International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

YOLOv11-Based AI-Driven Road Defect Detection, Severity Assessment, and Repair Cost Estimation Using OpenCV

Chandan Kumar Yedavalli[1], Pavan Teja Goud Nathi[2] and Ashok Kumar Narisetti[3]

[1, 2, 3]Department of Computer Science and Engineering (AI & ML), CMR Technical Campus, Hyderabad, Telangana, India Aligned with SDG 11 — Sustainable Cities and Communities

---

Abstract - Road defects in many Indian urban areas are still reported and assessed largely by manual inspection, which is slow and varies between inspectors. This paper presents a system that makes this process faster and more consistent. A citizen uploads a phone photograph of a damaged road together with its GPS location. A YOLOv11 detector locates potholes, and OpenCV extracts the defect contour and estimates its physical size using a reference-object scale factor. The estimated area and the detection confidence are combined into a rule-based severity score (Low, Moderate or Severe), which feeds a parametric repair-cost model based on public-works schedule-of-rates concepts. The GPS location is checked against ward boundaries to route the complaint to the responsible municipal officer. Complaints are stored in MySQL; citizens track their status through a Flask, Node.js and React.js application, and officers use a dashboard. The methodology, system architecture, dataset preparation and evaluation protocol are described. Quantitative results for Precision, Recall, F1, mAP@50, mAP@50–95 and stage-wise latency are pending training and field testing; final values will come from actual measurement rather than expected figures.

Key Words: Road damage detection, YOLOv11, OpenCV, severity assessment, repair-cost estimation, GPS geofencing.

1. INTRODUCTION

Road infrastructure is used in the daily activities of a significant proportion of the economic activity of a region, and consequently has a direct impact on the safety of road travelers, on vehicle operating costs, and on reliability of travel time. Roads can quickly become run-down in tropical and in the country with a monsoon climate. Repeated heavy traffic can cause potholes to form and gradually increase their size and depth, leading to further pavement damage within a short period. Therefore, the most common defects such as potholes require proper maintenance at regular intervals.

In most or all cases like this, these defects are discovered by sending people in search of them. That works, but not at scale: an inspector can only inspect a few kilometres a day, and rarely, if ever, will two inspectors rate the same defect in the same way; and the time between inspections might be sufficiently long for the small problem to turn into a big problem. The instruments remove the subjectivity, but to buy such equipment for routine survey would be a considerable expense for municipal agencies. Citizen complaint portals are on the flip-side. They are inexpensive, they scale well, but a picture affixed to a web form doesn't contain any sort of objective information as to the extent of the defect or what it will cost to correct it, and the complaint often is not directed to the correct department – just because the citizen picked the wrong dropdown.

A one-stage object detection method has become accurate and rapid enough to be operated on regular phone photos in recent years and a significant number of studies have experimented in this area with pavement distress in the last few years. A lot of that work comes to a halt at the stage when a ribbon of a box around the signature is created. But for the municipality, "Bounding box" is not the end of the story. The missing element is the chain of activities from detection to correction—how big is the defect, how critical compared to the other hundred that were reported this week, an estimate of the cost of repair, and of course who's responsible and who now has the accountability for it.

Whereas the proposed system addresses the same gap in connecting a number of stages in one workflow. It incorporates You Only Look Once version 11 (YOLOv11) detector and Open Source Computer Vision Library (OpenCV) measurement stage, a rule-based severity model, a parametric repair-cost model and a geospatial routing part. These are all part of a complaint-management system, which can be accessed by citizens without having to register an account. The various contributions are: an end-to-end pipeline enabling the citizen's photo to be passed through detection and assessment to an assigned public functionary at the Municipality; a measurement and severity procedure, aiming to provide citizens with a more consistent procedure to evaluate their exposure; a configurable cost model, which provides the possibility to update the model, according to the different States or financial year; and a point-in-polygon jurisdiction method, that gives the possibility to determine the responsible Ward without asking citizens to provide the exact one of which they are residing. In the text, components which have been actually implemented are highlighted as such whilst components that are still proposed/validating are not highlighted.

The remainder of this paper is organised as follows. Section 2 reviews related work on road-damage detection and explains where the proposed system is placed. Section 3 states the problem and presents the proposed methodology and system design, covering the system

architecture, the dataset, the measurement pipeline, the detector, the severity model, the cost model and the jurisdiction resolver. Section 4 describes the implementation and experimental setup and the evaluation protocol. Section 5 outlines the planned reporting of results, the comparison with previous approaches, and a discussion of the strengths and limitations of the system. Section 6 gives the conclusion and future scope.

