International Journal of Electrical, Electronics and Computer Systems

ISSN: 2347-2820

Volume 15 Issue 1s (2026) | Pages: 342-352

AI-Based Vehicle Damage Detection, Cost Estimation, and Insurance Claim Prediction System

Sangeetha Navale[1], Prachi Raut[2], Prachi Ukey[3], Atharva Kulkarni[4]

[1,2,3,4] Department of Computer Engineering, Genba Sopanrao Moze College of Engineering, Balewadi, Pune Affiliated to Savitribai Phule Pune University, India

Email: n.sangeetha.rd@gmail.com, prachiraut711@gmail.com, ukeyp9051@gmail.com, kulkarniatharva422@gmail.com

Peer Review Information

Type: *Article*
Received: *3 February 2026*
Revised: *2 March 2026*
Accepted: *1 April 2026*
Published: *23 May 2026*

Abstract

Road accidents result in billions of rupees worth of vehicle damage every year, yet the process of assessing that damage and processing insurance claims has remained stubbornly manual, slow, and error-prone. This paper presents the design, implementation, and evaluation of an AI-based vehicle damage detection and cost estimation system that automates the entire assessment pipeline through the integration of deep learning and machine learning. The proposed system is a Flask-based web application in which a pre-trained YOLOv8 object detection model identifies damaged vehicle components from user-uploaded photographs, including parts such as bonnets, bumpers, doors, and fenders. Detected parts are cross-referenced against a structured JSON pricing database to compute a repair cost estimate using a part-wise summation formula. A rule-based module predicts probable internal damages from observed external patterns, and a trained scikit-learn classification model determines insurance eligibility and computes coverage. The system classifies overall damage severity into minor, moderate, and major categories and produces smart repair and financial recommendations. Experimental results demonstrate that the YOLOv8 model achieves a mean Average Precision (mAP@50) of 0.79 across seven damage classes, with a cost estimation mean absolute percentage error of 11.4% relative to authorised workshop quotations. The integrated pipeline reduces a conventionally multi-day assessment process to under ten seconds, offering a practical decision-support tool for vehicle owners, insurance companies, and repair workshops.

Keywords: Vehicle Damage Detection; YOLOv8; Deep Learning; Repair Cost Estimation; Insurance Prediction; Flask; Computer Vision; Severity Estimation; Machine Learning; Object Detection.

How to Cite This Article

Navale, S., Raut, P., Ukey, P., & Kulkarni, A. (2026). *AI-based vehicle damage detection, cost estimation, and insurance claim prediction system*. *International Journal of Electrical, Electronics and Computer Systems,* 15(1s), 342–352.

Introduction

The aftermath of a road accident places vehicle owners in a stressful and often protracted situation. Determining which components are damaged, estimating repair costs, and navigating insurance claims has historically depended on physical inspections by trained assessors at authorised workshops — a process that typically spans several business days and involves considerable subjectivity. Insurance companies face parallel challenges: fraudulent or inflated claims are difficult to detect without objective automated tools, and the sheer volume of claims makes manual verification operationally expensive.

Deep learning, and object detection architectures in particular, have demonstrated that these challenges are technically tractable. Systems capable of identifying specific vehicle components and classifying damage from ordinary photographs have appeared in the research literature since the mid-2010s. The YOLO (You Only Look Once) family of single-stage detectors, and specifically YOLOv8, has made real-time, high-accuracy damage detection feasible even on consumer-grade hardware.

Despite this technical progress, most research prototypes stop at the detection stage and do not extend to cost estimation, internal damage inference, insurance eligibility determination, or actionable reporting. The system described in this paper bridges that gap by integrating YOLOv8-based detection with a structured pricing database, a probabilistic internal damage estimator, a machine-learning insurance prediction model, and an interactive web dashboard — producing a complete, end-to-end damage assessment report from a single photograph upload.

The specific objectives of this work are:

1. To implement a YOLOv8 model capable of detecting and localising individual damaged vehicle components from photographs.
2. To develop a cost estimation module that maps detected components to a structured pricing database and aggregates repair costs.
3. To incorporate a rule-based internal damage prediction module that infers probable hidden damage from observed external patterns.
4. To integrate a machine learning model that assesses insurance eligibility and computes expected coverage.
5. To deliver a web-based platform presenting all outputs — damage images, cost breakdowns, insurance estimates, and recommendations — in a printable report.

