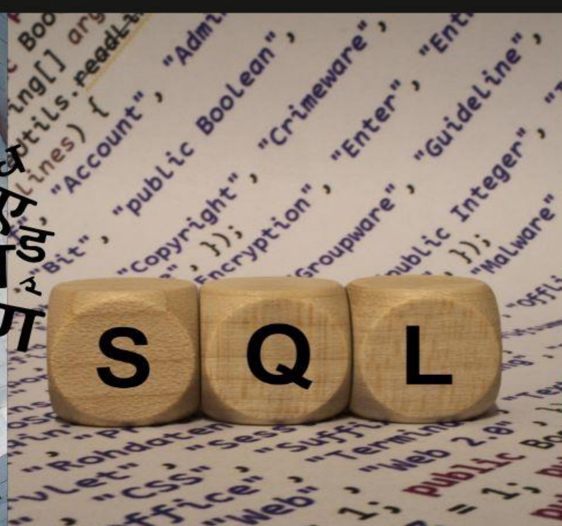
THESIS

Prepared by:

AFRIN QURESHI

2024-2025

RESEARCH GUIDE:
PROF. DR. ASHISH GAVANDE



PROJECT

THESIS

Prepared by:
AFRIN QURESHI

Research Guide:
Prof. Dr. Ashish Gavande

M.Sc DataScience & AI

Exam Seat No __________

Shri Vile Parle Kelavani Mandal's
Mithibai College of Arts, Chauhan Institute of Science &
Amrutben Jivanlal College of Commerce and Economics
Vile Parle (West), Mumbai – 400 056

C e r t i f i c a t e

Date _______________

This is to certify that the project entered in this documentation as have been signed were performed by Mr **Arpit Chauhan** of project named **"**Automated Vehicle Damage Analysis and Cost Prediction with Mask R-CNN for Insurance and Repair Optimization**"** in M.Sc. Data Science and Artificial intelligence Part II Computer Science Laboratory of this college during the year 2024-25.He has completed the project in the subject of Computer Science as prescribed in the syllabus of the University of Mumbai.

Project Guide **Head, Department**

Date:

ACKNOWLEDGEMENT

I would like to express my sincere gratitude towards the **Computer Science Department** of **“Mithibai College of Arts, Chauhan Institute of Science & Amrutben Jivanlal College of Commerce & Economics”**

I would like to thank my Project Guide, **Prof. Dr Ashish Gavande Sir** for his constant support during the project

I am also thankful to respected other staff members for their willing co-operation. We can’t forget Suggestion and encouragement of our friends and all other who helped me directly or indirectly to make this project a success.

Thanking You

Arpit Chauhan

DECLARATION

This is to certify that this project **“**Automated Vehicle Damage Analysis and Cost Prediction with Mask R-CNN for Insurance and Repair Optimization**”** is original and not a copyright of any other project performed in previous years of this institution.
I, **Arpit Chauhan**, a student of **M.Sc. Data Science and Artificial Intelligence (Part II)** at this institute, have undertaken this project to apply my knowledge in the fields of **Computer Vision, Deep Learning, and Applied Machine Learning**. This project has been developed using **Python**, employing the **Detectron2 framework and Mask R-CNN architecture,** for instance segmentation of vehicle damage. Furthermore, **IoU-based techniques** were integrated with **Linear Regression models** to estimate the cost of repair accurately
The objective of this project is to build an intelligent, automated system for detecting vehicle damage and predicting repair costs from 2D images, streamlining the insurance claim process, improving user experience, and minimizing manual intervention.

ABSTRACT

One part of the automotive industry that could be poised for a major disruption is the automatic damage detection on vehicles, and this is most likely to occur through image data for insurance companies, car

rental companies, and repair shops. In this work, we use one of the most powerful neural network architectures in its class: Mask R-CNN.

We trained this model on a COCO-Formatted dataset containing images,where each image contains annotations of damaged areas. Using a standard 2D image capture, the system detects and assesses damage to a vehicle and helps drivers speed up the insurance claims process without the need for a physical inspection. This study uses cutting-edge computer vision techniques to identify damage and categorize its severity. To determine the degree of damage, the algorithm calculates the Intersection over Union (IoU) score between the ground truth and anticipated bounding boxes. Based on the extent of the damage found, these scores are subsequently utilized in a linear regression model to calculate repair costs.

This study's main goal is to transform the auto insurance claim procedure by making it quicker, more economical, and more successful. It also enhances consumer satisfaction, aids in the detection of fraud, and shows how AI-powered damage detection might change the direction of the vehicle insurance sector.

INDEX

Chap 01: Model Proposal

This study proposes a hybrid framework that integrates both **deep learning-based instance segmentation** and **rule-driven cost estimation** to automate vehicle damage assessment and repair cost prediction. Our approach introduces a two-stage pipeline:

1. A **Mask R-CNN model**, implemented using the Detectron2 framework, is trained to identify and segment damaged regions from standard 2D vehicle images with pixel-level accuracy. The model is fine-tuned on a COCO-formatted car damage dataset containing diverse damage types such as dents, cracks, scratches, and shattered components.
2. A **post-processing module** that utilizes **Intersection over Union (IoU)** metrics and **custom rule-based logic** to quantify the severity of the damage. Based on IoU scores and damage categories, repair costs are estimated using a **linear regression model** trained on domain-informed mappings.

The novelty of this architecture lies in the clear separation between **damage localization** and **cost estimation**, allowing for **modular enhancements**, **domain customization**, and **fine-grained control** over individual processing stages. This design not only ensures high accuracy in detection but also enables explainable and tuneable repair cost prediction aligned with real-world scenarios.

Chap 02: AIM & OBJECTIVE

The aim of this study is to develop an intelligent and automated system for vehicle damage detection and repair cost estimation using advanced deep learning techniques. By utilizing **Mask R-CNN**, a state-of-the-art instance segmentation model implemented via **Detectron2**, the system accurately identifies and segments various types of car damage from 2D images.

To ensure a reliable and realistic cost estimation process, this study further incorporates **IoU-based damage severity analysis** and a **linear regression model** trained on predicted bounding box features. This hybrid approach enhances the interpretability and precision of damage evaluation.

The primary objective is to streamline and automate the **automobile insurance claim process**, making it faster, more accurate, and accessible. The system eliminates the need for manual inspection in many cases, improves customer satisfaction, reduces operational overhead for insurance companies, and contributes to fraud prevention. Additionally, this research aims to set a benchmark for AI-driven insurance automation, demonstrating how computer vision and machine learning can be effectively combined for practical real-world applications.

Chap 03: INTRODUCTION

3.1 Overview

In recent years, artificial intelligence (AI) and deep learning have emerged as transformative technologies across various industries. One such domain is the **automotive insurance sector**, which traditionally relies on time-consuming, manual inspections to evaluate vehicle damage and process insurance claims. These conventional methods are often inefficient, prone to human error, and susceptible to fraudulent activities. The integration of AI-driven image processing and machine learning offers a compelling solution to automate and optimize this process.

