© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715302

Smart-Claim: AI-Based Vehicle Damage Detection and Estimation Using YOLO And Large Language Models

ASAN NAINAR M[1], JEEVANANDAN B K[2]
[1, 2]*Department of Computer Applications, SRM Valliammai Engineering College*

Abstract- Vehicle damage inspection is an essential step in insurance claim processing and vehicle repair management. Traditional inspection procedures rely on manual verification by surveyors, which often results in delays, human errors, and inconsistencies in repair estimation. To address these challenges, this research proposes Smart-Claim, an artificial intelligence-based system designed to automate vehicle damage detection and provide repair estimation insights using computer vision and large language models. The system utilizes the YOLO object detection model to identify damaged regions in vehicle images uploaded by users. The detected damage information is analyzed using NVIDIA Nemotron-3 API powered by the Llama 3.1 90B large language model to generate intelligent estimation insights. The application is implemented using React for the frontend, Python Flask for the backend, and MySQL for database management. Experimental results demonstrate that the proposed system significantly reduces inspection time and enhances efficiency in insurance claim evaluation.

Index Terms- Artificial Intelligence, Computer Vision, Vehicle Damage Detection, YOLO, Insurance Claim Automation, Deep Learning.

I. INTRODUCTION

Vehicle accidents are a frequent occurrence in modern transportation systems, resulting in various levels of vehicle damage that must be inspected before repairs can begin. In conventional insurance claim systems, vehicle damage inspection is carried out manually by surveyors who analyze affected components and estimate repair costs.

Although this process has been widely used for decades, it is time-consuming and prone to inconsistencies due to differences in human judgment. Additionally, delays in inspection can negatively affect customer satisfaction and increase operational costs for insurance companies and service centers.

Recent developments in artificial intelligence and computer vision technologies have opened new opportunities for automating image-based inspection systems. Computer vision models can analyze images and detect objects or patterns with high accuracy. In particular, deep learning-based object detection models such as YOLO (You Only Look Once) have gained popularity due to their real-time detection capabilities and high performance.

The Smart-Claim system integrates computer vision with large language model reasoning to automate vehicle damage detection and estimation analysis. By combining YOLO-based detection with LLM-based reasoning using NVIDIA Nemotron API powered by the Llama 3.1 90B model, the system provides intelligent insights into damage severity and repair requirements. This integration improves efficiency and supports automated decision-making in insurance claim evaluation.

II. RELATED WORK

Various research studies have explored the use of machine learning and computer vision techniques for vehicle damage detection. Early approaches relied on traditional image processing methods such as edge detection, thresholding, and segmentation. While these techniques were useful for simple tasks, they often struggled to detect complex damage patterns under varying lighting conditions.

The emergence of deep learning significantly improved object detection accuracy. Models such as Faster R-CNN, Single Shot Detector (SSD), and YOLO introduced convolutional neural networks capable of detecting objects within images. Among these models, YOLO became widely used because it performs detection in a single forward pass through the neural network, enabling real-time performance.

Recent research has also focused on integrating artificial intelligence systems with decision-support tools.

However, many existing systems are limited to damage detection and do not provide contextual analysis or estimation insights. Large language models provide an opportunity to enhance such systems by interpreting detection results and generating reasoning-based recommendations.

The proposed Smart-Claim system extends existing research by combining YOLO-based damage detection with large language model analysis using NVIDIA Nemotron and Llama models. This hybrid architecture improves the ability of the system to not only detect damage but also generate meaningful estimation insights.

III. PROPOSED SYSTEM

The Smart-Claim system is designed to automate vehicle damage detection and estimation using computer vision and artificial intelligence techniques. The system workflow begins when a user uploads an image of a damaged vehicle through a web-based interface. The uploaded image is processed by the backend server, which performs preprocessing operations such as resizing, normalization, and noise reduction.

The processed image is then analyzed using the YOLO object detection model. YOLO identifies damaged regions such as dents, scratches, and broken vehicle components. The coordinates and classification results generated by YOLO are extracted and forwarded to the analysis module.

The analysis module communicates with the NVIDIA Nemotron-3 API powered by the Llama 3.1 90B large language model. The LLM analyzes the detected damage information and generates contextual insights related to damage severity and potential repair actions. These insights assist service centers in estimating repair requirements more effectively.