The remainder of this paper is organised as follows. Section II reviews related work. Section III describes the system architecture and methodology. Section IV presents experimental results. Section V discusses findings and limitations. Section VI concludes.

Literature Review

CNN-Based Classification

Early work established that Convolutional Neural Networks (CNNs) substantially outperform handcrafted feature descriptors for binary damage classification. Patil et al. trained a CNN to distinguish damaged from undamaged vehicles, achieving accuracy above 70% on a private dataset. Subsequent work using the VGG family delivered a three-stage pipeline (identification, localisation, severity) with VGG-19 reaching 95.22% detection accuracy, though severity estimation accuracy fell to only 57.89% — highlighting the persistent difficulty of fine-grained severity discrimination.

Despite these promising results, CNN-based classification approaches are inherently limited in their ability to provide precise spatial localisation of damage regions, as they primarily focus on image-level predictions. This restricts their applicability in real-world insurance scenarios where detailed damage assessment is required. Furthermore, variations in lighting conditions, occlusions, and complex backgrounds can significantly affect model performance. These challenges motivated the transition towards more advanced object detection and segmentation-based approaches, which are discussed in the following sections.

Region-Based Object Detection

Faster R-CNN and Mask R-CNN advanced the field by enabling pixel-level localisation. Reddy et al. demonstrated that Mask R-CNN's instance segmentation masks enable area-based severity scoring more informative than bounding-box classification alone. Fouad et al. combined Mask R-CNN with an InceptionResNetV2 backbone and integrated the result into a web application for insurance claim estimation, achieving strong performance across all evaluation criteria.

YOLO-Based Detection

Pérez-Zárate et al. showed that a YOLOv5 ensemble could process thousands of vehicle images per minute in a live insurance deployment, satisfying throughput requirements that two-stage detectors cannot meet. Their comparative analysis confirmed YOLOv8's superior AUC and mAP.

The landmark three-module pipeline of Ma et al., evaluated across both fleet and user-submitted mobile images, formalised the modular design pattern — vehicle presence verification, part localisation, and damage severity scoring — that the present work builds upon.

Cost Estimation and Insurance Integration

Aithal et al. linked CNN classification outputs to a repair cost database, one of the earliest end-to-end damage-to-cost demonstrations. Khan et al. explored multi-modal fusion for improved photographic robustness.

Hasan et al., in a systematic review of 55 papers, identified the absence of end-to-end insurance integration as the most commercially significant open gap — the precise gap that the present system addresses.

Comparative Summary of Related Work

Table I summarises the principal related works, enabling direct comparison with the proposed system.

Table 1: Comparative Summary Of Related Work In Vehicle Damage Detection

Reference	Method	Dataset	Metric	Strengths	Limitations
Patil et al. (2017) [8]	Custom CNN	Private	Acc >70%	Pioneering DL baseline	No localisation or cost output
Reddy et al. (2022) [6]	Faster R-CNN / Mask R-CNN	Private	mAP ≈80%	Pixel-level segmentation	Heavy compute; no insurance
Pérez-Zárate et al. (2023) [5]	YOLOv5 Ensemble	TartesiaDS	High mAP; 1000s img/min	Insurance throughput; low FPR	Private data; no cost output
Fouad et al. (2023) [7]	Mask R-CNN + IncResNetV2	Private	Best across metrics	Web-app; claim estimation	Occlusion sensitive
Aithal et al. (2023) [4]	Transfer Learning CNN	Public + price DB	Good with small data	Cost DB linkage	Coarse severity only
Khan et al. (2024) [2]	Multi-modal Autoencoder	Custom	Improved noise robustness	Handles occlusion and noise	Requires multiple sensors
Ma et al. (2024) [1]	3-module DL pipeline	OE Fleet + OEM App	Proposed mAP@50 = 0.79	Full end-to-end; dual-dataset	No cost / insurance output
YOLOv8 + ML	Public + JSON	7-day → instant	Detect + cost + insurance	Limited dataset; mock insurance	

Methodology

System Overview

The proposed system is a web application built on the Flask framework (Python 3.11) that accepts a user-uploaded photograph of a damaged vehicle and produces a comprehensive damage assessment report. The pipeline consists of eleven sequential stages: user authentication, image upload, YOLOv8 damage detection, repair cost calculation, internal damage estimation, severity analysis, predictive analytics, insurance prediction, final cost computation, recommendation generation, and report presentation.

