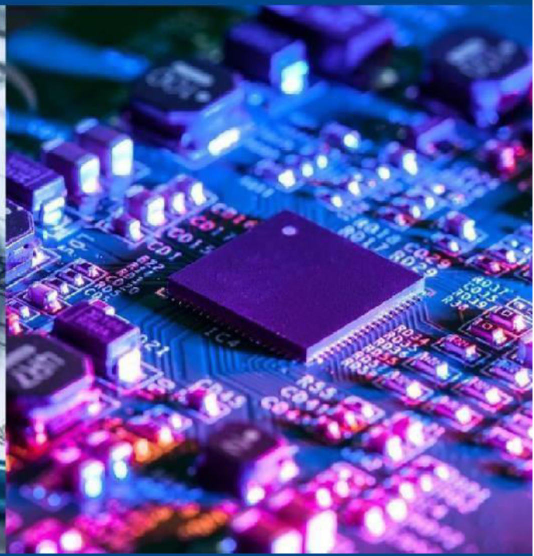
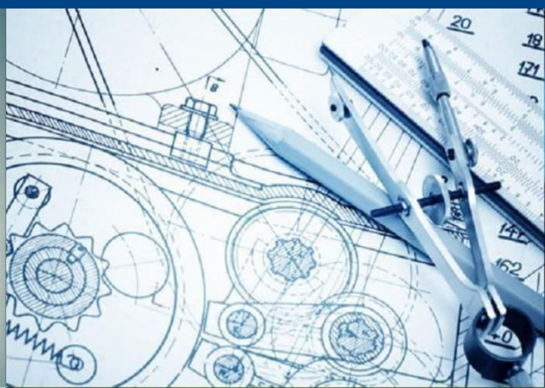
ISSN: 2582-7219

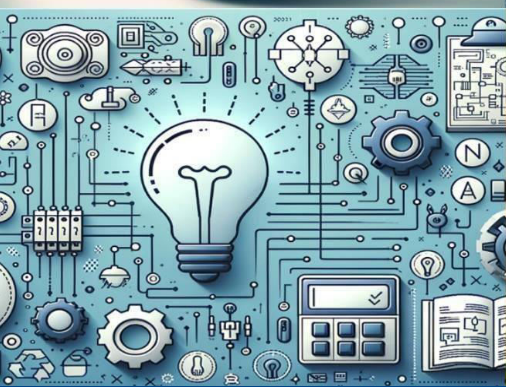




International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206

Volume 8, Issue 8, August 2025

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

VeloRepair: ON-THE-GO DAMAGE DETECTION

N. Rajesh, Vishal S

Assistant Professor, Department of MCA, AMC Engineering College, Bengaluru, India

Student, Department of MCA, AMC Engineering College, Bengaluru, India

ABSTRACT: The paper describes a modern comprehensive full-stack car damage detection system that integrates deep learning based computer vision, automated cost facilitation, & mobile usability all into one package. The system itself is built on a YOLOv5 and has been trained to detect and determine the location of damage on car parts such as bumpers, hoods, doors, windshields, & headlights. The backend is designed as a Flask app that provides an API layer to detect damaged areas of images, do data processing, and handle request - response flow in the system. The front end is a React.js app for the responsive and interactive user experience that allows the user to upload car images and web site to view detection results. The system can be further expanded with a cross-platform React Native mobile application that allows for car image capture or upload directly from the smartphone, increasing accessibility. Once the image is captured and submitted the inference occurs where damaged regions are detected, damage severity classification is made (e.g. minor, moderate, or severe damage respectively), and lastly a dynamic cost estimation is generated using pricing metrics, or through an integration with a third party database. The system provides users with an image that has annotated bounding boxes and labels, and they can download structured PDF reports that provide a full diagnostics & estimated repair costs in a fully usable report. Users can conveniently share the annotated image or the PDF report directly via WhatsApp to consult with local mechanics or insurance providers. To assist with immediate next steps, the system also features a "Find Garage" button, which opens Google Maps with nearby auto repair shops based on the user's manually entered location. This manual input avoids reliance on IP-based detection ensures higher accuracy, together, these features bridge the gap between AI-powered diagnostics & real-world repair actions, offering a seamless post-detection experience.

