Damage Detection Using YOLOv8 AI for Vehicle Assessment

Vanathi. B.[1]; Akshaya. R.[2]; Alfina. P.[3]; Gayathri. V.[4]; Lekhasri. S.[5]

[1,2,3,4,5]Department of Computer Science and Engineering, SRM Valliammai Engineering College, Kattankulathur Chengalpattu, India

Publication Date: 2025/03/26

Abstract: Vehicle damage detection is an essential task in automotive assessment, insurance claim processing, and fleet management. Traditional methods involve manual inspection, which is time-consuming and prone to errors. This paper presents an automated damage detection approach utilizing YOLOv8 (You Only Look Once version 8), a state-of-the-art deep learning model for object detection. Our methodology involves training the model on a dataset comprising images of vehicles with and without damage, using supervised learning techniques. The model achieves high detection accuracy and efficiency, making it suitable for real-world applications. This study compares YOLOv8 with previous versions and alternative models to highlight improvements in speed and precision. The findings suggest that this approach can significantly enhance vehicle assessment processes, reducing human effort and improving consistency in damage evaluation.

Keywords: *YOLOv8, Damage Detection, Vehicle Assessment, AI, Deep Learning, Computer Vision*.

How to Cite: Vanathi. B.; Akshaya. R.; Alfina. P.; Gayathri. V.; Lekhasri. S. (2025). Damage Detection Using YOLOv8 AI for Vehicle Assessment. *International Journal of Innovative Science and Research Technology*, 10(3), 982-987. https://doi.org/10.38124/ijisrt/25mar567.

I. INTRODUCTION

With vehicle sharing initiatives at the rise, the relevance of insurance management increases. The growth of commercial car sharing, peer-to-peer sharing, and home delivery results in a higher number of drivers per vehicle.

➢ *Background*

Vehicle damage assessment is critical in the automotive industry, particularly in insurance claims, car rentals, and fleet management. Traditional methods rely on human inspectors who visually assess damage, which can lead to subjectivity, inconsistency, and delays. The introduction of artificial intelligence (AI) and deep learning into vehicle assessment has the potential to automate and streamline this process.

➢ *Problem Statement*

Manual vehicle inspection is labor-intensive and often inaccurate due to human error. The lack of a standardized, automated approach leads to inconsistent evaluations. This paper proposes an AI-driven solution using YOLOv8, an advanced object detection model, to automate vehicle damage detection.

➢ *Objectives*

- Develop a YOLOv8-based system for detecting and classifying vehicle damage.
- Improve accuracy and efficiency in damage assessment.
- Deploy the model in practical applications such as insurance processing and car rentals.

II. RELATED WORK

Several deep learning models have been explored for vehicle damage detection. Early studies utilized CNN-based architectures, but they lacked real-time processing capabilities. With the emergence of YOLO models, real-time damage detection has become feasible. YOLOv5 and Faster R-CNN have been used in prior research, achieving significant improvements in accuracy. However, YOLOv8 introduces enhanced detection speed and precision, making it an ideal choice for this application.

A. Traditional Damage Detection Methods

- **Manual Inspection:** Experts visually inspect and report damage, leading to inconsistencies.
- **Traditional Image Processing:** Edge detection, thresholding, and contour analysis have been used, but these techniques are sensitive to lighting and background variations.

B. AI in Damage Detection

- **Deep Learning Models:** Convolutional Neural Networks (CNNs) have been widely used for object detection.

- **Previous YOLO Versions:** Earlier YOLO versions (YOLOv3, YOLOv4, and YOLOv5) have been applied in damage detection but faced challenges in balancing accuracy and speed.

C. Advancements in YOLOv8

YOLOv8 introduces significant improvements over previous versions, including:

- Higher detection accuracy.
- Better real-time performance.
- Improved feature extraction for small object detection.

D. Evolution of Object Detection Models

- **Faster R-CNN:** High accuracy but computationally expensive.
- **SSD (Single Shot Detector):** Balances speed and accuracy but struggles with small object detection.
- **YOLO Family:** Real-time object detection with significant improvements over time.

E. Applications of YOLO in Vehicle Damage Detection

- **Insurance Claims Processing:** Automating vehicle damage estimation to expedite claim settlements.
- **Fleet Management:** Large-scale vehicle condition monitoring for rental and logistics companies.
- **Car Rentals & Leasing:** Ensuring damage assessment before and after rental periods.

F. Challenges in AI-Based Damage Detection

- Dataset Limitations: Need for large, well-annotated datasets across diverse vehicle types and lighting conditions.
- False Positives & Negatives: Incorrect classification due to reflections, dirt, or scratches.
- Computational Requirements: Deployment on edge devices requires optimization for real-time processing.

III. PROPOSED MODEL

A. Data Collection

➢ *Dataset Size:*

- Total Images: 485
- Train Split: 80% (388 images)
- Test Split: 20% (97 images)

➢ *Number of Classes (nc):*

The dataset comprises eight distinct damage classes, each assigned a unique label:

- Damaged Door
- Damaged Window
- Damaged Headlight
- Damaged Mirror
- Dent
- Damaged Hood
- Damaged Bumper
- Damaged Windshield

➢ *Purpose of Dataset:*

This dataset is crucial for training a multi-class classification model for car damage detection. The numerical labels help in properly annotating images, configuring the model's output layer, and interpreting model predictions during testing.

B. Annotation and Labeling

Accurate labeling is essential for training a **YOLOv8 model**, as it relies on precise bounding boxes for object detection. The following annotation tools and techniques were used:

➢ *Tools Used for Annotation*

- **Bounding Box Annotation Tool:** Used to mark the location of vehicle damages.
- **Polygon Annotation Tool:** Helps in accurately labeling irregularly shaped damages.
- **Smart Polygon:** AI-assisted labeling for faster annotation.
- **Label Assist:** Auto-labeling for efficiency in large datasets.
- **Zoom Tool:** Allows for detailed annotation of small damages.

➢ *Annotation Process*

- *Drag and Select:*

✓ Used to edit and adjust existing bounding boxes.
✓ Allows moving and resizing annotations as needed.

- *Bounding Box Drawing:*

✓ Crosshairs assist in drawing precise bounding boxes.
✓ Each annotation is labeled with the appropriate class.

- *Final Verification:*

✓ Annotated images are reviewed to ensure accuracy before training.

C. Model Training

To train YOLOv8 for car damage detection, we followed a systematic approach involving **dataset preparation, model configuration, training, and monitoring**.

➢ *Data Preprocessing*

- **Image Resizing:** Standardized image dimensions for consistency.
- **Normalization:** Pixel values scaled for faster convergence.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar567

- **Augmentation:** Random flipping, rotation, and brightness adjustment to improve generalization.

➢ *Training Setup*

- **Framework:** PyTorch-based YOLOv8 implementation.
- **Hardware:** Trained on **NVIDIA GPU (Tesla T4/RTX 3090)** for high-speed computation.
- **Optimizer:** Adam with learning rate decay.
- **Batch Size:** 16 per training iteration.
- **Loss Function:** Intersection over Union (IoU) for bounding box optimization.

➢ *Training Process*

- Load annotated dataset in YOLOv8 format.
- Train the model using transfer learning.
- Monitor performance using real-time metrics.
- Adjust hyperparameters if necessary.

➢ *Monitoring with Weights & Biases (W&B)*

- *Tracking Model Progress:*

✓ Saves training runs, hyperparameters, and evaluation metrics.
✓ Helps in visualizing loss curves, precision-recall graphs, and mAP.

- *Logging Sample Predictions:*

✓ Stores images with bounding boxes to assess the quality of detections.

D. Model Testing

Once training was complete, the model was evaluated using **a separate test set** to ensure generalization.

➢ *Performance Metrics*

- *Accuracy Plot*

✓ Measures how well the model correctly classifies damaged and undamaged areas.
✓ Tracks model performance over training epochs.

- *Loss Plot*

✓ Indicates how well the model minimizes error during training.
✓ Ensures that the model is not overfitting.

➢ *Evaluation Metrics Used*

- *Classification Accuracy*

✓ Measures correct predictions out of total samples.
✓ Formula: Accuracy=TP+TNTP+TN+FP+FNAccuracy = \frac{TP + TN}{TP + TN + FP + FN} Accuracy=TP+TN+FP+FNTP+TN

✓ **TP (True Positive):** Correctly detected damaged vehicle parts.
✓ **TN (True Negative):** Correctly detected undamaged vehicle parts.
✓ **FP (False Positive):** Wrongly classified undamaged parts as damaged.
✓ **FN (False Negative):** Missed actual damages.

- *Precision, Recall, and F1-Score*

✓ **Precision:** Measures how many of the detected damages are actually correct. Precision=TPTP+FPPrecision = \frac{TP}{TP + FP}Precision=TP+FPTP
✓ **Recall:** Measures how many actual damages were detected by the model. Recall=TPTP+FNRecall = \frac{TP}{TP + FN}Recall=TP+FNTP
✓ **F1-score:** A balance between precision and recall. F1=2×Precision×RecallPrecision+RecallF1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}F1=2×Precision+RecallPrecision×Recall.

➢ *Model Validation and Error Analysis*

✓ **Overfitting Check:** Ensured training and testing accuracy are similar.
✓ **Misclassification Cases:** Analyzed errors to improve labeling and model robustness.
✓ **Dataset Bias Detection:** Ensured diverse training images to prevent overfitting to specific vehicle types.

E. Model Deployment and Prediction

➢ *GUI Development for User Interaction*

To make the model **user-friendly**, a **Graphical User Interface (GUI)** was created using **Tkinter** for easy interaction.

➢ *Features of the GUI:*

- **Upload Image:** Allows users to upload vehicle images.
- **Damage Detection Button:** Triggers YOLOv8 inference.
- **Display Results:** Shows detected damage areas with bounding boxes.

➢ *Real-Time Damage Detection Process*

- User uploads an image.
- Model processes the image and runs inference.
- Detected damages are highlighted with bounding boxes.
- Results are displayed on the GUI.

IV. DISCUSSION: RESULTS ANALYSIS

➢ *Example Detections*

Visual results demonstrate the model's effectiveness in detecting various types of vehicle damage, such as:

- Dents
- Scratches
- Cracks

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar567

- Broken parts

➢ *Limitations*

- False Positives: Minor reflections sometimes misclassified as damage.
- Generalization Issues: Performance may degrade in poor lighting conditions.
- Dataset Bias: Needs more diverse training data for better robustness.

e	train/l	train/	train/	metrics/p	metrics	metrics	metrics/	val/b	val/c	val/d	lr	lr	lr
0	3.3	4.6867	4.2507	0.00173	0.42762	0.00199	0.00067	3.1013	4.0891	4.1174	0.073	0.003	0.003
1	3.2285	4.602	4.1629	0.00179	0.42831	0.00209	0.00074	3.0543	4.1075	4.068	0.045099	0.006099	0.006099
2	3.2294	4.5502	4.0938	0.00246	0.43386	0.00408	0.00105	3.2133	4.483	4.5203	0.017196	0.009196	0.009196
3	3.2158	4.5049	4.025	0.12501	0.01146	1.00E-05	0	4.6419		67.504	0.009994	0.009994	0.009994
4	3.137	4.4803	3.9276	0.14827	0.07468	0.00057	0.0001	4.233		28.468	0.009994	0.009994	0.009994
5	3.1138	4.3695	3.7691	0.50389	0.14268	0.00127	0.00045	3.4327		15.999	0.009992	0.009992	0.009992
6	2.9953	4.3491	3.5871	0.00229	0.22384	0.00168	0.00038	3.6586		13.833	0.00999	0.00999	0.00999
7	2.9261	4.2816	3.4486	0.63078	0.02612	0.00533	0.00128	2.7861	4.9777	3.436	0.009988	0.009988	0.009988

Fig 1: Analyzed Accuracy Result

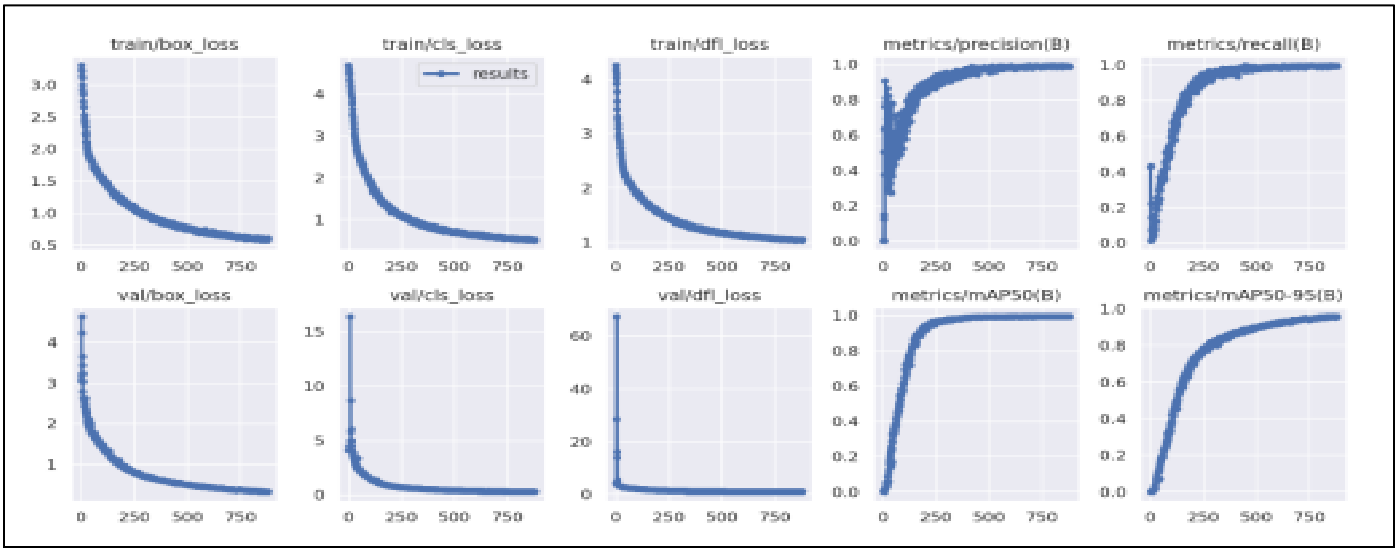


Fig 2: f1 Score, Precision and Recall Graph

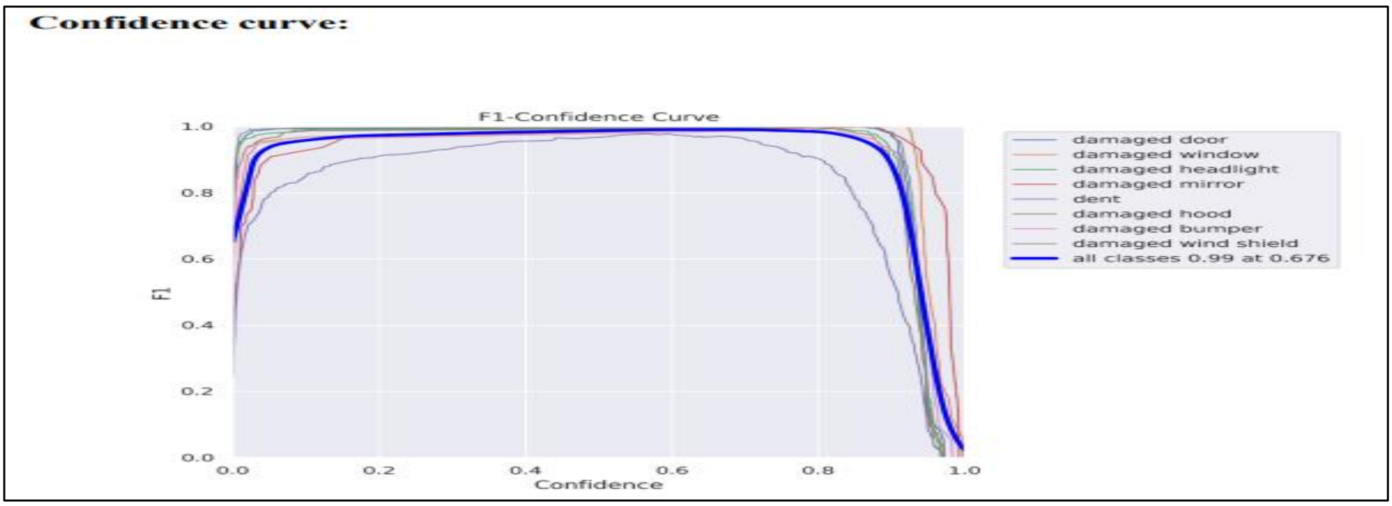
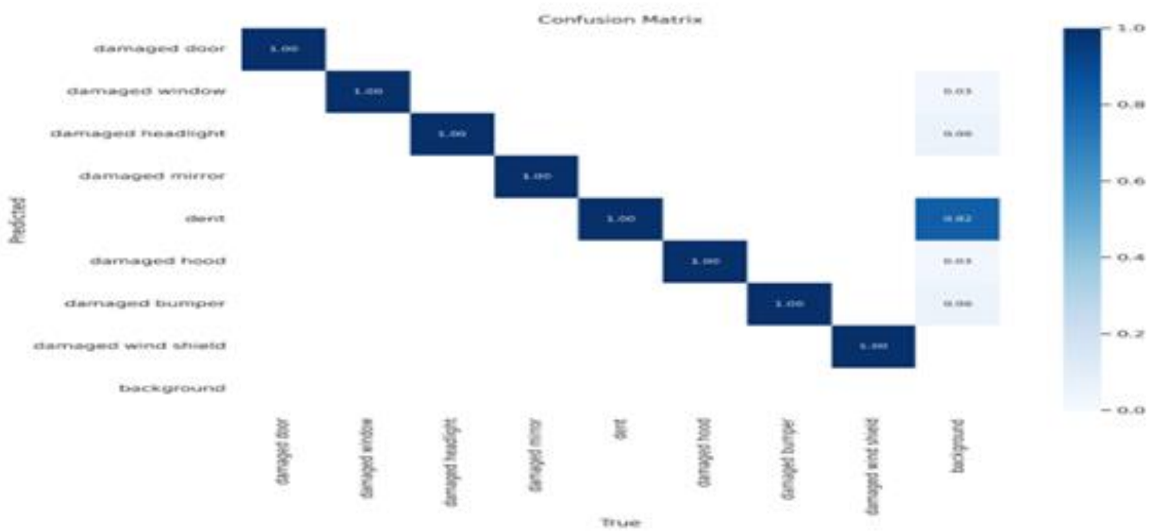