2. RELATED WORK

The number of image datasets for road damage with the help of the smart phones attached to cars on the roads became even larger when Maeda et al. [3] generated a large dataset by mounting smart phones on vehicles in Japan and demonstrated the ability of convolutional detection models to detect damage through a user's smartphone photos. It evolved into Road Damage Detection 2022 (RDD2022) [4] later—a challenge that contains images from countries other than Iceland and which has been an established standard for healthcare RDD in the field with the use of Crowdsensing-based Road Damage Detection Challenge (CRDDC'2022) [5].

The detector side of the literature comes and goes with the evolution of the YOLO family. The methodology follows the plan of Redmon et al., [6] which is the one with a single stage and passed down via YOLOv8 and YOLO-NAS, as surveyed by Terven et al., [7]. Khanam and Hussain [1] outline the changes to the architecture of YOLOv11, including the Spatial Pyramid Pooling – Fast (SPPF) stage and the C3k2 backbone blocks, which achieves comparable accuracy with fewer parameters. For pavement distress recognition, there are two architectures that have been adapted: Zhang et al. [8] introduce multi-scale feature extraction to a fusion attention mechanism for multi-class damage recognition. Although these are not the newest versions, they are equally relevant to deployment concerns as well as accuracy, such as the development of an even more lightweight detector that is YOLOv7-tiny [9] for edge hardware. There are two studies which are directly applicable to our deployment context with respect to traffic loading and the quality of road surfacing which are far removed from Japanese and European subsets: Sathvik et al. [10] and Dharneeshkar et al. [11].

Comparatively smaller studies have been carried out measuring a road defect once detected. The boundaries of potholes are estimated using OpenCV contour extraction and morphological operations [12]. For analysis of irregularity, classical shape and acutance measures [13] are helpful, and Koch and Brilakis [14] showed that the use of a reference object could help to convert measurements made on the image to approximate physical units. The current work is based on this idea of reference-object.

From the literature reviewed, a study was not located which correlated the detection phase to the entire process of severity measurement, cost estimation, jurisdiction resolution, and officer assignment. The tasks are typically not included in detection system or performed manually upon detection. The integration gap is the area of interest of the present work. This work does not claim to have a fundamentally superior detector, but rather aims to consider the linking of the detector stages as a viable workflow.

3. PROPOSED METHODOLOGY AND SYSTEM DESIGN

3.1 Problem Statement

There are several practical limitations to the current flow. Manual inspection is difficult to scale up to a city-level road network at regular intervals. While complaint portals can take photos, they do not automatically analyze the photos and generally do not provide a useful objective defect class, measurement or 'severity score' for the complaint. Estimates are typically made manually upon arrival at a site to determine repair costs, thus slowing prioritisation process. Last but not least, it is sometimes necessary for the complainant to understand the assignment of the ward and/or department, and, at times, this can be slow and less accurate if it is near the ward and/or department separations.

The problem in this work is thus formulated. Starting from a single photo captured by an untrained citizen using a consumer camera, in conjunction with Global Positioning System (GPS) coordinates at the time of photography, the system is intended to generate: (i) a characterization of the state of the road defect, including its type and location, (ii) an estimate of the size of the road defect, (iii) a severity classification on a fixed scale of ordering, (iv) an estimated repair cost, and (v) a nominated responsible municipal authority and officer. The entire process should not require any special worker road-inspection equipment.

3.2 Proposed Methodology

This proposed workflow has nine stages in it. First, an image machine is captured via the web and mobile application. Uploaded GPS location data comes from the geolocation API or from the Exchangeable Image File Format (EXIF) data of the photograph, if available. Subsequent to detection, image reduced in size and normalised and colour corrected. The trained model based on the concept of YOLOv11 then outputs a corresponding box, class label and level of confidence around every detected pothole. The cropped and OpenCV-processed area of the region is then cropped and OpenCV processes to extract the contour and get physical size based on the calibration factor shown in Sec. 3.5. The severity is based on the defect class, measured area and confidence score. The measurement area and the severity is then passed on to the cost model. Simultaneously the GPS point is compared against the stored ward polygons to find the appropriate municipality. Lastly, the complaint is put into the MySQL database, a unique complaint identifier is generated, the complaint is added to an officer and the

identifier is sent back to the citizen so they will be able to follow it.

The four Analytical components remain independent to enable their separate updating. Other modules or the main application program can be changed without affecting the detector, adjustable severity thresholds can be corrected, adjusting repair rates can be recalibrated, and new boundaries for wards can be drawn without changing the other modules. The separation helps to be useful as public works rates can also vary on an annual basis and ward boundaries can change following administrative delimitation.

3.3 System Architecture

The overall system structure is shown in Fig. 1. The public portal will be implemented as a React client, and citizens will not have to log in to it. Sends the road image and GPS location to the backend. The inference tasks are split from the application and transaction tasks: The Python service handles the YOLOv11 model running and OpenCV processing, whereas application server runs authentication, complaint status, officer and administrator dashboards. Complaint, municipality, officer, and complaint-status history are persistent data that are stored in MySQL. Access control is done at municipal level to ensure that the officer assigned to one ward can't access complaints from another ward.

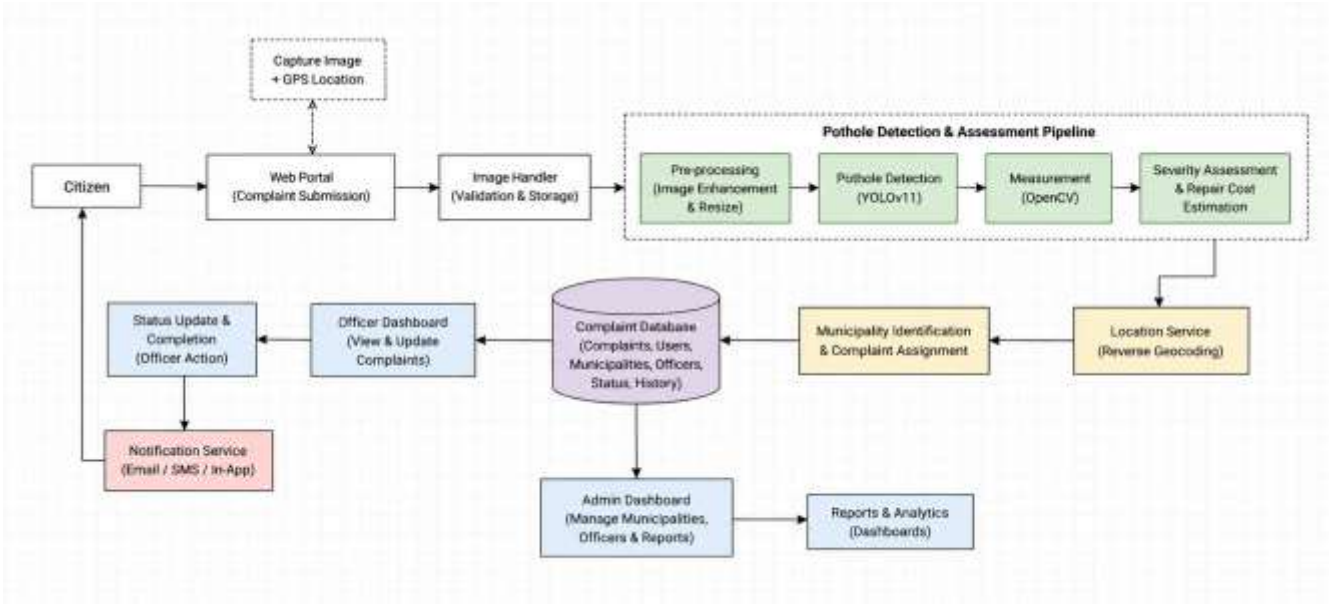


Fig. 1: System architecture of the proposed framework.

3.4 Dataset

The Pothole Detection dataset from Roboflow Universe consists of 1,482 road images containing potholes. These images are annotated with bounding boxes that mark the location of each pothole. The dataset is used to train and evaluate the YOLOv11 model for detecting potholes in road images.

The dataset has a single class, named *pothole*, which allows the model to focus specifically on pothole detection. The images are divided into three sets: training, validation, and testing, following a 70:20:10 ratio. The training set helps the model learn pothole features, while the validation set is used to monitor its performance during training. The testing set is used to evaluate the trained model on images that were not used during training. The resulting split is summarised in Table 1.

Before training, the images and annotations are checked for errors. The images are resized to 640 × 640 pixels while maintaining their aspect ratio, and the annotations are converted into the format required by YOLOv11. Corrupted images and incorrect annotations should be removed or corrected. Data augmentation can also be applied to the training images to help the model recognize potholes under different lighting conditions, road surfaces, and viewing angles. The validation and testing sets are kept separate from training augmentation to ensure a fair evaluation of the model.

Table 1: Dataset split. Figures are pre-cleaning approximations and are to be updated with final counts.

Split	Images (approx.)	Proportion
Training	1,037	70%
Validation	296	20%
Testing	149	10%
Total	1,482	100%

3.5 Measurement Pipeline

The cropped area of each region identified by YOLOv11 is passed through various OpenCV processes. The cropped image is then converted to a grayscale and filtered by Gaussian filter to diminish the texture effect of the asphalt road. Using Otsu thresholding to extract the defect from the road surface. Applying morphological opening and closing operations with a structuring element of 5×5 helps to reduce noise and merge gaps in detected boundary. The resulting contours are extracted using cv2.findContours, and the largest contour by area is

treated as the main defect boundary. Its size is represented using both an axis-aligned rectangle from cv2.boundingRect and a rotated rectangle from cv2.minAreaRect.

To convert pixel measurements into approximate physical dimensions, a scale reference is required. The method uses a reference object with a known width W_{ref} that appears in the image. Its width is measured in pixels as P_{ref}, producing the following scale factor:

$$k = W_{ref} / P_{ref} \quad (1)$$

in centimetres per pixel. Length and width of the defect follow as

$$L = l_{px} \times k, \quad W_d = w_{px} \times k \quad (2)$$

and surface area from the contour pixel count C_{px} as

$$A = C_{px} \times k^2 \quad (3)$$

In these instances, the minimum contour width is measured along several points of the skeleton and the distribution the results is passed along to the severity stage, not using one width value. There are two major restrictions. With the reference-object method it is assumed that the object measured is part of the visible road surface and that the reference object and the defect are not too far apart. As depth is not provided by a single image of red–green–blue (RGB) data, the areas reported are surface areas, and not the true three-dimensional volume.

3.6 Detection with YOLOv11

The proposed method of road-defect detection is based on the Ultralytics YOLOv11 framework. The added C3k2 blocks in the backbone, modified SPPF module, and C2PSA attention block in the neck are also new features of YOLOv11 compared to previous versions like YOLOv8 or YOLOv9. The changes reported here are said to enhance either the precision of small targets or the accuracy of irregular targets [1],[2]. The detector should also deliver fast enough results for this application, which is accessible by citizens.

- Backbone: CSPDarknet-based with C3k2 blocks. Neck: PANet with C2PSA multi-scale fusion.
- Head: anchor-free and decoupled. Input resolution 640×640, raised to 1280×1280 where fine cracks dominate.
- Loss: binary cross-entropy for classification with Complete Intersection over Union (CIoU) and distribution focal loss (DFL) for box regression.

The total training objective is a weighted sum of the three terms,

$$L_{total} = \lambda_{cls} \cdot L_{cls} + \lambda_{box} \cdot L_{CIoU} + \lambda_{dfl} \cdot L_{DFL} \quad (4)$$

where the box term is the complete IoU loss

$$L_{CIoU} = 1 - IoU + \rho^2(b, b_{gt}) / c^2 + \alpha v \quad (5)$$

with $\rho(\cdot)$ the distance between box centres, c the diagonal of the smallest enclosing box, v the aspect-ratio consistency term and α a trade-off weight. During training, the YOLOv11 model starts with weights pretrained on the Common Objects in Context (COCO) dataset and is further trained using the selected pothole detection dataset to learn pothole-specific features.

Algorithm 1: YOLOv11-based road defect detection.

Input: road image I, confidence threshold τ_c, non-maximum suppression (NMS) threshold τ_{nms}. Output: D = {($class_i$, $bbox_i$, $conf_i$)}.

1: $I_{pre} \leftarrow$ Preprocess(I)
2: F $\leftarrow$ Backbone(I_{pre}); F′ $\leftarrow$ Neck(F); P $\leftarrow$ DetectionHead(F′)
3: $D_{raw} \leftarrow$ { p ∈ P : p.confidence ≥ τ_c }
4: D $\leftarrow$ NonMaxSuppression(D_{raw}, τ_{nms})
5: return D

3.7 Severity Grading

Severity is computed based on three parameters already calculated in previous stages—for example, the defect area, divided by a reference area, the risk weight of the defect class, and the detection confidence of the model. A score is called severity score and is defined as:

$$S = w_1 \cdot \hat{A} + w_2 \cdot R_c + w_3 \cdot conf \quad (6)$$

$$\begin{aligned} &Low: S < \theta_1; \\ &Moderate: \theta_1 \leq S < \theta_2; \\ &Severe: S \geq \theta_2 \end{aligned} \quad (7)$$

As working values, the two thresholds used in the current implementation are set to be $\theta_1 = 0.35$ and $\theta_2 = 0.65$. Currently, these thresholds and the weights used in Eq. (6) need to be calibrated with expert assessments. Thus, Table 2 is intended to be illustrative of the desired severity mapping, and is not a fully validated rule set.

Table 2: Illustrative severity mapping. Thresholds require calibration against expert assessment before deployment.

Defect type	Area range (cm²)	Severity	Indicative action
Pothole	< 150	Low	Routine maintenance
Pothole	150–500	Moderate	Scheduled repair (2–4 weeks)
Pothole	> 500	Severe	Urgent repair (< 72 hours)

Two main reasons explain why a rule-based approach was chosen over a learned severity classifier. First, there is currently not enough data labelled by experts to train such a classifier. The labeling with author judgment could lead to another subjectivity that is out of the reduction system's intention. Second, in order to contribute to decision-making in municipalities, the concept of accountability is necessary. There are reasons why a repair was prioritised and a weighted score with clearly defined thresholds can be made easily comprehensible by an officer. This step might be skipped once a large enough expert-labelled dataset is available and an alternative architecture involving a learned model can be used instead.

3.8 Repair-Cost Estimation

The repair-cost model is parametric and consists in a "public works schedule of rates" type, similar to a conventional schedule of rates. A unit rate is used on the quantity of the repairs being performed and adjusted

depending on the nature and urgency of the repairs being performed. The model is represented as follows:

$$C = A \times R_{type} \times M_{severity} + C_{fixed} \quad (8)$$

where A is the area of the defect measured in sqm, R_{type} is the cost per sqm of the repair commensurate with the defect type, $M_{severity}$ is a multiplier usualling at least 1, reflecting the premium to be placed on work being carried out outside normal operating hours, and C_{fixed} reflects costs of mobilisation that are not proportional to area. Recalibration to a different state or fiscal year is a configuration change because it involves entering all four quantities into the database, and they are not compiled into the application.

Instead of assuming a value for R_{type}, we refer to the Telangana Common Schedule of Rates 2022-23 [15] from which Public Works estimates R_{type} values in the state. It was developed from three primary inputs instead of directly per square metre, i.e. the money paid to the skilled and unskilled labourers, followed by the machinery hire rate if a pothole-repair machine is used specifically for pothole work (Roads & Bridges, Part-II, Item 5: patching with bituminous: 40–45 mm HBG metal, 10–12 mm key chips 80/100-grade bitumen binder). These primary inputs are scheduled for 2022-23 and are presented in Table 3.

Table 3: Primary cost inputs for bituminous pothole patching, Telangana Common Schedule of Rates 2022–23 [15]. Figures are the scheduled basic rates as published, not derived.

Input	Scheduled rate (2022–23)	Source item
Bitumen binder, 80/100 grade	₹52 / kg	I&CAD basic materials, Sl. No. 2
Unskilled (mazdoor) labour	₹520 / day	R&B labour rates, unskilled category
Skilled mason / patching crew	₹580–610 / day	R&B labour rates, skilled category
Dedicated pothole-repair machine (hire)	₹1,041 / hour	R&B machinery hire charges

This build-up of primary inputs, or material data against the quantity of bitumen, aggregate and labour in Item 5, is scaled to the mix depth used on site as this changes, and is the standard data used in R_{type} for Eq. (8) which is promulgated by Ministry of Road Transport and Highways (MoRTH) in State Schedules. We do not report a calculated R_{type} value, as the type constructed will be dependent on the patch depth, proportion of aggregates that the actual implementation adopts and the result will only be the final composite R_{type} once the end-user has independently checked against the current version of the index in force at the time of deployment.

There are two restrictions to be observed. First, the value produced by Eq. (8) is an indicative value which is intended as a means of prioritising and not as a formal tender quotation. It is displayed in the user interface as such. Second, A is a projected surface area and, therefore, it is not as accurate for estimating the extent of damage on the surface if it is a deep pothole as if it is surface damage. If there is a large difference in repair costs based on depth, the estimated number is to be viewed as a minimum figure.

