

Vehicle Damage Detection and Repair Cost Estimation System

Mekha S Nair¹, Preethi Thomas²

¹Department of Computer Applications, Musaliar College of Engineering & Technology, Pathanamthitta, Kerala, India
Email: [mekhasanthoshnair\[at\]gmail.com](mailto:mekhasanthoshnair[at]gmail.com)

²Professor, Department of Computer Applications, Musaliar College of Engineering & Technology, Pathanamthitta, Kerala, India

Abstract: *The system applies deep learning (using YOLO/CNN models) to identify and classify vehicle damages such as dents, scratches, and broken parts from accident images. It automatically estimates repair costs based on detected damage severity and part type, referencing a predefined pricing chart. Deployed as a Flask-based web interface, the software supports use by insurance firms, repair workshops, and vehicle owners, enabling faster and more transparent cost assessments.*

Keywords: Face Recognition, OTP Authentication, Voting Security, Biometric Verification, Fraud Detection, Django, OpenCV

1. Introduction

An AI-based vehicle damage detection and repair cost estimation system is an intelligent solution designed to automate the process of assessing vehicle condition after accidents or wear. Traditionally, damage inspection requires manual evaluation by experts, which can be time-consuming, subjective, and prone to inconsistencies. By integrating computer vision and deep learning techniques, this system analyzes images or videos of vehicles to automatically identify and classify different types of damage such as dents, scratches, broken headlights, cracked bumpers, and deformed body panels. Advanced object detection and image segmentation models are used to precisely locate the damaged regions and determine their severity level. Once the damaged parts are detected, the system evaluates the extent of the damage by comparing visual features with trained datasets and predefined severity scales. Based on this analysis, it estimates repair costs by referencing historical service records, spare-part pricing databases, or predefined repair rules. This enables the system to generate quick and approximate cost predictions without requiring a physical inspection. Such automation significantly speeds up the claims and repair process, improves cost estimation accuracy, and ensures transparency in decision-making. This technology is particularly beneficial for insurance companies, automobile service centers, and vehicle owners, as it reduces processing time, minimizes human error, and supports remote damage assessment. With the increasing availability of smartphones, CCTV footage, and dashboard cameras, AI-driven damage detection systems are becoming a practical and scalable solution for modern automotive maintenance and insurance workflows, paving the way for faster, smarter, and more reliable vehicle damage evaluation.

2. Related Works

In recent years, significant research has been conducted in the field of automated vehicle damage detection and repair cost estimation using artificial intelligence and computer vision techniques. Early studies primarily focused on applying Convolutional Neural Networks (CNNs) to classify whether a vehicle image contained damage or not. However, these

approaches were limited because they could not accurately localize the damaged regions or differentiate between types of damage such as scratches, dents, and cracks. To overcome these limitations, researchers began adopting advanced object detection and instance segmentation models such as YOLO, Faster R-CNN, and Mask R-CNN. These models enabled precise localization of damage by predicting bounding boxes and segmentation masks around affected areas. For example, Mask R-CNN has been widely used in vehicle damage assessment due to its ability to perform pixel-level segmentation, which improves the detection of fine-grained damage such as small scratches and broken components. One of the major challenges in this domain was the lack of large, annotated datasets for training deep learning models. To address this issue, researchers introduced the Car Damage Detection (CarDD) dataset, which contains thousands of high-resolution vehicle images annotated with multiple categories of damage. This dataset has become an important benchmark for evaluating modern detection and segmentation models and has significantly accelerated research in this area. Recent studies have also explored the integration of AI systems into real-world applications such as insurance claim automation. Automated damage assessment frameworks combine image processing, deep learning, and database systems to streamline the claim verification process and reduce manual inspection time. These systems demonstrate how AI can improve efficiency, reduce operational costs, and enhance transparency in insurance workflows.