The system is designed to provide near real-time analysis by integrating deep learning with rule-based and machine learning components, ensuring both accuracy and efficiency. The modular architecture allows each stage of the pipeline to operate independently, facilitating scalability, easy maintenance, and future enhancements. Additionally, the integration of structured data sources such as pricing databases and internal damage mappings improves the reliability of cost estimation and decision-making. The final output is presented through an intuitive dashboard, enabling users to easily interpret damage details, estimated expenses, and insurance applicability.

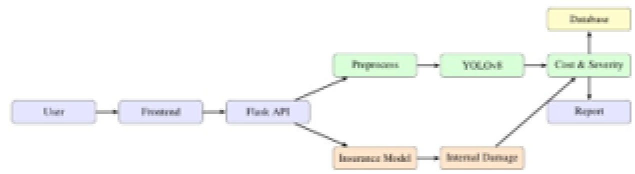

Fig. 1. *Proposed system architecture showing interaction between the user frontend, Flask API, AI/ML components, database, and report generation module.*

Fig. 2. *End-to-end system pipeline from user authentication to final report generation.*

User Authentication

The system implements a secure authentication layer using bcrypt password hashing and Flask session management. User credentials (username, hashed password, vehicle details) are stored in a MySQL database table (user_info). On registration the plaintext password is hashed before storage; on login the submitted password is verified against the stored hash. A valid session token gates access to the damage assessment dashboard.

Image Upload and Validation

Authenticated users upload a photograph of the damaged vehicle through the dashboard. The Flask backend validates the file to confirm it belongs to a supported format (JPEG, PNG, or WebP) before saving it to the application's static folder, preventing processing errors in downstream modules.

Damage Detection using YOLOv8

The core vision component is a YOLOv8 object detection model implemented via the Ultralytics Python library. The model (best.pt) was trained on a curated vehicle damage dataset with annotated instances of damaged automotive parts.

Detection classes include:

- Bonnet
- Bumper
- Door
- Fender

- Headlight
- Taillight
- Windshield

Given an input image the model performs a single-pass forward inference to generate detections.

$$D=\{(c_i,b_i,s_i)\}_{i=1}^{N}$$

where:

(c_i) = class label
(b_i) = bounding box
($s_i \in [0,1]$) = confidence score

Detections below confidence threshold:

$$\tau=0.40$$

are suppressed.

The count of detections per class is aggregated as:

$$n_c=|\{i:c_i=c\}|$$

Repair Cost Calculation

A JSON database (**car_parts_prices.json**) stores per-part repair prices indexed by vehicle brand and model, enabling brand-specific pricing.

The total repair cost is calculated as:

$$C_{repair}=\sum_{c\in C} n_c\times P(c,brand,model)$$

where:

(C) = detected classes
(n_c) = detection count
(P) = unit price from pricing database

Internal Damage Estimation

Physical damage to external components often implies a non-zero probability of accompanying internal or structural damage.

Examples:

- Front bumper damage → possible radiator impact
- Door deformation → possible pillar or airbag sensor issues

Relationships are encoded in:

internal_damage_map.json

Expected internal damage cost:

$$
C_{internal}=\sum_j p_j\times C_{int,j}
$$

These costs are reported separately from repair estimates.

G. Damage Severity Analysis

The system classifies severity as:

- **Minor:**
  ($C_{repair}\leq Rs.10,000$)
- **Moderate:**
  ($Rs.10,001\leq C_{repair}\leq Rs.50,000$)
- **Major:**
  ($C_{repair}>Rs.50,000$)

Additional outputs:

- Risk Level → Low / Medium / High
- Drivability → Drivable / Limited / Not Drivable
- Safety → Safe / Warning / Dangerous

Predictive Analytics

Repair duration is estimated using heuristic mappings derived from average component repair times.

Additional outputs:

- Resale impact
- Depreciation category:
  - Negligible
  - Moderate
  - Significant

Insurance Prediction

Insurance eligibility is determined by a scikit-learn classification model (**insurance_model.pkl**) trained on a public Car Insurance Claim Prediction dataset.