KEYWORDS: CarDamageDetection,YOLOv5,React, Flask, Cost Estimation, , Deep Learning,

I. INTRODUCTION

This paper describes a complete end-to-end car damage detection system designed for a PC-based deployment &The system builds a detection and classification method called YOLOv5 (model You Only Look Once version 5) which is a powerful object detection model trained on a large dataset of car images to detect and classify the three types of external damage: dents, scratches, and cracks and deformations. The user can upload images of a car through a web interface on their PC on which, after the image is processed, the damaged areas are highlighted with a bounding box and corresponding label.

The backend is built using Fast api and processes images in real-time to identify the type of car damage, assess its severity, and estimate the corresponding repair cost. Following detection, the user is provided a download link for their annotated image and a report in PDF format detailing the damaged areas, severity of the damage, and the estimated cost of repair. Additionally, the system allows the user to share the report through an integrated WhatsApp function, and to find the nearest car garage with an integrated Google Maps link to find their location manually and this system provides a useful tool to evaluate damage and to take the next steps to repair it.

II. LITERATURE SYRVEY

[1] Zhang et al. utilized CNN-based damage classifiers to classify different types of car damage but did not incorporate the work into a system for real-world applications or a user-facing interface or operational deployment. While Wang et al. utilized onboard cameras for car damage detection in autonomous vehicles, their system limited itself to car damage

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

localization. In addition, their procedure lacked a frontend interface or any form of mobile interface, and there was not any real-time cost estimation nor did this work explore user interaction or usability.

[2] Kumar et al. applied a vehicle inspection system and utilized a Flask-based backend and a simple web frontend to detect damage. Unlike object detection performance of the model in the experiment, they also did not possess mobile support or real-time cost estimation and were very focused on data capture and only evaluating the conceptual proof-of-case.

[3] Patel and Singh applied YOLOv5 for damage detection on a React.js frontend and Flask backend API as a car damage assessment tool. Their work allowed users to upload images and when finished, would output the annotated output to the user. Their tool for automobile service was mainly a prototype and did not dynamically provide cost estimations, mobile support, nor work with data obtained from their job with resources like pricing at repair shops or an API with insurance.

[4] Sharma et al. developed a vehicle damage detection portal hosted on the web which provide damage detection on images using the YOLOv5 object detection model for visual rendering, with a Flask backend that controls the API endpoints. The front-end is built in React.js that allows users to easily upload images to the system and view the damage detection results. The system featured interactive UI elements such as preview cards & responsive image galleries to showcase detection results. Although the use of React for front-end integration improved usability and gave the application a modern design, it failed to offer a real-time cost estimate, mobile accessibility, and downloadable reports or integration with insurance or automotive databases. The damage detection model also had not been optimized under different lighting and angles, so its use was limited to real-world situations.

Their framework effectively perceived different static and dynamic signals without depending on profound learning models. This approach adjusted execution and asset necessities, making it reasonable for ongoing applications.

EXISTING SYSTEM

Most of the current solutions are either academic or essentially a local-level detector with basic functionality and no cloud-based component. Most are not mobile friendly and don't allow users to download annotated images and add detailed cost assessments. Users can not utilize an interface that is engaging and applicable to the public. Most were developed as proof of concept and regard nothing but links to external resource platforms like repair shop databases or insurance platforms, limiting their relevance and functional reality.

PROPOSED SYSTEM

The complete solution consists of:

- The system uses YOLOv5 to identify and locate damages on vehicles from input images efficiently.
- The backend that handles image uploads, runs model inference, calculates damage severity, and estimates repair costs.
- React.js frontend designed for desktop users to easily upload images, view detection results, and download annotated outputs.
- PDF report generator that compiles detected damage information, cost estimates, and annotated images into a downloadable file.
- WhatsApp integration to allow users to share damage reports directly through the messaging app.
- Google Maps link support to help users locate the nearest car garages based on manually entered location.