3. Outlined Method

The proposed vehicle damage detection and repair cost estimation system follows a structured methodology that integrates image processing, deep learning, and cost analysis. Initially, vehicle images are acquired from users through a web interface or camera input and are forwarded to the preprocessing module. In this stage, the images are resized to a standard resolution, normalized, and enhanced to ensure consistent input quality for the model. The pre-processed images are then passed to a YOLOv11-based object detection model, which identifies and localizes damaged regions such as dents, scratches, and broken components using bounding

boxes. To eliminate redundant detections, Non-Maximum Suppression is applied to retain only the most accurate predictions. The detected damages are further analyzed by a classification module that determines the severity level-minor, moderate, or severe- based on the size and location of the affected area. Following this, the system estimates the repair cost by mapping each detected damage to a predefined pricing database or historical repair records. Finally, the processed results, including highlighted damage regions, severity labels, confidence scores, and the total estimated repair cost, are presented to the user through an interactive web interface. This end-to-end methodology enables fast, automated, and consistent vehicle damage assessment without the need for manual inspection.

3.1 Requirement Analysis

The requirement analysis phase focuses on identifying the limitations of traditional vehicle damage inspection and manual repair cost estimation processes. Conventional methods rely heavily on human inspection, which is time-consuming, prone to subjective errors, and often leads to inconsistent damage assessments and inaccurate cost estimates. Additionally, communication between vehicle owners, service centers, and insurance companies is often slow due to the lack of a centralized digital platform. To overcome these challenges, the proposed system introduces an AI-based automated damage detection and cost estimation solution. The system processes vehicle images, detects damaged areas, and generates repair cost estimates without requiring manual inspection. This reduces processing time, improves accuracy, and ensures consistent evaluations.

a) System Design

The system architecture is designed as a modular structure where different components interact with each other through a centralized database. The major modules of the system include:

- **Web Interface Module:** This module allows users to upload vehicle images and view results. It provides an intuitive interface for image submission and displays detected damages with cost estimates.
- **Image Acquisition Module:** This module captures images either through file upload or direct camera input and forwards them to the preprocessing stage. It ensures that only supported image formats are accepted and properly stored.
- **Data Preprocessing Module:** The preprocessing module prepares images for model inference by performing resizing, normalization, and noise reduction. This step ensures that all images meet the input requirements of the YOLOv11 model and improves detection accuracy.
- **Damage Detection Module:** This is the core component of the system. It uses the YOLOv11 deep learning model to detect and localize damaged regions in the vehicle image.
- The model processes the image in a single forward pass and generates bounding boxes with confidence scores for each detected damage.
- **Damage Classification Module:** After detection, this module categorizes the detected damage based on severity levels such as minor, moderate, and severe. The

classification is performed using features such as damage area, location, and detection confidence.

- **Cost Estimation Module:** The cost estimation module maps detected damage types and severity levels to a predefined pricing database. It calculates both individual repair costs and the total estimated cost for all detected damages.

All these modules are interconnected and communicate with a centralized database system that stores user data, content details, and generated outputs.

b) Development

The development of the Vehicle Damage Detection and Repair Cost Estimation System involved designing and implementing multiple modules to automate the detection and analysis of vehicle damage. The system was built using Python, PyTorch, and OpenCV for deep learning and image processing, while Flask was used to develop the web-based interface. A YOLOv11 object detection model was trained on labeled vehicle damage images to identify damaged parts such as dents, scratches, and broken components. The detected damage was then classified based on severity and mapped to a predefined pricing chart to estimate repair costs. All modules were integrated to create a complete pipeline that allows users to upload images and receive automated damage analysis and cost estimation results in real time.

c) Integration & Testing

The integration phase involved combining all modules of the vehicle damage detection system- including image upload, preprocessing, YOLOv11 detection, damage classification, cost estimation, and result visualization- into a single unified application using a Flask backend. This ensured smooth data flow from the user interface to the deep learning model and back to the output display. Testing was conducted to verify the correctness and performance of the system. Unit testing was used to validate individual modules, while integration testing ensured that the modules worked properly when connected. System testing was performed using real vehicle damage images to evaluate detection accuracy and cost estimation reliability. These testing processes helped identify errors, improve performance, and ensure that the system functions accurately and efficiently in real-world usage.