Finally, the results are stored in a MySQL database and displayed to the user through the web interface. The system architecture enables efficient processing and can be integrated with existing insurance claim management systems.

The proposed system also improves transparency in the estimation process by providing quick and automated analysis of vehicle damage. Furthermore, the use of deep learning models reduces human intervention and increases accuracy in identifying damaged areas.

IV. SYSTEM ARCHITECTURE

The Smart-Claim system architecture consists of several integrated modules that work together to perform automated damage detection and analysis. The major components of the system include the user interface, backend server, detection model, reasoning module, and database.

The user interface is developed using React, allowing users to upload images and view damage analysis results. The backend server is implemented using Python Flask and serves as an API layer connecting the frontend with the AI models and database.

The YOLO detection module performs object detection on uploaded vehicle images. The model identifies damaged areas and generates bounding boxes around affected regions. The Nemotron API powered by the Llama model then analyzes the detection output and provides reasoning-based insights.

The database module uses MySQL to store uploaded images, detection results, and generated analysis reports. This architecture ensures scalability and supports integration with enterprise-level insurance platforms.

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715302

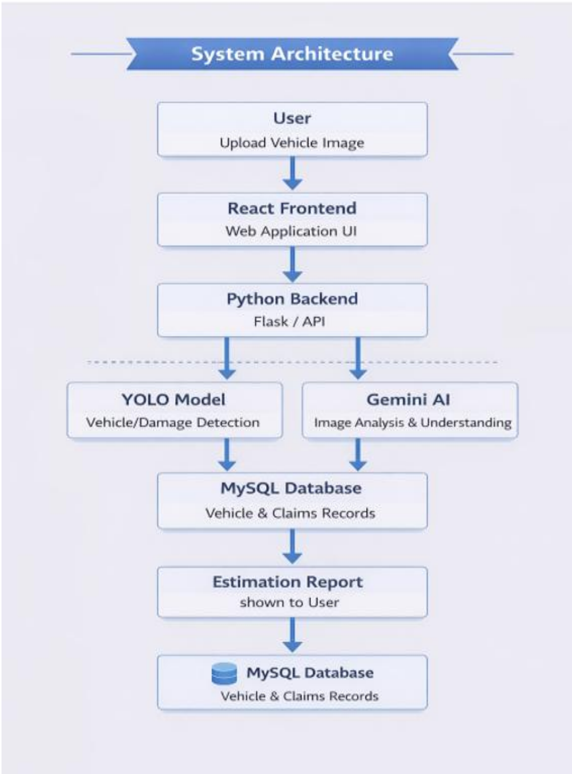

Figure 4.1 System architecture

Figure 4.2 Workflow Diagram

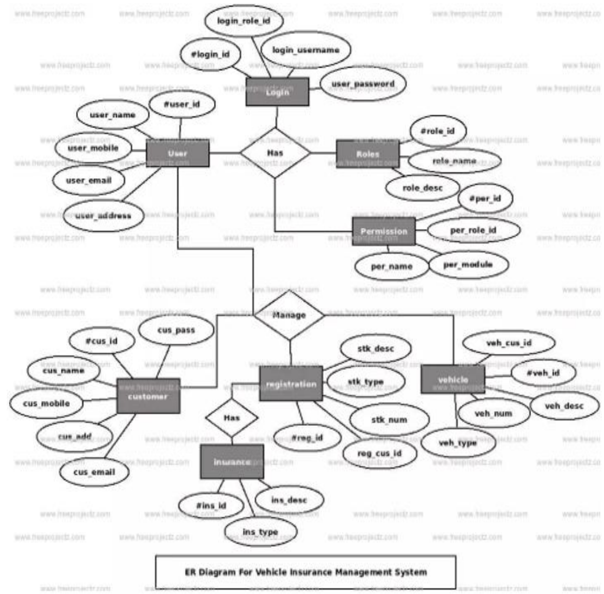

Figure 4.3 Database Diagram

The diagrams represent the overall design and operation of the Smart-Claim system. The Figure 4.1 system architecture shows how the user uploads a vehicle image through a React-based web interface, which is then processed by the Python Flask backend and analyzed using the YOLO model for vehicle damage detection. The detected damage information is further analyzed by the Nemotron/Llama AI model to generate repair estimation insights, and the results are stored in a MySQL database. The Figure 4.2 workflow diagram illustrates the step-by-step process including image upload, preprocessing, damage detection, AI analysis, and report generation. The Figure 4.3 database diagram represents how different entities such as users, vehicles, insurance records, and system permissions are organized and related to store and manage system data efficiently.