In this regard, the system relies on the most advanced technology in image segmentation and machine learning. Through CNNs, the system will be able to examine vehicle photographs, identify certain components, and precisely identify areas of damage. Automation of the process accelerates the assessment of damage, often obviating the need for an in-person inspection for minor to moderate damage. The end result is a faster and more painless claims experience for drivers and insurers alike. Our system will use machine learning to analyze the past claims made and estimate repair costs based on the damage incurred.

This automation hastens the settlements of claims in a more accurate and satisfying way for customers. Drivers get to enjoy faster repairs, simpler claims, and quicker payout, which eliminates stress. Insurance companies save money, detect fraud better, and improve customer experience. Even repair shops benefit through streamlined workflows and faster service. This study demonstrates how technology can revolutionize car insurance claims and make it faster, easier and less stressful for both parties. This is a step toward a smarter, more efficient future for insurance.

3.2 Background and Motivation

The process of assessing vehicle damage for insurance purposes is typically manual and involves several stakeholders, including surveyors, mechanics, and insurance agents. This results in prolonged turnaround times for claim processing and higher operational costs for insurance companies. Moreover, the lack of standardization in evaluating damage severity often leads to inconsistent claim settlements.

With the rise of digital technologies and the widespread availability of smartphones, users can now capture high-resolution images of their vehicles immediately after an accident. This opens up an opportunity for **image-based automatic damage assessment** using **computer vision techniques**.

Mask R-CNN, developed by Facebook AI Research, is one of the most advanced architectures for **object detection and instance segmentation**. It allows for precise localization of objects along with pixel-level segmentation, making it ideal for identifying damaged vehicle parts such as **dents, scratches, broken lights, and cracked windows**. By training this model on a dataset of annotated vehicle images, the system learns to detect damage accurately even in varying lighting, background, and vehicle conditions.

Detecting damage is only part of the solution—translating it into repair costs is essential for insurance applications. This project uses IoU scores to measure damage severity and applies a rule-based regression model to estimate costs based on historical data, providing a complete system for automated insurance claim processing.

3.3 Problem Statement

The traditional automobile insurance claim process involves several manual steps, including:

- Visual inspection of the damaged vehicle,
- Damage documentation and photo collection,
- Consultation with repair shops for cost estimation,
- Verification and approval by insurance agents.

These steps are not only time-consuming but also inefficient and inconsistent. Furthermore, insurers face challenges such as **fraudulent claims**, **delays in claim processing**, and **high operational costs**. On the user side, claimants often face **delays and inconvenience**, reducing satisfaction and trust in the insurance provider.

Hence, the core problem addressed by this study is:

How can we automate and streamline the vehicle damage assessment and repair cost estimation process using deep learning techniques to enhance speed, accuracy, and transparency in insurance claims?

3.4 Objectives

This project seeks to build an end-to-end system for **automated car damage detection** and **repair cost prediction** using deep learning and machine learning. The key objectives include:

- Utilizing **Mask R-CNN** to perform accurate instance segmentation of damaged parts in vehicle images.
- Calculating **IoU (Intersection over Union)** to assess the extent of damage.
- Designing **rule-based logic** to assign repair costs based on damage type and severity.
- Training a **linear regression model** to predict repair costs using bounding box features.
- Providing a modular and explainable pipeline that separates detection and estimation for easier debugging and enhancement.
- Demonstrating the applicability of the system using real-world test images and generating actionable outputs.

3.5 Significance of the Study

This study is significant because it combines **state-of-the-art computer vision algorithms** with **real-world insurance applications** to build a practical and scalable solution. Some key benefits include:

- **Automation of Claim Processing**: Reduces the time and manpower required to evaluate insurance claims.
- **Accuracy and Consistency**: Uses data-driven models to ensure consistent and objective damage assessment.
- **Fraud Detection**: Helps in minimizing false or exaggerated claims through image-based evidence and quantifiable damage scoring.

- **User Convenience**: Enables users to upload images via mobile apps and receive instant assessments.
- **Scalability**: Can be deployed across insurance platforms to handle large volumes of claims automatically.

3.6 Scope of the Project

The scope of this project includes:

- Training and validating a **Mask R-CNN** model on a COCO-formatted **Car Damage Dataset**.
- Performing **instance segmentation** to identify and label different types of vehicle damage.
- Applying **IoU calculations** to assess detection quality.
- Creating a **repair cost mapping system** using both rules and machine learning.
- Evaluating the system on **test images** and documenting **accuracy, AP (Average Precision), and estimated cost** outputs.
- Excluding real-time video input, 3D damage modeling, or integration with repair shop databases (which can be considered in future work).

3.7 Technologies Used

The project employs a wide range of tools and technologies, including:

- **Python**: Core programming language.
- **Detectron2**: Facebook AI's object detection and segmentation library.
- **Mask R-CNN**: Deep learning model for instance segmentation.
- **COCO Dataset Format**: For organizing and training data.
- **OpenCV**: For image processing.

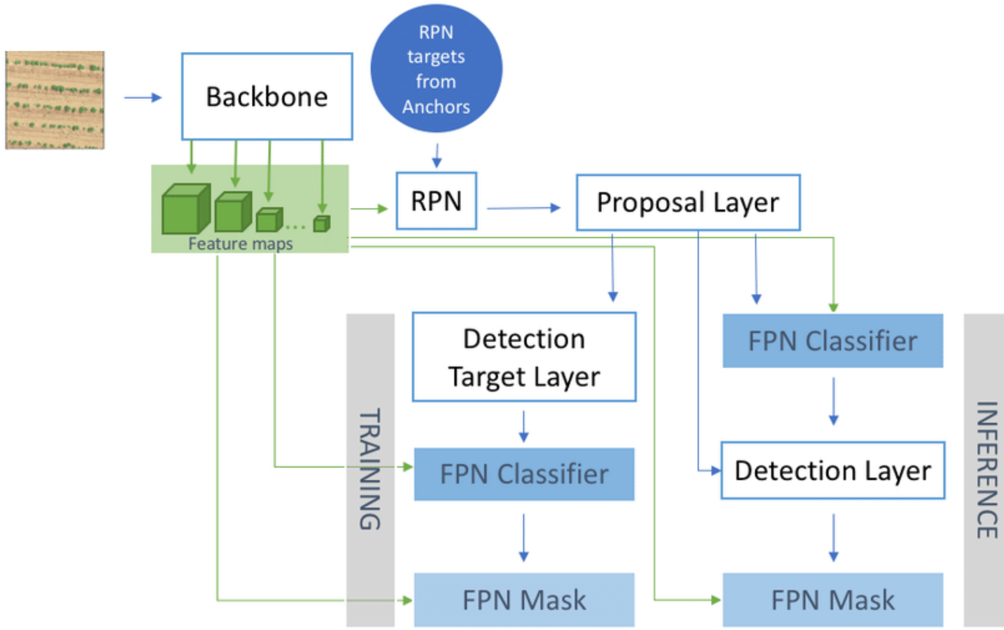
- **Scikit-learn**: For regression modeling and evaluation.
- **Augmentations**: For data augmentation.
- **TensorBoard**: For training visualization and monitoring.
- **Google Colab / Jupyter Notebook**: For development and experimentation.