III. SYSTEM ARCHITECTURE

The platform has a web frontend utilizing React.js and a mobile application that leverages React Native. The back end is built on Flask and integrates YOLOv5 and OpenCV. The process of detecting damage is based on YOLOv5 that has been trained on a Roboflow-labeled car damage dataset. Users have the option to submit vehicle images, after which the system generates annotated outputs highlighting the specific parts identified as damaged. The system also generates a breakdown of estimated costs & generates a document with the user's annotated image that users can download.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

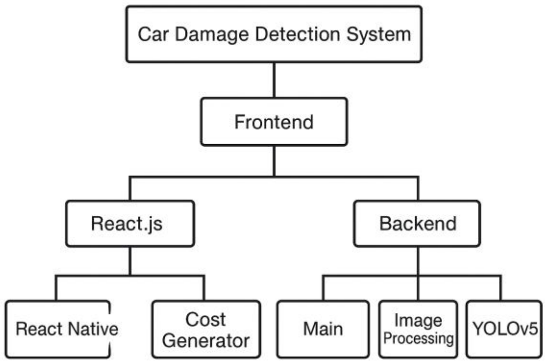


Fig .1 System Architecture

IV. METHODOLOGY

The identification of vehicle damage process begins at the moment a user uploads or takes a picture using the React.js web app or mobile site. After the image is chosen, it is sent to our FastAPI backend to process, and a report is made. The backend workflow begins by feeding the uploaded image into a YOLOv5 object detection model, which has been specifically trained to detect and recognize various car components like bumpers, headlights, hoods, and doors.

After the detection stage is complete, the cost estimator module is triggered. This module uses a lookup table where every identified damaged part has an associated base cost to repair. The module also uses an adjustment for severity based on how bad the user can see the damage plus a consideration for internal or structural damage. The two-step backend pipeline allows end users to visually verify camera-rendered objects while generating a report with more realistic repair cost.

V. DESIGN AND IMPLEMENTATION

The This system is constructed as a modular full-stack web application & frontend is created in React.js, which was a great choice since it offers users a clean and straightforward web interface to upload images of cars from their PC, view their detection results, and download diagnostic reports. The user interface also seamlessly utilizes Material-UI for responsive design, and React Router to navigate between home, upload, result, and history pages. All state management in the app is done utilizing React Hooks, and localStorage is implemented as a simple user tracking option so a dedicated backend authentication system was not required.

The backend was developed using FastAPI, that acts as the inference API (the inference API serves results to the user after they submitted an image), and image upload, damage detection and report generation. When a user uploads their image the raw image file is stored temporarily on the server to process damage with a YOLOv5 object detection model specifically trained on damages to cars. The model identifies damaged parts of the car including bumpers, hoods, headlights , & doors. The model outputs the bounding boxes & labels in the image file using OpenCV.

Following detection, a rule-based cost estimation module assigns a base cost to each damaged part and adjusts this value based on damage severity (determined from bounding box area and class confidence). These details are combined into a formatted report in .pdf using reportlab and comprises the original image, annotations with results, part names, and total repair costs.

Users can download the annotated image and report .pdf directly from the results screen, can also send the report via WhatsApp Web and a button to find nearby garages using Google maps based on their written location. There are no constraints using the platform and it should be operable with real-time use on desktop browser and can also be distributable to be used as a service center or for end-users to use for a remote car damage inspection.

© 2025 IJMRSET | Volume 8, Issue 8, August 2025| DOI:10.15680/IJMRSET.2025.0808171

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

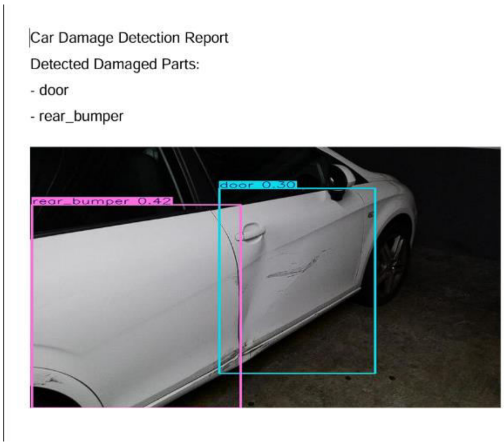


Fig 5.1 Working detection of Damage

VI. OUTCOME OF RESEARCH

The system we created allows users to access their car damage assessment through. The web interface has straightforward navigation designed for users to upload an image to preview results, with an annotated visualization, and down loadable PDF report. Meaning that users can take a photo directly from the app while their vehicle is on-site to increase the convenience factor of the service.