V. IMPLEMENTATION AND RESULTS

The Smart-Claim system was implemented using modern web technologies and deep learning frameworks. The YOLO model was trained using publicly available vehicle damage datasets containing images of damaged vehicles with annotated bounding boxes.

Experimental testing demonstrated that the system could accurately detect common damage types such as dents and scratches in vehicle images. The integration with the Nemotron API enabled contextual reasoning about the detected damage, providing estimation insights that assist repair planning.

Compared with traditional manual inspection processes, the Smart-Claim system significantly reduces the time required for damage evaluation. Automated analysis allows service centers to quickly understand the severity of vehicle damage and prepare repair estimates. These results demonstrate the effectiveness of combining computer vision and large language models for real-world applications.

VI. FUTURE WORK

Future improvements to the Smart-Claim system may include the development of mobile applications that allow users to capture vehicle images directly from smartphones. Additional enhancements may involve integrating the system with insurance company databases to automate claim approval procedures.

Another potential improvement is the implementation of real-time video-based damage detection. Analyzing video streams could provide more comprehensive information about vehicle damage and enable more accurate estimation analysis.

VII. CONCLUSION

This research presented Smart-Claim, an AI-based vehicle damage detection and estimation system integrating computer vision and large language models. The system uses YOLO for detecting damaged regions and NVIDIA Nemotron-3 API powered by the Llama 3.1 90B model for reasoning-based analysis. The proposed solution improves efficiency in insurance claim processing by reducing manual inspection time and providing automated damage assessment insights.

VIII. ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to the Department of Computer Applications at SRM Valliammai Engineering College for providing the opportunity and resources to carry out this research work. We would also like to thank our project guide and faculty members for their valuable guidance, suggestions, and continuous support throughout the development of the Smart-Claim system.

Special thanks are extended to our colleagues and friends who provided constructive feedback and encouragement during the implementation and documentation phases of the project. Their support played an important role in the successful completion of this work.

REFERENCES

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

[2] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv preprint arXiv:1804.02767, 2018.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017.

[4] W. Liu et al., "SSD: Single Shot MultiBox Detector," European Conference on Computer Vision (ECCV), 2016.

[5] Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Advances in Neural Information Processing Systems, 2012.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[7] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," International Conference on Learning Representations (ICLR), 2015.

[8] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.

[9] Goodfellow, Y. Bengio, and A. Courville, “Deep Learning,” MIT Press, 2016.

[10] Vaswani et al., “Attention Is All You Need,” Advances in Neural Information Processing Systems, 2017.

[11] NVIDIA Corporation, “Nemotron Large Language Model Documentation,” NVIDIA AI Research, 2024.

[12] Google AI Research, “Gemini: A Family of Highly Capable Multimodal Models,” Google Research Publications, 2024.

[13] R. Szeliski, “Computer Vision: Algorithms and Applications,” Springer, 2010.

[14] M. Abadi et al., “TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems,” Google Research, 2015.

[15] Rosebrock, “Deep Learning for Computer Vision with Python,” PyImageSearch Publications, 2019.

[16] Z. Chen, T. Ellis, and S. A. Velastin, “Vehicle Damage Detection Using Deep Learning Techniques,” IEEE International Conference on Image Processing (ICIP), 2019.

[17] T.-Y. Lin et al., “Microsoft COCO: Common Objects in Context,” Proceedings of the European Conference on Computer Vision (ECCV), 2014.

[18] Long, E. Shelhamer, and T. Darrell, “Fully Convolutional Networks for Semantic Segmentation,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[19] Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” International Conference on Learning Representations (ICLR), 2015.

[20] Dosovitskiy et al., “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” International Conference on Learning Representations (ICLR), 2021.