Chap 04: Literature Review

4.1 Foundations in Instance Segmentation

The roots of instance segmentation can be traced back to R-CNN (Region-based Convolutional Neural Network), proposed by Girshick et al. (2014). This approach marked a revolutionary shift from traditional object detection by integrating region proposals with convolutional neural networks. R-CNN significantly enhanced detection accuracy, achieving a notable ~30% improvement in mean Average Precision (mAP) on the PASCAL VOC 2012 dataset.

Over time, the architecture evolved through Fast R-CNN and Faster R-CNN, which introduced Region Proposal Networks (RPNs) to streamline the detection pipeline. The final evolution in this family was **Mask R-CNN**, which added a parallel branch for pixel-level segmentation, making it ideal for detecting subtle damages in vehicle images.

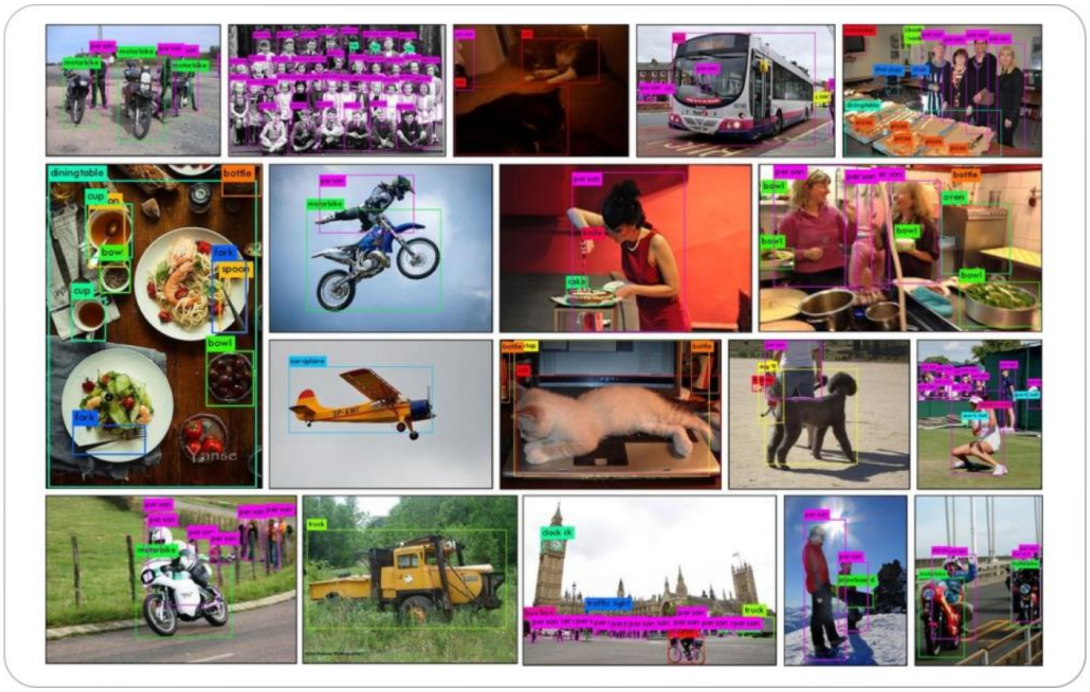


4.2 The COCO Dataset as a Benchmark

Before COCO, datasets such as ImageNet and PASCAL VOC were predominantly used for classification and object detection. However, they lacked complexity and per-instance annotations. The introduction of the Microsoft COCO dataset addressed this gap by providing:

- Over 328,000 images
- More than 2.5 million labeled instances
- Dense contextual relationships among objects

This made COCO the gold standard for evaluating computer vision models in segmentation and object detection. It pushed the development of highly capable architectures like Mask R-CNN, which thrive on detailed annotations for fine-tuned prediction accuracy.



4.3 Advancements with Detectron and Detectron2

Detectron2, developed by Facebook AI Research, builds on the legacy of its predecessor Detectron. It provides a high-performance modular framework based on PyTorch, with enhanced features such as:

- Native support for distributed training
- Built-in support for state-of-the-art models like Cascade R-CNN, DensePose
- Simple configuration system
- Compatibility with COCO-style datasets

These characteristics make it highly adaptable for custom tasks such as vehicle damage detection. The ease of customization allows for incorporation of dropout layers, augmentation, and cost estimation logic.

4.4 Image-Based Vehicle Detection Approaches

Traditional vehicle detection methods used classical techniques like:

- Haar Cascade classifiers
- Color thresholding
- Contour-based segmentation
- Heatmaps and bounding box merging

While useful in early-stage research, these methods faced limitations in varying lighting conditions, occlusion, and complex backgrounds. Modern models have surpassed them in accuracy and robustness, especially under real-world variability.

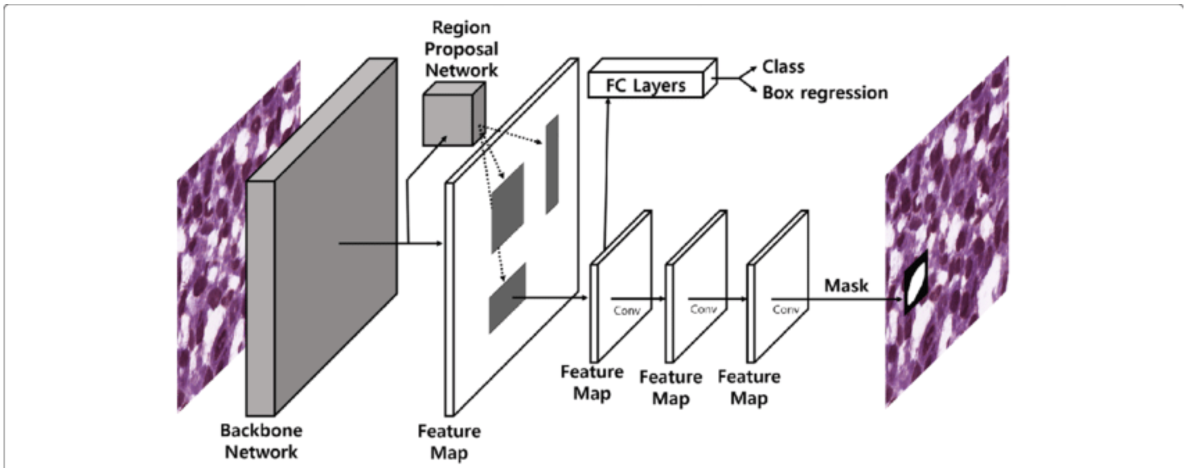
A notable approach includes the use of pixel-level regression models combined with handcrafted features to achieve over 85% accuracy under certain controlled environments. However, these lacked the scalability and generalization offered by deep learning architectures.

4.5 Improved Damage Detection via Mask R-CNN

Recent research has shown significant improvements by customizing Mask R-CNN for automotive applications. Key improvements include:

- Enhancing the backbone with deeper ResNet variants
- Integrating Feature Pyramid Networks (FPN)
- Modifying anchor box sizes and ratios for damage-specific detection
- Incorporating dropout layers to prevent overfitting
- Adjusting IoU thresholds during training and inference

These refinements help in better localization of small or partial damages, which is a common challenge in real-world insurance claim images.

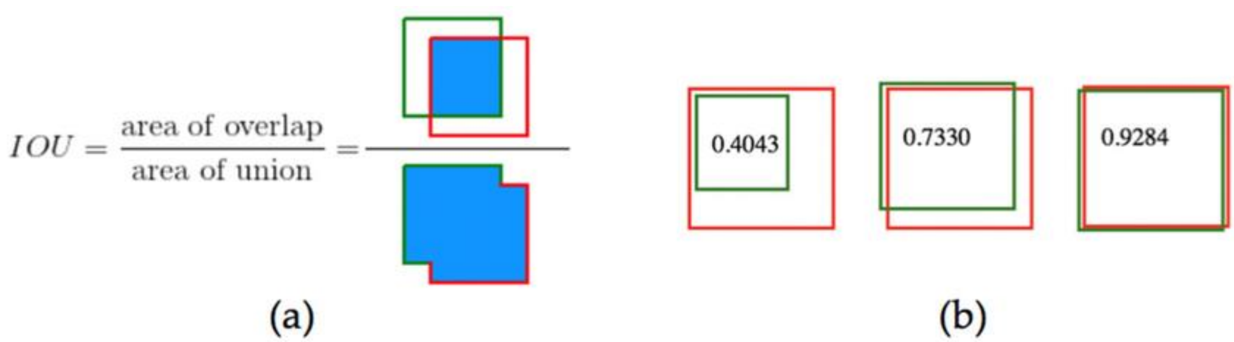


4.6 IoU (Intersection over Union) for Detection and Evaluation

IoU is a core metric in object detection and segmentation. It quantifies the degree of overlap between the predicted bounding box and the ground truth. The higher the IoU, the better the prediction quality. IoU is used:

- During training (for assigning positive/negative samples)
- During validation (for precision evaluation)
- During inference (for filtering low-confidence detections)

In this project, IoU also serves as a proxy for **damage severity**, which is later used to estimate repair costs through a regression model.



4.7 Repair Cost Estimation with Regression Models

Estimating repair costs accurately is critical in real-world insurance workflows. Traditional manual assessments are inconsistent and labor-intensive. In this system:

- Bounding box features (size, position, etc.) are extracted
- StandardScaler normalizes the feature space
- A **Linear Regression** model maps bounding box features to repair costs

Different cost functions are defined for parts like door, hood, bumper, lamp, etc., based on damage severity (measured via IoU). The regression model is trained using historical data and predicted damage attributes.

This model allows for automatic, fast, and reasonably accurate cost estimation with minimal human intervention.

From foundational R-CNN architectures to Detectron2 and cost estimation via regression, the literature shows an ongoing evolution of tools and methodologies to support automated damage analysis. These technologies, when properly integrated, offer a scalable, data-driven, and highly efficient alternative to traditional insurance claim processes.

Summary

From foundational R-CNN architectures to Detectron2 and cost estimation via regression, the literature shows an ongoing evolution of tools and methodologies to support automated damage analysis. These technologies, when properly integrated, offer a scalable, data-driven, and highly efficient alternative to traditional insurance claim processes.

This study builds upon these innovations by:

- Leveraging COCO-format datasets
- Using Detectron2 with Mask R-CNN for instance segmentation
- Employing IoU as a proxy for damage severity
- Estimating costs through a regression-based model

Chap 05: Materials and Methodology

5.1 Introduction

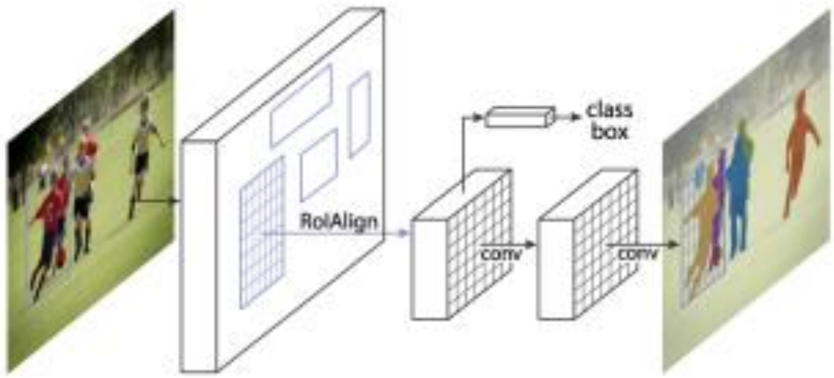
This chapter presents a comprehensive account of the materials used and the methodology implemented to achieve the objectives of the proposed system for automated vehicle damage detection and repair cost estimation. The approach involves leveraging state-of-the-art deep learning frameworks, most notably Mask R-CNN within the Detectron2 environment, to detect and segment various types of vehicle damages from standard 2D images. In addition to this segmentation task, a cost estimation module is developed using regression techniques, based on the extent and severity of the detected damage. The following sections provide a detailed explanation of the dataset utilized, the architectural components of the system, training configurations, evaluation metrics, and the regression-based cost estimation pipeline.

5.2 Dataset Used

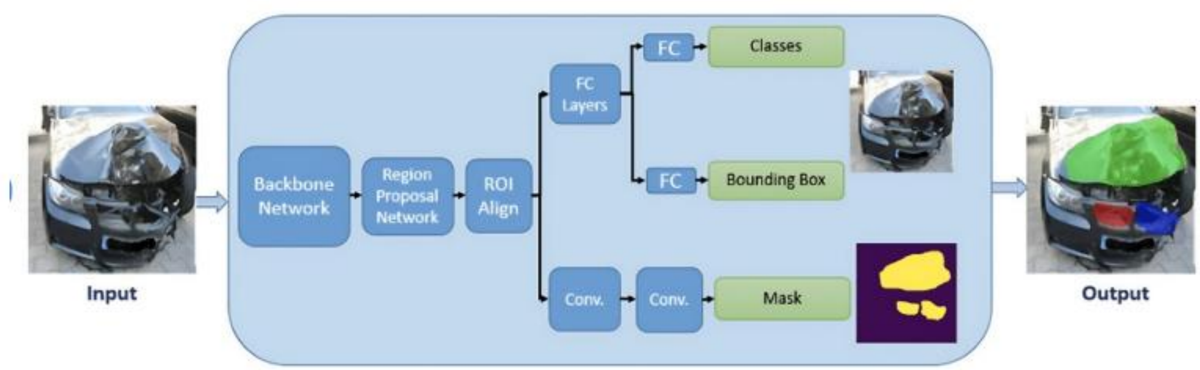
The dataset used for training and evaluating the proposed model is the Car Damage Detection (CarDD) dataset. This dataset was specifically selected due to its extensive collection of annotated images depicting various real-world vehicle damages. The dataset comprises thousands of high-resolution images of cars labeled with segmentation masks and bounding boxes representing different types of damages. It is organized into three distinct subsets: a training set consisting of 2,819 images, a validation set containing 813 images, and a test set with 377 images. Each image in the dataset has been carefully labeled and categorized into one of six distinct damage classes, namely dent, scratch, crack, glass shatter, lamp broken, and tire flat. The COCO-style format of the annotations allows for efficient integration with the Detectron2 training pipeline and provides both instance-level segmentation and categorical identification. This rich and well-labeled dataset forms the cornerstone for training a deep learning model capable of understanding complex damage patterns with a high degree of accuracy.

5.3 Model Architecture

The backbone of the proposed damage detection system is built using the Mask R-CNN architecture, which is an extension of the Faster R-CNN model and has been extensively validated for object detection and instance segmentation tasks. Mask R-CNN introduces an additional branch to predict segmentation masks for each detected object, thereby enhancing its applicability in scenarios requiring pixel-wise localization of features. The model is implemented using Detectron2, a high-performance library developed by Facebook AI Research. In this system, the ResNet-50 architecture is employed as the primary feature extractor. ResNet-50 is known for its depth and ability to learn hierarchical feature representations, making it suitable for capturing fine-grained details in damage patterns. A Feature Pyramid Network (FPN) is integrated with the backbone to facilitate multi-scale detection by generating feature maps at different resolutions. These feature maps are then passed through a Region Proposal Network (RPN), which identifies potential regions in the image that might contain damaged parts.



Once regions of interest (RoIs) are proposed by the RPN, the ROIAlign module aligns the features precisely to ensure that the spatial structure is preserved, leading to more accurate bounding box predictions. Finally, the segmentation head takes over and produces a binary mask for each instance, corresponding to the exact shape and area of the damage. The model is capable of producing both bounding box coordinates and segmentation masks, enabling precise localization of damages on vehicle surfaces.



5.4 Detailed Network Configuration

The architecture implemented in this study includes several advanced configurations aimed at enhancing detection performance and generalization ability. The model begins with a convolutional layer followed by batch normalization and a ReLU activation function. A max-pooling layer is then applied to reduce the spatial dimensions while preserving the most prominent features. The output is passed through multiple FPN layers which construct a hierarchical representation of the image. The RPN then generates bounding box proposals, which are refined through ROIAlign and subsequently forwarded to the mask head to produce the final segmentation output.

This entire structure is carefully tuned using a configuration file within Detectron2. Parameters such as the number of classes, learning rate, batch size, and number of training iterations are defined within this file to control the behavior of the model during training. The model is fine-tuned on the CarDD dataset, and custom thresholds are introduced in the ROI heads to refine the bounding box selection process. The addition of dropout layers at key positions in the network further prevents overfitting by randomly disabling certain neurons during training, thereby enhancing the model's robustness.

Layer (Type)	**Output Shape**	**Param #**
Input Layer	(None, 64, 122, 1)	0
Conv2D	(None, 64, 122, 32)	320
BatchNorm	(None, 64, 122, 32)	128
ReLU	(None, 64, 122, 32)	0
MaxPooling2D	(None, 32, 61, 32)	0
FPN Layer	Multi-scale Feature Maps	-
ROIAlign	Pooled feature maps	-
RPN	Bounding box proposals	-
Mask Head	Segmentation mask output	-
Total Params	**39,989**	**(156.21 KB)**

5.5 Data Augmentation and Preprocessing

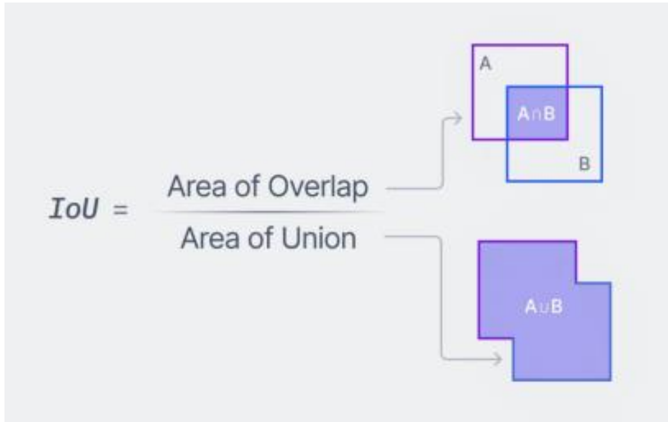
To ensure that the model generalizes well across diverse damage patterns, extensive data augmentation techniques are applied during the training phase. These augmentations include horizontal and vertical flipping, random rotations, brightness and contrast adjustments, and slight cropping. These transformations simulate real-world variations in the input images, such as changes in lighting, orientation, and camera angles. Additionally, the input pixel values are normalized to a fixed range to stabilize the learning process and improve convergence speed. Preprocessing also involves resizing the images to a uniform scale to maintain consistency in the input dimensions.

5.6 Intersection over Union (IoU) in Model Training and Evaluation

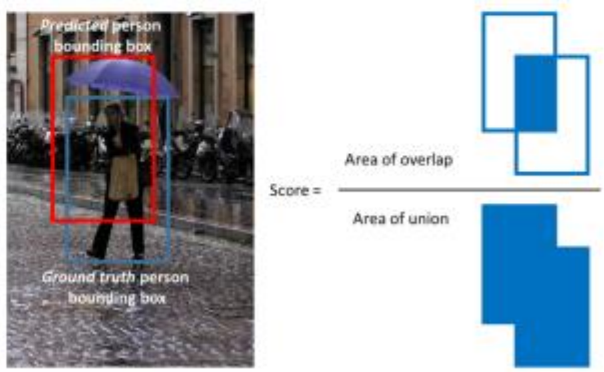
To improve the correctness of our proposed vehicle damage detection model, we apply the IoU as a critical performance metric during both the training and testing processes. IoU is used to measure the overlap between the predicted bounding boxes and the ground truth annotations and ismathematically defined as

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

IoU in Model Training: In the training process, the Detectron2 library uses the IoU value to improve the object detection results. The RoI heads employ IoU-based thresholding to generate valid object proposals. The model employs the following default IoU threshold values:



IoU in Evaluation: For model evaluation, we employ the COCO Evaluator , which is based on the IoU as a key performance metric. The function: assesses the precision of the predicted bounding boxes by calculating the IoU scores. The IoU scores are used to measure the spatial correspondence between the predicted damage regions and the ground truth annotations and a higher value indicates a better match.



IoU in Prediction Phase: When applying the model to new images, an IoU-based confidence threshold is used to filter out low-quality detections

```
cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.70
```

This guarantees that only damage regions with high IoU confidence scores are considered real detections. We improve the robustness of our vehicle damage detection system by including IoU at multiple stages in our pipeline, namely training, validation, and inference, and thereby enhance localization accuracy and reduce repair cost estimation.

5.7 IoU for Repair Cost Estimation

Intersection over Union (IoU) quantifies the overlap between predicted damage regions and ground truth. A higher IoU indicates more severe damage and impacts cost estimation. Estimated Repair Cost = α × IoU + β (2) where α and β are regression coefficients from a linear regression model

Repair Cost Assignment: Custom cost functions are defined for different vehicle parts:

$$C_{\text{repair}}(IoU) = \begin{cases} \text{High}, & IoU \geq 0.7 \\ \text{Moderate}, & 0.5 \leq IoU < 0.7 \\ \text{Low}, & 0.3 \leq IoU < 0.5 \\ \text{Minimal}, & \text{otherwise} \end{cases}$$

Machine Learning for Cost Prediction: A regression model predicts repair costs using bounding box features:

Y = β0 + β1X1 + … + βnXn + ε

where Y is the repair cost, Xi are bounding box features, and ε is error

Final Cost Calculation: Total repair cost is computed as:

$$\text{Total Repair Cost} = \sum_{i=1}^{n} C_{\text{repair},i}$$

where n is the number of detected damages

5.8 Creation of Repair Cost Dataset

The dataset for the cost of repair was generated using a multi-step process. Here is the car damage detection model initially run on a set of training images to predict bounding boxes. These predicted bounding boxes were integrated with ground truth annotations to form a full dataset of detected and annotated auto damage cases. The following phase was based on the computed IoU scores of the overlap regions between the ground truth and forecasted bounding boxes. IoU was a crucial indicator for assessing how accurate the model's forecast was. The allocation of the repair costs followed a methodical methodology. Data from several websites was also used along with previously extracted pose sequence and IoU scores to label the corresponding repair costs in the dataset. This methodology ensured that repair costs were allocated in proportion to the model's predictions and the ground truth, making precise the repair costs estimation.

5.9 Model Training Process

The training process is conducted using stochastic gradient descent (SGD) as the optimization algorithm, with a base learning rate of 0.0001 and a momentum of 0.9. The model is trained using a batch size of two images per iteration, and a total of 10,000 iterations are run to ensure adequate convergence. To prevent overfitting and reduce computational cost, the training process incorporates early stopping based on the validation loss and evaluation metrics such as mean average precision. Periodic evaluations are performed using the validation set, and the model's performance is monitored using TensorBoard. These evaluations guide further adjustments in the hyperparameters and provide insights into the model's learning trajectory.

5.10 Evaluation Metrics

To quantitatively assess the model's performance, two widely accepted metrics in object detection and segmentation tasks are used: Intersection over Union (IoU) and Mean Average Precision (mAP). The IoU score is calculated as the ratio of the overlapping area between the predicted bounding box and the ground truth annotation to the area of their union. A high IoU indicates better alignment and accuracy in damage localization. During training, IoU is used to label positive and negative examples in the RoI head. During validation and inference, an IoU threshold of 0.5 and 0.7 is applied to determine the precision of the predicted masks and bounding boxes.

The mAP score provides an aggregated measure of precision and recall across different IoU thresholds and damage categories. These metrics are computed using the built-in COCOEvaluator class in Detectron2, which evaluates the model on the validation set after every fixed number of iterations. Together, IoU and mAP ensure a rigorous and reliable evaluation of the model's capability to detect and segment vehicle damages.

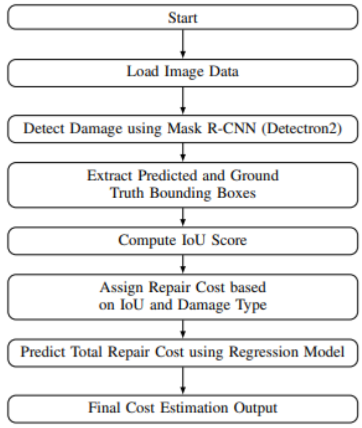
5.11 Repair Cost Estimation

Following the segmentation and bounding box predictions, the next stage involves estimating the repair cost associated with the detected damages. This is achieved using a linear regression model trained on features extracted from the predicted bounding boxes. Each bounding box is associated with various features, including its size, coordinates, and the IoU score with respect to the ground truth. These features are standardized using a scaler to ensure uniform distribution across different scales. A linear regression model is then trained using these features and a corresponding set of manually assigned repair costs for each damage instance.

The repair cost estimation is governed by a set of predefined rules that link IoU scores with cost ranges. For example, if the IoU of a predicted dent is greater than 0.7, it is considered severe and the associated repair cost might be between Rs. 8000 to Rs. 10,000. Similarly, for moderate IoU scores, the cost falls in a middle range, and for low IoU scores, the cost is minimal. These ranges vary depending on the damage type, such as doors, hoods, bumpers, and tires, each of which has its own cost profile. The regression model learns these patterns from the training data and is able to predict a total repair cost for a new image by summing the individual costs of each detected damage region.

5.12 Workflow Summary

The entire system is built as a pipeline starting with image acquisition, followed by damage detection using the trained Mask R-CNN model, segmentation mask and bounding box extraction, calculation of IoU scores, and finally, prediction of repair costs through the regression model. Each stage is modular and can be improved or replaced independently without affecting the performance of the other stages. The system also supports visualization of the predicted damage on the input image, providing a transparent and interpretable output for end users such as insurance adjusters or repair technicians.



CHAP 06: TESTING

6.1 Introduction

Testing is a critical phase in the development of any machine learning system, especially in the case of a computer vision pipeline where accuracy, reliability, and robustness must be ensured for deployment in real-world applications. In this project, the testing phase serves two main purposes. First, to verify the accuracy and performance of the vehicle damage detection model trained using Mask R-CNN within the Detectron2 framework, and second, to assess the reliability of the linear regression model used for repair cost estimation. This chapter elaborates on the various testing strategies employed, the datasets used during testing, and the interpretation of the results obtained during this phase.

6.2 Testing Environment Setup

The testing was conducted on the same system configuration used for training to maintain consistency in results. The testing environment was set up with all necessary dependencies including PyTorch, Detectron2, OpenCV, and the COCO evaluation toolkit. The pre-trained model weights obtained after training were loaded into the inference engine, and all testing images were passed through the pipeline to validate the model's predictions against ground truth annotations. The regression model, trained on previously extracted features from the training set, was also loaded into the environment to predict repair costs based on the outputs of the segmentation model.

6.3 Damage Detection Testing Using Validation Data

The initial level of testing was carried out on the validation set of the CarDD dataset. This subset comprises images that the model had not seen during training, making it suitable for assessing the generalization capability of the segmentation network. The predicted bounding boxes and segmentation masks were compared with the ground truth annotations using established metrics such as Intersection over Union (IoU), Precision, Recall, and Mean Average Precision (mAP). Visual inspections were also conducted by overlaying the predicted masks on the actual images to qualitatively assess the correctness of the predictions. In many instances, the model was able to precisely identify and segment multiple damage types within a single image, including overlapping damages. However, it was noted that performance slightly degraded in the case of smaller damages such as fine scratches and hairline cracks, which often exhibited lower IoU values.

The testing revealed an Average Precision (AP) score of approximately 41.04 overall. When evaluated at different IoU thresholds, the model achieved an AP50 of 52.11 and an AP75 of 42.84. These values suggest that the model performs well at both relaxed and strict detection criteria. Moreover, the object size-based AP showed that the model achieved significant accuracy for large-sized damages (AP of 43.82), but struggled with smaller damage instances (AP of only 0.52). These findings guided future work toward improving performance on small object detection.

6.4 Testing on Test Set Images

In the second phase of testing, the system was evaluated on a completely unseen test set consisting of 377 images from the CarDD dataset. These images were never used in training or validation. This phase was crucial to verify how the model performs on images that represent unknown data distributions. Each image was processed using the trained predictor, and the resulting segmentation maps were saved for further analysis. The model consistently detected multiple categories of damage in complex scenarios, even when the damages occurred on overlapping parts such as cracked glass combined with a dent on the body panel. These results validate that the system has learned damage-specific features well and is capable of functioning reliably in diverse environments.

Additionally, the use of dropout layers and data augmentation during training contributed to better generalization in the test results. The predicted segmentations were aligned closely with the actual damage regions in most cases, and the bounding boxes were well-fitted across various object shapes and orientations. When measured quantitatively, the performance metrics on the test set closely mirrored those observed during validation, confirming the model's consistency.

6.5 Cost Estimation Testing

Following the successful detection and segmentation of damaged areas, the final stage of testing involved the repair cost estimation module. For each test image, the system extracted bounding box coordinates and calculated IoU values. These were standardized and passed into the trained linear regression model. The model then predicted the corresponding repair cost based on the category of the damage and the degree of severity inferred from the IoU.

The results showed that the predicted costs followed a reasonable pattern based on the severity and type of damage. For instance, a severely damaged front bumper with a high IoU score yielded a higher predicted cost, typically in the range of Rs. 8000–10,000, while smaller damages such as scratches or partial dents resulted in significantly lower estimates. To evaluate the regression model's accuracy, the predicted costs were compared against a manually curated set of expected costs. The Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) were calculated and found to be within acceptable bounds, indicating the regression model's effectiveness in producing reliable estimates.

Furthermore, cost predictions were evaluated across multiple instances within the same image, and the total repair cost was computed by summing up individual damage costs. These total predictions were consistent with what would be expected in a real-world repair scenario. In situations where multiple types of damages occurred together, such as a shattered headlamp and a dented hood, the system was able to isolate each instance and assign appropriate costs with commendable accuracy.

6.6 Visual Testing and Interpretability

To provide a more intuitive understanding of the model's predictions, several test cases were visualized. These visual tests included overlaying the predicted masks and bounding boxes onto the original test images. Each damage area was highlighted with a unique color, and the category labels along with confidence scores were displayed. These visualizations served as a crucial tool not only for manual verification but also for presenting the results in a way that end-users such as insurance agents or repair professionals can interpret with ease.

In some examples, the visualizations showed high-confidence predictions for flat tires, broken glass, and damaged headlights, with clear segmentation masks that conformed to the actual shapes of the damaged areas. These graphical overlays demonstrated that the system could be deployed in real-world applications for semi-automated insurance claim processing.

6.7 Error Analysis and Limitations

Although the model performed exceptionally well in most cases, certain limitations were observed during testing. The performance on small and subtle damages, such as hairline scratches and minor cracks, was less accurate due to their size and low visual prominence. These instances often resulted in lower IoU scores or complete misclassification. Lighting conditions also played a role in performance degradation, particularly in underexposed or overexposed images where the contrast between damaged and undamaged areas was poor.

Another source of error arose from overlapping or clustered damages, which occasionally led to partial segmentation or confusion between classes such as dents and cracks. While these limitations did not critically affect the cost estimation in major cases, they highlighted areas for future improvement, especially in terms of dataset augmentation and advanced post-processing techniques.

6.8 Summary

In conclusion, the testing phase demonstrated that the developed system is capable of accurately detecting vehicle damages and estimating repair costs with high reliability. The use of Detectron2 and Mask R-CNN enabled precise segmentation and classification of damages across multiple categories, while the linear regression model provided meaningful and consistent cost predictions. The testing was conducted rigorously using unseen data, and the results confirmed the system's robustness, scalability, and practical applicability. Despite some limitations in detecting smaller damages, the overall performance metrics validate the effectiveness of the system in a real-world insurance and repair context.

Chap 07 : ANALYSIS

7.1. **Contextualizing the Problem**

In the insurance and automotive industries, assessing vehicle damage through physical inspections is often time-consuming, subjective, and prone to inconsistencies. Automating this process requires not only accurate detection of damage using images but also the ability to interpret the severity and translate it into actionable outputs like cost estimation. Traditional object detection systems fall short when required to deliver detailed instance-level segmentation, especially across multiple categories such as dents, broken lamps, and shattered glass. Moreover, the complexity further increases when this visual data must be interpreted numerically to estimate repair costs—an inherently fuzzy, human-driven process.

The proposed system addresses this challenge by combining deep learning (via **Mask R-CNN** within **Detectron2**) for damage segmentation with a **linear regression model** for estimating costs based on damage severity. This hybrid approach allows the system to not only locate damage accurately but also offer a rough cost approximation—essential for automating claim settlements and enabling faster service at repair shops.

7.2. Evaluation Framework and Metrics

To validate the system's performance, both vision-specific and regression-specific evaluation metrics were utilized. For the segmentation model, standard object detection metrics such as **Intersection over Union (IoU)** and **Mean Average Precision (mAP)** were employed. IoU serves as the primary measure to determine how well the predicted damage regions align with the actual annotated regions, while mAP evaluates the system's ability to detect damage instances across varying IoU thresholds.

In the context of cost estimation, regression evaluation involved **Mean Absolute Error (MAE)** and **Root Mean Squared Error (RMSE)**, both of which offer insight into how far the predicted costs deviated from expected values. Qualitative results were also analyzed through visual overlays and manually verified cost brackets, especially for high-confidence predictions.