4. Evaluation & Optimization

The evaluation phase was carried out to measure the performance, accuracy, and reliability of the vehicle damage detection system. The trained YOLOv11 model was tested using a separate test dataset to assess metrics such as precision, recall, and mean Average Precision (mAP). The predicted damage regions and severity levels were compared with ground truth annotations to evaluate detection correctness and cost estimation accuracy. Based on the evaluation results, optimization techniques were applied to improve system performance. These included fine-tuning hyperparameters such as learning rate, batch size, and number of epochs, as well as adjusting confidence and Non-Maximum Suppression thresholds to reduce false detections. Image preprocessing steps and data augmentation were also refined to enhance model generalization. As a result, the optimized system achieved improved detection accuracy,

faster inference time, and more reliable repair cost predictions in real-world scenarios.

4.1 Machine Learning Approach

The system uses a supervised deep learning approach to automatically detect and analyze vehicle damage from images. A large dataset of vehicle images containing various types of damage—such as dents, scratches, cracks, and broken parts—was collected and manually annotated with bounding boxes and labels. This labeled dataset was used to train a YOLOv11 object detection model, which learns to identify damaged regions by extracting visual features such as shape, texture, and color variations. During training, the model adjusts its internal parameters through backpropagation and gradient optimization to minimize detection and classification errors. Once trained, the model can accurately detect damaged parts in unseen images and classify their severity. The output of the model is then used by a rule-based or data-driven cost estimation module to predict repair expenses. This machine learning approach enables the system to improve its accuracy over time as more training data becomes available and helps automate vehicle inspection with high speed and consistency.

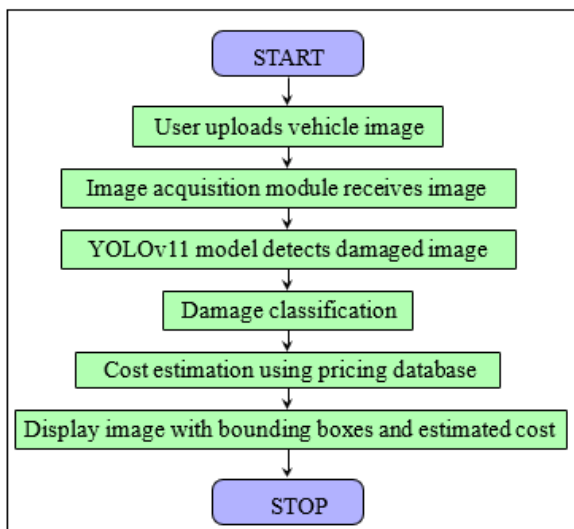


Figure 1: ROC Curve for Vehicle Damage

4.2 Dataset Description

The dataset used in this project consists of images of vehicles with different types of visible damage collected from publicly available sources and custom image captures. The dataset includes various damage categories such as dents, scratches, cracked windshields, broken headlights, and bumper damage. Each image was manually annotated using bounding boxes to mark the exact location of damaged regions and labeled according to the type of damage. The dataset contains images captured under different lighting conditions, angles, backgrounds, and vehicle models to ensure that the trained model can generalize well in real-world scenarios. To train and evaluate the YOLOv11 model effectively, the dataset was divided into three subsets: training, validation, and testing.

5. Result & Discussion

5.1 System Performance and Functionality

The performance of the Vehicle Damage Detection and Repair Cost Estimation System was evaluated based on accuracy, speed, and reliability. The YOLOv11 model demonstrated high detection accuracy in identifying various types of vehicle damage and provided consistent results across different image conditions. The system achieved fast processing times, enabling near real-time analysis of uploaded images through the web interface. Efficient preprocessing and optimized model parameters helped reduce inference latency while maintaining precision. Overall, the system performed reliably during testing, handling multiple images and different damage scenarios with stable and accurate output, making it suitable for practical deployment in insurance and service center environments.

5.2 Test Cases and Outcomes

Several test cases were designed to evaluate the functionality, accuracy, and reliability of the Vehicle Damage Detection and Repair Cost Estimation System. These test cases covered different scenarios such as uploading valid vehicle images, detecting multiple damages in a single image, handling images with no visible damage, and processing images with varying resolutions and lighting conditions. The system successfully detected damaged regions, classified their severity, and generated corresponding cost estimates in most cases. When high-quality images were provided, the detection accuracy and cost estimation were highly reliable.

5.3 Comparative Analysis with Existing Systems

A comparative analysis evaluates the differences between the traditional vehicle damage assessment approach and the proposed AI-based vehicle damage detection and repair cost estimation system. This analysis highlights how the proposed system improves efficiency, accuracy, and scalability compared to existing methods used by insurance companies and vehicle service centers.

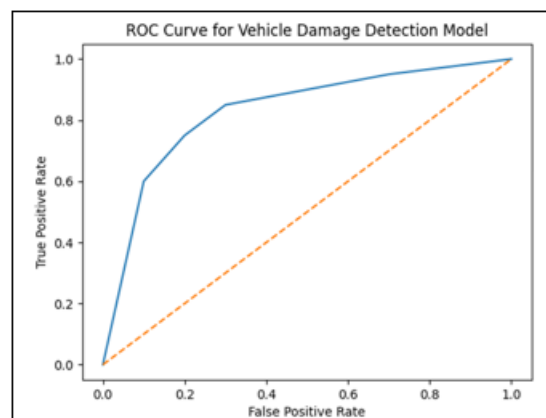


Figure 2: ROC Curve for Vehicle Damage

6. Conclusion

The Vehicle Damage Detection and Repair Cost Estimation

System was successfully designed and implemented using deep learning and computer vision techniques. By utilizing a YOLOv11-based object detection model, the system is capable of accurately identifying damaged vehicle parts such as dents, scratches, and broken components from uploaded images. The integration of image preprocessing, damage classification, and cost estimation modules enabled the system to provide automated and real-time analysis through a user-friendly web interface. The developed system significantly reduces the time and effort required for manual vehicle inspection while improving transparency and consistency in repair cost estimation. Experimental results demonstrated reliable detection accuracy and fast processing speed, making the solution suitable for practical use in insurance claim processing and vehicle service centers. Although minor limitations were observed in detecting very small or partially occluded damages, the overall performance confirms that the proposed system is an effective and scalable approach for automated vehicle damage assessment. Future enhancements such as expanding the dataset, improving segmentation accuracy, and integrating real-time video analysis can further increase the system's robustness and applicability in real-world scenarios.

References

- [1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi (2016). "You Only Look Once: Unified, Real-Time Object Detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*,
- [2] K. He, G. Gkioxari, P. Dollár, and R. Girshick (2017). *ML-“Mask” R-CNN*," *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*,
- [3] S. A. Perez-Zarate et al (2024). "Automated Car Damage Assessment Using Computer Vision: Insurance Company Use Case," *Applied Sciences*, vol. 14, no. 20
- [4] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, and Y. Sekimoto (2021) "RDD2020: An Annotated Image Dataset for Automatic Road Damage Detection Using Deep Learning," *Data in Brief*
- [5] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao (2020). "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv preprint arXiv:2004.10934*,
- [6] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao (2022) "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," *arXiv preprint*.
- [7] A. Paszke et al. (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library," *Advances in Neural Information Processing Systems (NeurIPS)*
- [8] G. Bradski (2008). OpenCV Library," *Dr. Dobb's Journal of Software Tools*
- [9] I. Goodfellow, Y. Bengio, and A. Courville (2016). *Deep Learning*," MIT Press
- [10] J. Wang, X. Yin, L. Wang, and L. Zhang (2020). *Hashing-Based Non-Maximum Suppression for Crowded Object Detection*," *arXiv preprint*