3.9 Jurisdiction Resolution and Officer Assignment

Ward polygons are represented as GeoJSON polygons. Using the coordinates listed with the complaint, a point in polygon test locates the ward that is covered by the point. When the point is not in any of the polygons stored—how this can happen depends on how well the GPS location has been acquired and in redrawn boundaries—the system stops and records the problem to an internal flagging log, allowing them to mark the assignment for further review instead of turning down the complaint. The ward is then known and the complaint is taken up by the officer in the ward with the lowest number of active cases reducing the workload without the need for a supervisor to intervene.

Algorithm 2: GPS-based jurisdiction resolution and officer assignment.

Input: coordinates (lat, lon), ward polygon set W, officer roster O. Output: (ward_id, officer_id).
1: for each $w_i \in W$: if PointInPolygon((lat, lon), w_i.boundary) then ward_id ← w_i.id; break
2: if ward_id is null then ward_id ← NearestWardCentroid((lat, lon), W); flag for review
3: officer_id ← argmin over o ∈ O[ward_id] of o.active_case_load
4: return (ward_id, officer_id)

A ward code, pothole identifier and a running sequence number are then used to generate a complaint identifier and the record is committed to MySQL, and an officer is notified of the occurrence. The citizen is given the identifier and can use it (or the mobile number he submitted) to find its status without having to sign up for an account. In the tracking view, only information about the complaint (state), the detected defect, the severity, the estimated cost, and the municipality responsible for the defect has been made public; the information regarding the officer and the municipality's notes has not been included in the view.

4. IMPLEMENTATION AND EXPERIMENTAL SETUP

4.1 Software Environment and Training Configuration

The software environment is running Python 3.10 on PyTorch along with implementation of Ultralytics YOLOv11. The measurement stage is implemented using OpenCV 4.x, while the back end application layer is a developed Flask framework, along with Node.js and Express. For the frontend, they made it with React.js, and for the persistent data storage, they are using MySQL 8.x. The training hardware used has graphics processing unit (GPU) model Tesla T4 and total training time is 1 hr 15 min. The Ultralytics framework defaults [2] are shown in Table 4 as the default training configuration used to start training. If any changes are made while training will be explicitly mentioned.

Table 4: Training configuration. Values are the documented Ultralytics YOLO11 defaults [2], adopted as the training starting point; deviations used in the actual run remain to be recorded.

Hyperparameter	Value	Remarks
Input resolution	640 × 640	Ultralytics default; 1280 tried for fine-crack sensitivity
Batch size	16	Ultralytics default; reduced under GPU-memory constraints
Epochs	100	Ultralytics default, with early stopping (patience = 100)
Optimizer	Stochastic gradient descent (SGD), auto	Ultralytics default optimizer selection
Initial / final learning rate (lr0 / lrf)	0.01 / 0.01 × lr0	Ultralytics default cosine-style decay
Momentum / weight decay	0.937 / 0.0005	Ultralytics default
Warm-up epochs	3.0	Ultralytics default
NMS IoU / confidence (inference)	0.7 / 0.25	Ultralytics default predict-time thresholds

4.2 Evaluation Protocol

The performance of the trained detector is tested on the held-out test set with Precision, Recall, F1, mean average precision (mAP) at IoU 0.5 (mAP@50) measures. These measures have the following meaning:

$$Precision = TP / (TP + FP), \quad Recall = TP / (TP + FN) \tag{9}$$

$$F1 = 2 \cdot Precision \cdot Recall / (Precision + Recall) \tag{10}$$

$$mAP = (1/N) \Sigma AP_i \tag{11}$$

Here, TP, FP and FN denote the numbers of true positives, false positives and false negatives, N is the number of classes and AP_i is the average precision of class i.

The average precision at each IoU thresholds between 0.5 and 0.95 in increments of 0.05 is then averaged to calculate the mAP from 0.5 to 0.95. The task of severity is being considered as an ordinal classification problem and compared against labels assigned expertly on a set that remains unseen during the learning process. Measures reported are accuracy, per-class precision and recall and a confusion matrix. The cost estimation is evaluated against the true municipal repairs costs upon completing the work with the mean absolute error (MAE), root mean square error (RMSE) and mean absolute percentage error (MAPE) metrics. The jurisdiction performance is calculated as the number of complaints received in the correct ward / total number of complaints received. Individual system latency is reported, as well as overall end-to-end latency.