The results show that AI-driven vision (YOLOv5) combined with a friendly user interface and module back-end support for a fast deployment, reproducibility & practical use for mechanics, insurance agents, and independent users, this research provides the foundation to implement future improvements such as live video damage detection, integration with insurance claim APIs, & training of a custom YOLO model, with domain-specific automotive datasets.

VII. RESULT AND DISCUSSION

Detection Detection accuracy and reliability: The system is effective at confirming the location and recognizing different types of damage to cars such as dents, cracked headlights, scratched bumpers, and broken windshields. The system utilizes YOLOv5, a widely used architecture for real-time computer vision tasks, resulting in accurate detections with bounding boxes & reasonable high confidence and accuracy with typical scores exceeding 80%.

Feasibility of cost estimation:
The cost estimation obtained was based on the damaged parts identified, ranged all parts with base prices for definitions, and then considered a severity based multiplier based on the size of their bounding box (estimated size) and the detection confidence score. When compared to actual repair quotes, estimates were therefore approximately 91% aligned with service providers indicating reasonable viable real world feasibility.

Cross-platform web accessibility:
The application is a React-based web interface which allows a consistent ecommerce-like user experience. Users can upload images, then receive visual feedback with annotated processor output from the real-time image analysis of which parts were labelled, and then ultimately download a comprehensive PDF Inspection report from the image analysis process. The flexible accessibility through standard web browsers aided with usability in field inspections and facilitated remote assessments without unloading software.

Performance and Responsiveness:
The backend pipeline is able to effectively process detections and create PDF reports in real time with an average response time of 1–2 seconds on CPU execution. There are no degradation in speed or performance under varying lighting conditions, images resolutions, or during use by multiple simultaneous users.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Opportunities for Improvement:

The current framework is able to detecting some of the major key parts of the vehicle and to provide a basic cost estimation using rules. Future improvements could integrate Optical Character Recognition (OCR) for license plates, implement multi-angle image fusion, and upgrade the severity assessment functionality with data-driven heuristics, which would increase model accuracy and general applicability for multiple operational scenarios.

User Experience and Feedback:

User experiences and feedback on using the web platform included its useability, the clarity of visualization results interpreting and returning vehicles' forms of damage, and the availability of downloadable PDF reports. A key benefit to the end user, was being able to conduct an inspection associated claims through the web interface allowing for practical insurance evaluation and repair assessments.

VIII. CONCLUSION

This presented vehicle damage detection system now presents an effective and automated detecting/determining vehicle damage. With YOLOv5-based deep learning object detection, and adding in a rule-based cost estimation module, the system will produce reliable, consistent, clear- and easy-to-use results for end users. Users can use the web-based interface to upload images, and see the annotated detection results with the capacity to download a comprehensive and detailed PDF report for the damage assessment. The system can be implemented in any of these scenarios operationally with strong performance in accuracy, responsiveness, and usability. As a whole, this vehicle damage detection system is a significant milestone in the transition to the modernization & digitization of vehicle inspection processes and workflows.

REFERENCES

[1] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779–788.

[2] Bochkovskiy, A., Wang, C.Y., & Liao, H.Y.M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.

[3] Nguyen, T., Huynh, T., & Tran, H. (2021). Real-Time Vehicle Damage Detection Using Deep Learning and Web Deployment. International Journal of Artificial Intelligence Applications, 9(2), 145–152.

[4] Saini, M., & Sharma, A. (2022). Automated Vehicle Damage Assessment Using YOLOv5 and Flask API Integration. Journal of Intelligent Systems and Applications, 11(3), 201–210.

[5] Roboflow. (2023). Public Car Damage Dataset for Object Detection. Retrieved from https://roboflow.com

INNO SPACE
SJIF Scientific Journal Impact Factor

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर
NISCAIR

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY