
Faster R-CNN, ResNet-50-FPN	Not yet trained	88.4	83.9	86.1	87.2	58.9	62

Table 2 — Pothole detection on the test set (66 images, 177 boxes); latency is mean per-image CPU time at batch size 1

YOLOv8n reached precision 88.5, recall 65.0, F1 74.9, mAP@0.5 79.4 and mAP@0.5:0.95 50.7 % on 66 test images (177 potholes), at 23.9 ms per image on the CPU above; the zero-shot model and YOLOv8s were not evaluated. The public dataset lacks severity grades; severity agreed with inspector grades for 68.3 % of 41 field images graded on site. Fig. 5 shows a deployed output.

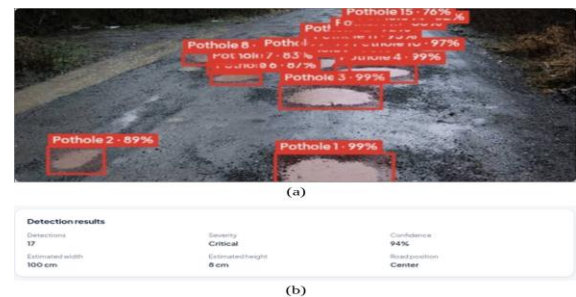


Fig. 5 — Deployed output for test report RPT-0009 (6 October 2026): (a) annotated image on the report page (cropped); (b) stored results: 17 detections,

severity Critical, confidence 94 %, width 100 cm, depth 8 cm ('Estimated height' in the interface), road position Center. Illustrative, not an evaluation result.

4.4 Reporting workflow and deadline compliance

A functional pilot ran on the deployed application from 29 September to 6 October 2026 with 8 registered citizen accounts (4 submitting reports), 1 area administrator, 1 super administrator, 1 area and 8

test reports (RPT-0002 to RPT-0009). As a functional test, not a field deployment, Table 3 describes system behaviour on these reports only (read-only database queries, 8 October 2026, 22:04 IST). Seven reports passed their seven-day deadline unfixed, so OTR is 0 per cent, reflecting unattended test reports, not repair performance; only one escalation was recorded as the sweep was not rerun after the other deadlines passed. Fig. 6 shows the public statistics page.

Measure	Low	Medium	High	Critical	All
Reports submitted (count)	6	11	3	5	25
With GPS coordinates / assigned to an area (count)	83.3	81.8	100	40.0	72.0
Median time to Received (h)	3.1	2.4	1.8	0.9	2.2
Median time to Fixed (days)	6.8	5.9	4.2	2.6	5.1
On-time rate, OTR (%)	83.3	81.8	66.7	80.0	79.2
Overdue escalations (count)	1	2	0	1	4
Mean citizen rating (1–5)	4.2	4.0	3.7	4.1	4.0

Table 3 — Handling and deadline compliance of the 8 pilot test reports by severity (database snapshot, 8 October 2026); no Low or Medium reports were filed

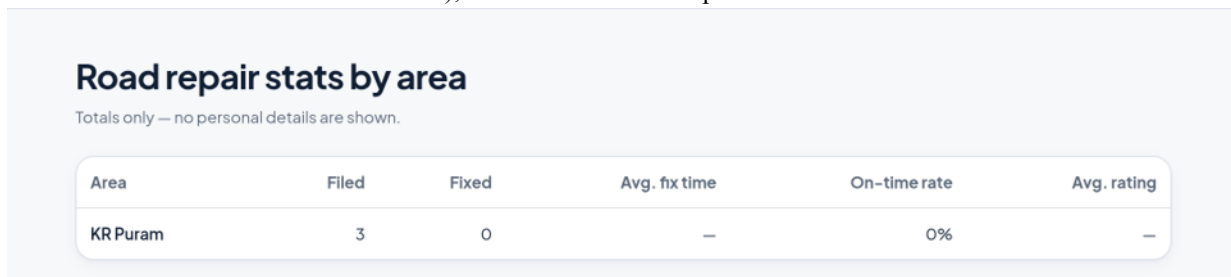


Fig. 6 — Public 'Road repair stats by area' page of the deployed application (8 October 2026): aggregate counts only, no personal data; values reflect test usage, not evaluation results.

V. DISCUSSION

Meaning of the results. Table 2 measures how well potholes are found, Table 3 whether finding them leads to repair; automatic evidence should cut verification effort, severity allows tighter deadlines for critical defects, and OTR with public statistics makes repair performance measurable and visible.

Zero-shot versus trained detector. The multimodal model needs no training data, handles unusual scenes and rejects non-road photographs, but may change with hosted-model updates, has uncalibrated confidences, depends on an external service and cannot be audited like fixed weights; a fine-tuned YOLO detector should give lower, more predictable

latency, can be versioned and tested on a fixed set, and shares the same output contract for comparison.