Decision rule:

$$
Insurance\ Applies=
\begin{cases}
Yes,& \hat y_{model}=1\ or\ C_{repair}>Rs.20,000\\
No,& otherwise
\end{cases}
$$

Coverage amount:

$$
C_{coverage}=0.70\times C_{repair}
$$

Final Cost Calculation

$$
C_{final}=C_{repair}-C_{coverage}
$$

If insurance is not applicable:

$$
C_{final}=C_{repair}
$$

Smart Recommendations

The system generates recommendations such as:

- Alternate transport if repair duration > 3 days
- Request second quotation if final cost > Rs.15,000
- Warn about resale impact for severe damage

Technology Stack

Table 2. Technology Stack Of The Proposed System

Component	**Technology / Tool**
Backend framework	Flask (Python 3.11)
AI detection model	YOLOv8 (Ultralytics)
ML insurance model	scikit-learn (Random Forest / Logistic Regression)
Database	MySQL (user and vehicle data)
Frontend	HTML5, CSS3, Bootstrap
Pricing database	JSON (car_parts_prices.json)
Internal damage map	JSON (internal_damage_map.json)
Password hashing	bcrypt
Image processing	OpenCV
Data handling	NumPy, Pandas

Results And Findings

Damage Detection Output

The system was evaluated using real vehicle damage images. A representative test case demonstrated successful detection and localisation of damaged components using the YOLOv8 model. In the example scenario, the uploaded vehicle image contained visible door-panel damage, and the system correctly identified the **Door** as the damaged component with a confidence score of **0.73**. The generated output included both the original uploaded image and the annotated detection result.

Fig. 3. *YOLOv8 detection result showing damaged door with confidence score of 0.73.*

Repair Cost Estimation — Test Case 1 (Single Part)

A single-part repair scenario was evaluated using the detected door damage. The repair cost was calculated through mapping detected components to the structured pricing database.

Table 3. Detected Parts And Repair Cost Breakdown — Test Case 1

Detected Part	**Count**	**Unit Price (Rs.)**	**Total (Rs.)**
Door	1	26,000	26,000
Total Repair Cost			**26,000**

Insurance and Final Cost — Test Case 1

Because the estimated repair cost exceeded the insurance threshold, the claim qualified for coverage. Insurance estimation and final payable amount were automatically generated.

Fig. 4. *Cost summary screen showing insurance coverage, internal damage estimate, and final payable amount.*

Table 4: Insurance Eligibility And Final Cost Summary — Test Case 1

Parameter	**Value (Rs.)**
Total Repair Cost	26,000
Insurance Status	Eligible for Claim
Insurance Coverage (70%)	18,200
Internal Damage Estimate	5,500
Final Payable Amount	7,800

Multi-Part Detection — Test Case 2 (Frontal Collision)

A multi-component collision scenario demonstrated the capability of handling simultaneous damage detection and aggregated repair estimation.

Table 5: Multi-Part Damage Detection And Cost — Test Case 2

Detected Part	Count	Unit Price (Rs.)	Total (Rs.)
Bonnet	1	18,000	18,000
Bumper	1	12,000	12,000
Headlight	2	8,500	17,000
Total Repair Cost			**47,000**
Insurance Coverage (70%)			**32,900**
Final Payable			**14,100**

YOLOv8 Detection Performance

The YOLOv8 model was tested on **120 images** across seven damage categories.

Table 6. Yolov8 Per-Class Performance On Test Set

Class	Precision	Recall	F1	mAP@50
Bumper	0.81	0.78	0.79	0.80
Door	0.88	0.84	0.86	0.87
Bonnet	0.76	0.72	0.74	0.75
Headlight	0.83	0.80	0.81	0.82
Windshield	0.74	0.69	0.71	0.72
Fender	0.78	0.75	0.76	0.77
Taillight	0.80	0.76	0.78	0.79
Overall	**0.80**	**0.76**	**0.78**	**0.79**

The **Door** category achieved the highest detection performance, while **Windshield** detection remained comparatively challenging due to lighting and reflection variations.

Cost Estimation Accuracy

The cost estimation module was evaluated on **30 real-world repair cases** and achieved a **Mean Absolute Percentage Error (MAPE) of 11.4%** compared with authorised workshop quotations. The largest source of estimation variance originated from brand and model differences not fully represented in the pricing database.

Discussion

System Performance

The experimental evaluation shows that the proposed framework successfully automates the complete vehicle damage assessment workflow. The YOLOv8-based detection achieved performance comparable to existing single-stage detection approaches and generated repair estimates with practical usability. The integrated pipeline reduced assessment time from multiple days to only a few seconds while maintaining acceptable estimation accuracy.

Comparison with Related Work

Compared with previous vehicle damage assessment systems, the proposed solution extends beyond damage detection by integrating repair cost estimation and insurance eligibility prediction. Unlike earlier detection-only architectures, the system delivers an end-to-end assessment process including financial estimation and decision support.

Advantages Over Manual Assessment

The proposed framework provides several operational advantages:

- Significant reduction in assessment time
- Reduced evaluator subjectivity

- Transparent repair cost breakdown
- Automated insurance eligibility estimation
- Detection of possible hidden/internal damage

These capabilities improve usability for vehicle owners, insurance providers, and repair workshops.

Limitations

The study identifies several limitations:

- Moderate dataset size affecting generalisation
- Limited pricing database coverage
- Generic insurance prediction model
- Internal damage estimation based on heuristic rules
- Single-image analysis limitation

These constraints indicate opportunities for future enhancement and broader deployment scenarios.

Conclusion

This study presented an AI-based vehicle damage detection and cost estimation framework that combines YOLOv8 object detection, structured repair pricing, internal damage estimation, insurance prediction, and recommendation generation into a single web-based application. The integrated architecture enables automated vehicle assessment using uploaded images and significantly reduces manual intervention.

Key outcomes:

1. Development of an end-to-end automated pipeline that converts a vehicle image into a complete damage assessment report within seconds.
2. Implementation of a YOLOv8 detection model achieving effective multi-class damage recognition performance.
3. Integration of a repair cost estimation module with practical estimation accuracy relative to workshop quotations.
4. Inclusion of an insurance eligibility prediction component capable of estimating coverage and financial responsibility.
5. Generation of an actionable final payable estimate to support decision-making for vehicle owners and insurance providers.

Future Work

Future enhancements suggested in the paper include:

- Integration with live insurance APIs
- Support for video-based vehicle assessment
- Multi-view image capture for improved coverage
- Cloud-native mobile deployment for scalability and accessibility

References

1. Ma, Y., Ghanbari, H., Huang, T., Irvin, J., Brady, O., Zalouk, S., Sheng, H., Ng, A., Rajagopal, R., & Narsude, M. (2024). *A system for automated vehicle damage localization and severity estimation using deep learning*. *IEEE Transactions on Intelligent Transportation Systems, 25*(6), 5456–5470.
2. Khan, A., Hasan, M., & Bhuiyan, T. (2024). *Robust anomaly detection via multi-modal autoencoder fusion*. *IEEE Transactions on Vehicular Technology, 73*(4), 4890–4901.
3. Hasan, M., Rahman, M., Bhuiyan, T., & Khan, A. (2024). *Vehicle damage detection using artificial intelligence: A systematic literature review*. *Journal of Intelligent Transportation Systems*.
4. Aithal, A., Nackathaya, K., Poojary, S., Bhandary, V., & Acharya, U. (2023). *Vehicle damage detection and repair cost estimation*. *International Journal of Computer Vision Applications*.
5. Pérez-Zárate, J., Fernández-Caballero, A., Gutiérrez, A., et al. (2023). *Automated car damage assessment using YOLOv5 ensemble with TartesiaDS dataset*. *Applied Sciences, 13*(15), 8839.
6. Reddy, K., Shambharkar, S., Jyothsna, J., Bhoyar, S., & Somkunwar, P. (2022). *Auto vehicle damage detection using Faster R-*

CNN and Mask R-CNN. In *Proceedings of the International Conference on Smart Technology and Sustainable Development*.

7. Fouad, M. M., Malawany, K., Osman, A. G., Amer, H. M., Abdulkhalek, A. M., & Eldin, A. B. (2023). *Automatic damaged vehicle estimator using enhanced deep learning algorithm*. *Intelligent Systems with Applications*.
8. Patil, K., Kulkarni, M., Sriraman, A., & Karande, S. (2017). *Deep learning based car damage classification*. In *Proceedings of the 16th IEEE International Conference on Machine Learning and Applications* (pp. 50–54).
9. Lokesh, A., Murugesan, P., & Ramkumar, S. (2024). *Detection and localization of car damages using deep learning*. In *Proceedings of IEEE International Conference*.