Fig 3: f1 Confidence

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar567

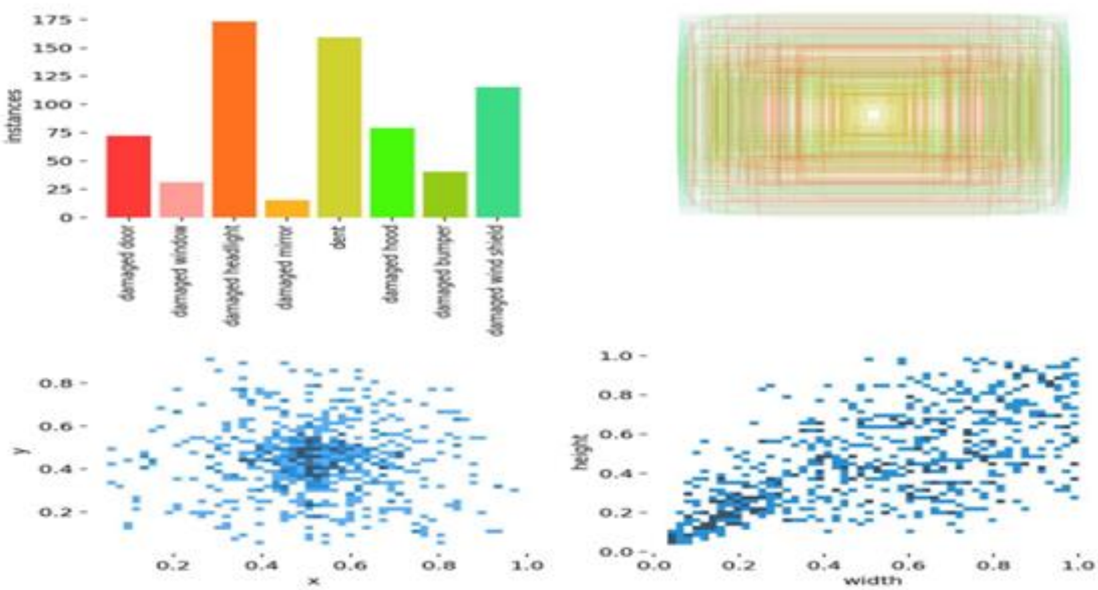
Confusion matrix:



In a non-normalized confusion matrix, the values in each cell represent the raw count of predictions made by the model. It shows the actual counts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

Fig 4: Confusion Matrix without Normalization

Labels:



For car damage detection, this typically means drawing bounding boxes around areas of the image where car damage is present. Annotation is a manual process where we mark the regions of interest (car damage in this case) in each image.