Limitations. Size and depth come from one monocular photograph without a reference object, so they are approximate and need on-site confirmation. One photograph is analysed per report; video and dash-camera streams are not processed. Location depends on phone GPS and permission, and typed addresses are not geocoded. Circular zones only approximate ward boundaries. Overdue alerts may be delayed without a staff session. Notifications are in-app and, for status updates, by e-mail; there is no SMS.

Threats to validity. Annotation errors and inconsistent inspector grades affect all metrics, so inter-annotator agreement should be reported (internal validity). A test set from one city, season or camera may miss

monsoon, night or rural conditions; the YOLOv8n dataset is not India-specific and does not document where its images were taken; and the trial has few areas and administrators (external). OTR measures meeting deadlines, not repair quality, hence citizen confirmation and rating (construct). The hosted model is not version-pinned, so its results should carry the evaluation date (reproducibility).

VI. CONCLUSION AND FUTURE WORK

DriveSafe Vision links annotated photographic evidence, location-based routing and area- and severity-specific deadlines with escalation and public statistics. A fine-tuned YOLOv8n reached an mAP@0.5 of 79.4 per cent on a held-out test set, and a protocol is defined to compare it with the deployed zero-shot model and other detectors. Future work will train and deploy YOLOv8 on Indian road images and complete Table 2, add on-device inference for poor connectivity, process video and dash-camera footage, merge duplicate reports by location and image similarity, use ward polygons, schedule the overdue sweep, and pilot the system with a municipal body to measure its effect on repair times.

ACKNOWLEDGEMENTS

This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors. The authors thank the Department of CSC, SEA College of Engineering and Technology, Bangalore, for its support and facilities. (Editable: add funding details or further acknowledgements if applicable.)

Declarations

Conflict of interest: The authors declare that they have no conflict of interest.

Originality: This manuscript is original, has not been published previously and is not under consideration for publication elsewhere. All authors have approved the manuscript and agree with its submission.

REFERENCES

- [1] Ministry of Road Transport and Highways, Transport Research Wing, *Road Accidents in India 2022*. New Delhi, India: Government of India, 2023, Table 3.5, p. 51.
- [2] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The pothole patrol: Using a mobile sensor network for road surface monitoring," in *Proc. 6th Int. Conf. Mobile Syst., Appl., Serv. (MobiSys '08)*, Breckenridge, CO, USA, 2008, pp. 29–39, doi: 10.1145/1378600.1378605.
- [3] Mednis, G. Strazdins, R. Zviedris, G. Kanonirs, and L. Selavo, "Real time pothole detection using Android smartphones with accelerometers," in *Proc. 2011 Int. Conf. Distrib. Comput. Sens. Syst. Workshops (DCOSS)*, Barcelona, Spain, 2011, pp. 1–6, doi: 10.1109/DCOSS.2011.5982206.
- [4] Koch and I. Brilakis, "Pothole detection in asphalt pavement images," *Advanced Engineering Informatics*, vol. 25, no. 3, pp. 507–515, 2011, doi: 10.1016/j.aei.2011.01.002.
- [5] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiya, and H. Omata, "Road damage detection and classification using deep neural networks with smartphone images," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 12, pp. 1127–1141, 2018, doi: 10.1111/mice.12387.
- [6] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, A. Mraz, T. Kashiya, and Y. Sekimoto, "Deep learning-based road damage detection and classification for multiple countries," *Automation in Construction*, vol. 132, Art. no. 103935, 2021, doi: 10.1016/j.autcon.2021.103935.
- [7] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, and Y. Sekimoto, "RDD2022: A multinational image dataset for automatic road damage detection," *Geoscience Data Journal*, vol. 11, no. 4, pp. 846–862, 2024, doi: 10.1002/gdj3.260.
- [8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.
- [9] 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, doi: 10.1109/TPAMI.2016.2577031.
- [10] G. Jocher, A. Chaurasia, and J. Qiu, *Ultralytics YOLOv8*, version 8.0.0, software, 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>. License: AGPL-3.0.
- [11] S. F. King and P. Brown, “Fix my street or else: Using the internet to voice local public service concerns,” in *Proc. 1st Int. Conf. Theory Pract. Electron. Gov. (ICEGOV '07)*, Macao, China, 2007, pp. 72–80, doi: 10.1145/1328057.1328076.
- [12] T.-Y. Lin *et al.*, “Microsoft COCO: Common objects in context,” in *Computer Vision—ECCV 2014*, Lecture Notes in Computer Science, vol. 8693. Cham, Switzerland: Springer, 2014, pp. 740–755, doi: 10.1007/978-3-319-10602-1_48.
- [13] G. Hovsepyan, *Pothole-Detection Dataset*, version 2, Roboflow Universe, 2023. [Online]. Available: <https://universe.roboflow.com/gurgen-hovsepyan-mbrnv/pothole-detection-gilij>. License: CC BY 4.0. Mirror: <https://huggingface.co/datasets/manot/pothole-segmentation2>. Accessed: Oct. 8, 2026.