5. RESULTS AND DISCUSSION

5.1 Detection Results

Precision, Recall, mAP@50 with the overall average will be reported for pothole detection. Precise Measure and Recall are grouped together as the creative presence of each measure alone may provide a limited view. The higher the Precision, the less uncertain detections a model may have, and the higher the Recall, the more false detections a model may have. mAP@50 evaluates localisation based on a relatively "lenient" overlap criterion while mAP@50 evaluates localisation over multiple overlap thresholds. This is especially true of the stages after measurement, severity, and cost, since these depend upon the quality of the detected region. However, per-class results also are important, as the model might be able to detect better on easy-to-see defects like potholes. Table 5 lists the metrics that will be reported for pothole detection.

Table 5: Pothole detection performance

Metric	Test-set result
Precision	89.1 %
Recall	85.9 %
mAP@50	91.7 %
F1	87.5 %

The final version will also feature the following curves: in order to determine if the model has converged, the training loss and validation-mAP curves will be provided; common class confusions will be presented in a 4-class confusion matrix; and validation precision–recall curves, showing the classes at varying confidence levels, will be displayed. Examples of the contours extracted, and the resulting contour severity will be given in qualitative examples for the original road image. These examples will enable the reader to understand the average or typical levels of measurement and severity rather than just numerical averages.

In addition to the detector delay, the citizen will add the delay of the work required for the detector, the delay for the measurement of the severity and the cost, the delay necessary for storage of the data in the database and the delay necessary for the transmission between the detector and another node. The reported breakdown will consist of YOLOv11 Inference, OpenCV Measurement, Severity & Cost Calculation, and Overall End to End Response Time with Network Latency. This will help to identify possible bottle necks and see if the application can return a result in a timely manner for the interface that the citizen would face.

5.2 Comparison with Existing Methods

Not yet a direct comparison between all these studies and studies mentioned in the Table 6 is possible due to the fact that the testing protocol followed in the studies listed in the table does not remain same and some of the listed studies use partly overlapping dataset and some uses private datasets as well. As such, Table 6 is employed to compare the scope of these systems, not their numbers.

Table 6: Comparison of scope with representative prior work. Numerical benchmarking is pending.

Method	Backbone	Task scope	Severity / cost / routing
Sathvik et al. [10]	YOLOv7	Pothole detection	Not addressed
Zhou et al. [9]	YOLOv7-tiny	Road damage detection	Not addressed
Zhang et al. [8]	Improved YOLOv11	Multi-class damage	Not addressed
Proposed system	YOLOv11 + OpenCV	Detection and measurement	Severity, cost and GPS routing

5.3 Discussion

The significant practical advantage of the system is that it may contribute to minimizing two kinds of delay. The first is the time taken to do the triage on complaints. A complaint may come with a class of defect, a measurement, level of severity and a place to be assigned to it, so that the officer only needs to deal with the decision, instead of manual data entry. The second one is subjective assessment variation. Damage to similar complaints can be assessed with the same rules regarding the complaint from different wards. As the primary data collection tool will be a cell phone (which citizens already own), coverage can expand as the phone is used without any significant investment in additional hardware.

The advantages of the approach must be balanced with its constraints. Crowdsourced reports are more likely to be from complaints areas, which may not be the same location as poor road conditions. The system should thus, not touch upon the replacement of formal road survey programmes, but should complement. Furthermore, automated prioritisation encoded a decision by a person into a set of weights and thresholds. The second consequence is that defective weight could be undercut every time, making it easier to see how that can lead to a clear audit trail that is also transparent. This is in the same way that severity calibration, as described in Sec. 3.7 is important to be validated before a system is made operational.

Respecting privacy is also a concern. The system captures a picture, cell phone number and location, which can combine to identify a complainant. There is no public tracking endpoint for mobile numbers, they are only provided for tracking and notifications. They issued the complaint/defect information in Public Status View—where they did not reveal complainant information. Images are stored indefinitely until the time of the complaint, and a specified audit period of time, and the location information is only held to the level of precision needed for the jurisdiction process.

5.4 Limitations

The accuracy of measurement relies fully on the calibration assumption, which is very fragile in its normal usage. The scale factor will be biased in an opposite direction if a reference is at a different distance from the camera than the defect, or if a photograph is taken from a sharp oblique angle. There is no recovery of depth, hence no estimate of volume costs for any deep pothole.

Low light, good shadow contrast - much like the wet pavements, standing water that obscures the pothole completely, and moving vehicles leading to motion blur are common issues and create poor detection quality on the routes on which we are focusing. This Consumer GPS can be within 3 to 10 metres of the point of complaint, which could be a complaint recorded near a ward boundary for which the wrong ward could be given credit. The severity thresholds and the cost rates are not learned end to end, and must be hand-set and periodically re-calibrated by experts.

6. CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

This paper proposes an end-to-end system where a citizen will submit a picture of the road defect and it will get connected to the relevant Municipal Officer. YOLOv11 detects and classifies, OpenCV contour pipeline gives an approximation of the measurement, rule-based model assigns severity, parametric model estimates the repair cost, and the point-in-polygon method is used to determine the responsible ward and officer. These components are embedded in a web application, written in Flask, Node.js, React.js, and using MySQL, which citizens will then use without signing up for it. The key element is the integration of these stages and not the detector itself. However, with many of the systems in place, all the information is limited to the drawing of a bounding box, whereas the municipal application requires information on measurement, severity, cost, jurisdiction and assignment information as well. These are only placeholders as model training and deployment in the field are ongoing. The following steps involve recording the benchmarked performance outcomes, setting severity levels according to expert calibration and comparing costs with actual repair costs.

6.2 Future Scope

Monocular Depth Estimation is one significant extension in the future. This is likely to decrease reliance on a reference object and could enable volume-based cost estimation based on a single image. The addition of supporting dash camera footage could also transform the system from one that sticks out a ticket into a door-to-door road monitoring system. An empirical control mechanism involving a comparison between estimated costs and the actual costs after the repairs has been accomplished will gradually lead to better improvements of the cost model for a specific municipality. The information can then be used in the future to train a severity model, rather than the current rule-based approach, under the guidance of experts who have labeled the data. An on-device YOLOv11-nano version can be used to report in an area with poor connectivity for deployment. The officer dashboard could also include an explainability overlay that would indicate what portion(s) of the picture caused the severity decision.

Incorporating more data from India is likely to prove more helpful in improving performance in the target area.

REFERENCES

1. Khanam, R., Hussain, M.: YOLOv11: An overview of the key architectural enhancements. arXiv:2410.17725 (2024)
2. Ultralytics: YOLO11 Documentation and Configuration Reference. https://docs.ultralytics.com/usage/cfg (accessed 2026)
3. Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., Omata, H.: Road damage detection and classification using deep neural networks with smartphone images. Computer-Aided Civil and Infrastructure Engineering 33(12), 1127–1141 (2018)
4. Arya, D., Maeda, H., Ghosh, S.K., Toshniwal, D., Sekimoto, Y.: RDD2022: A multi-national image dataset for automatic road damage detection. arXiv:2209.08538 (2022)
5. Arya, D., Maeda, H., Ghosh, S.K., Toshniwal, D., Omata, H., Kashiyama, T., Sekimoto, Y.: Crowdsensing-based road damage detection challenge (CRDDC'2022). In: 2022 IEEE International Conference on Big Data, pp. 6378–6386 (2022)
6. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proc. IEEE CVPR, pp. 779–788 (2016)
7. Terven, J., Córdova-Esparza, D.M., Romero-González, J.A.: A comprehensive review of YOLO architectures. Machine Learning and Knowledge Extraction 5(4), 1680–1716 (2023)
8. Zhang, L., Deng, Y., Zou, Y.: Automatic road damage recognition based on improved YOLOv11 with multi-scale feature extraction and fusion attention mechanism. PLOS ONE (2025)
9. Zhou, H., Chen, S., Yan, G., Huang, Y., Lu, Z.: An intelligent road damage detection method based on YOLOv7-tiny. In: 2023 IEEE ICCT, pp. 286–290 (2023)
10. Sathvik, M., Saranya, G., Karpagaselvi, S.: An intelligent CNN-based potholes detection using YOLO-v7. In: 2022 ICACRS, pp. 813–819 (2022)
11. Dharneeshkar, J., Aniruthan, S., Karthika, R., Parameswaran, L.: Deep learning based detection of potholes in Indian roads using YOLO. In: 2020 ICICT, pp. 381–385 (2020)
12. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
13. Rangayyan, R.M., El-Faramawy, N.M., Desautels, J.E.L., Alim, O.A.: Measures of acutance and shape for classification of breast tumours. IEEE Transactions on Medical Imaging 16(6), 799–810 (1997)
14. Koch, C., Brilakis, I.: Pothole detection in asphalt pavement images. Advanced Engineering Informatics 25(3), 507–515 (2011)
15. Government of Telangana, Board of Chief Engineers: Common Schedule of Rates as per T.S. Standard Data for the Year 2022–23 (Roads & Bridges, Part-II; Irrigation & CAD basic material and labour rates, Part-I), effective 1 June 2022. https://pred.telangana.gov.in (accessed 2026)