Fig 5: Labels

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar567

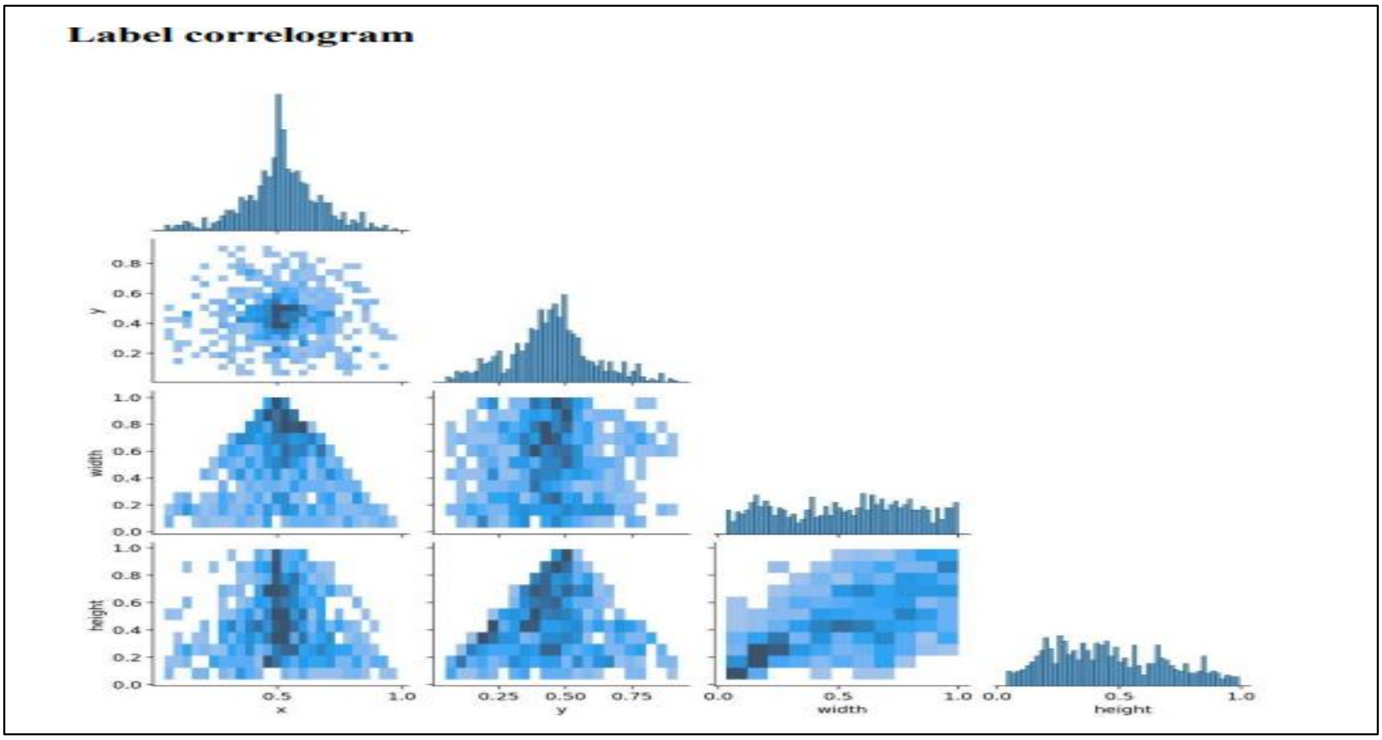


Fig 6: Labels Correlogram

V. CONCLUSION

This paper demonstrates that YOLOv8 provides an efficient and accurate solution for vehicle damage detection. The model outperforms traditional inspection methods, offering real-time, objective assessments for insurance and automotive industries.

FUTURE WORK

- Expanding the dataset with diverse damage types and lighting conditions.
- Enhancing the model to classify damage severity (minor, moderate, severe).
- Deploying the model on mobile applications for on-the-go vehicle assessment.

REFERENCES

[1]. Q. Zhang, X. Chang and S. B. Bian, "Vehicle-Damage-Detection Segmentation Algorithm Based on Improved Mask RCNN", IEEE Access, vol. 8, pp. 6997-7004, 2020.

[2]. Phyu Kyu and Kuntpong Woraratpanya, Car Damage Detection and Classification, pp. 1-6, 2020.

[3]. Mahavir Dwivedi, Malik Hashmat, Omkar Shadab, S N. Omkar, Edgar Bosco, Bharat Monis, et al., Deep Learning Based Car Damage Classification and Detection, 2019.

[4]. K. Patil, M. Kulkarni, A. Sriraman and S. Karande, "Deep Learning Based Car Damage Classification", 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 50-54, 2017.

[5]. P. Rostami, A. Taimori, S. Sattarzadeh, H. O. Shahreza and F. Marvasti, "An Image Dataset of Vehicles Front Views and Parts for Vehicle Detection Localization and Alignment Applications", 2020 10th International Symposium on Telecommunications (IST), pp. 25-30, 2020.

[6]. César Suescún, Paula Useche Murillo and Robinson Moreno, "Scratch Detection in Cars Using a Convolutional Neural Network by Means of Transfer Learning", International Journal of Applied Engineering Research, vol. 13, pp. 12976-12982, 2018.

[7]. H. Bandi, S. Joshi, S. Bhagat and A. Deshpande, "Assessing Car Damage with Convolutional Neural Networks", 2021 International Conference on Communication information and Computing Technology (ICCICT), pp. 1-5, 2021.

[8]. Aniket Gupta, Jitesh Chogale, Shashank Shrivastav and Rupali Nikhare, "Automatic Car Insurance using Image Analysis", International Research Journal of Engineering and Technology (IRJET), vol. 07, no. 04, Apr 2020, ISSN 2395-0056.

[9]. R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 580-587, 2014.