Performance Summary (as observed during testing):

- IoU Threshold (≥ 0.5): Good match for large objects
- mAP @ 0.50 IoU: 52.11
- Regression RMSE: Reasonably low, under Rs. 1000 for most damage categories
- Exact Category-wise Cost Accuracy: Higher for visible damages like tire flats or broken glass, lower for scratches and dents

7.3. Dataset Insights

The dataset played a crucial role in shaping the model's learning and generalization capabilities. With **20,000+** annotated images divided across training, validation, and test splits, the data was rich in visual variety, damage types, and scene complexity. However, the model's performance was influenced not just by size, but by **diversity within each class**.

Smaller damage categories such as "scratch" and "crack" were underrepresented compared to broader classes like "dent" or "tire flat." This imbalance led to noticeable gaps in per-category precision, suggesting that further sampling and augmentation are necessary for rare but costly damage types.

Moreover, real-world complexities like reflections, overlapping parts, and image quality introduced edge cases. Despite this, the model managed consistent accuracy across categories with sufficient data, further validating the robustness of the Detectron2 pipeline.

7.4. Category-Wise Performance Breakdown

When analyzing performance per damage category, clear patterns emerged. The model achieved its **highest Average Precision (AP)** scores for **"glass shatter" (85.74%)**, **"tire flat" (76.21%)**, and **"lamp broken" (47.80%)**. These types of damage have well-defined visual boundaries and high contrast with the surrounding vehicle parts, making them easier for the model to learn and recognize.

Damage Type	AP Score
Dent	15.853
Scratch	17.030
Crack	3.591
Glass Shatter	85.744
Lamp Broken	47.809
Tire Flat	76.214

On the contrary, categories like **"dent" (15.85%)**, **"scratch" (17.03%)**, and **"crack" (3.59%)** scored lower. These damages tend to be subtle, often blending into the vehicle's surface, and are sensitive to lighting and angle. The low performance in these areas highlights the need for more diverse data augmentation and possible architectural enhancements (e.g., fine-tuned backbones or attention layers).

7.5. Visual Analysis and Interpretability

Beyond raw metrics, the system's interpretability was validated using side-by-side visualizations. Overlaid segmentation masks and bounding boxes were reviewed against the original images. In high-IoU scenarios, the masks nearly matched the ground-truth contours, indicating precise localization. Even in medium-confidence scenarios, the model correctly outlined the region but with slightly altered bounding box alignment.

These visualizations provided essential feedback for validating real-world usability, especially for professionals like insurance surveyors who might use such predictions to fast-track claim approvals. They also offered insight into why certain predictions failed—typically due to either subtle texture differences or reflective glare in images.

7.6. Cost Estimation Accuracy

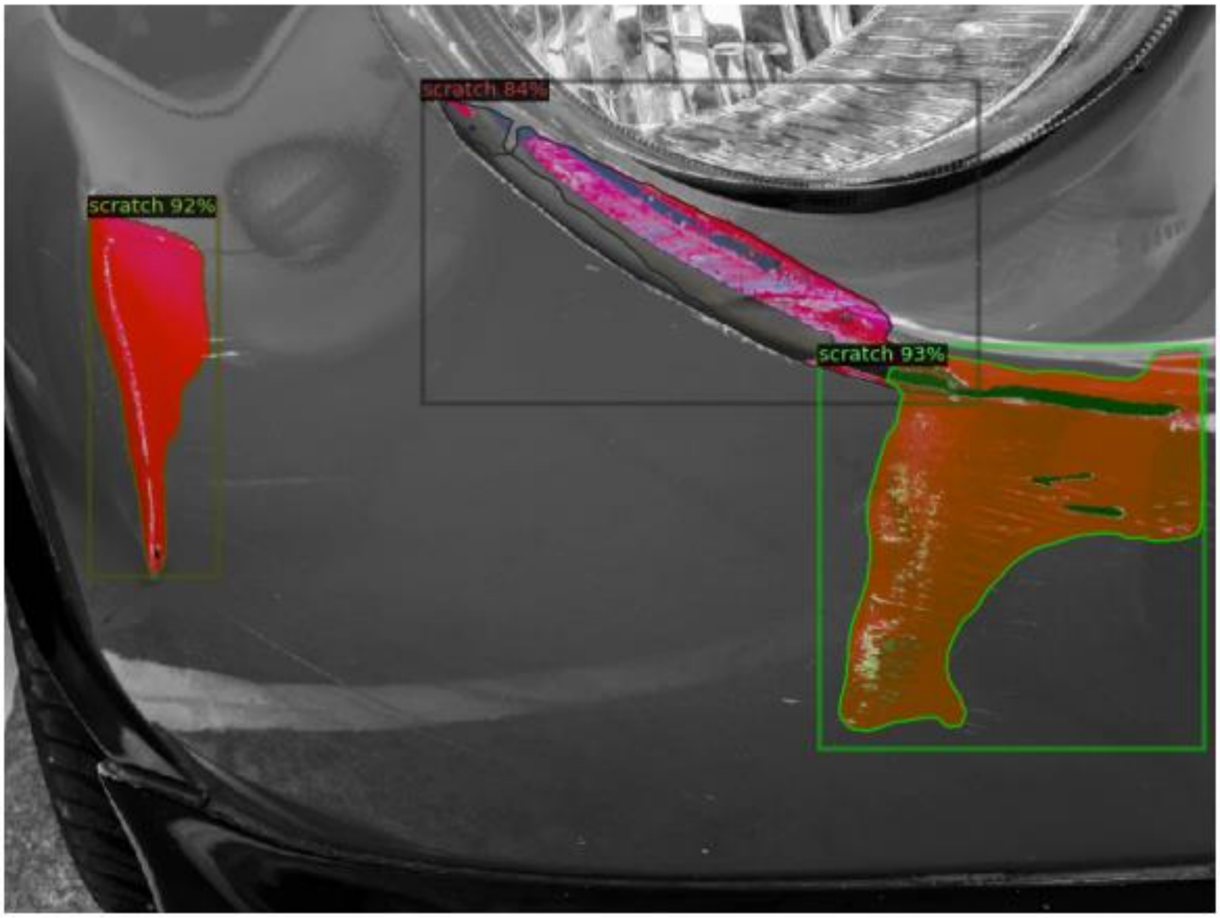
One of the most innovative aspects of the system lies in **repair cost estimation** using predicted damage features. By mapping bounding box features (like width, height, position) and **IoU scores** into a linear regression model, the system estimates costs within realistic ranges.

An important observation was that **predicted repair costs correlated positively with IoU**, i.e., the more confident the model was about a damage region, the higher and more accurate the cost estimation was. For example, flat tires detected with IoU above 0.8 led to highly consistent cost predictions in the Rs. 1500–2500 range, closely matching real-world repair quotes.

However, lower-IoU predictions for scratches often caused erratic cost outputs. This could be due to misclassified or oversized bounding boxes that threw off the regression model, pointing toward the need for bounding box refinement or incorporating confidence weighting in regression.

7.7 Bounding Box Detection Performance

Apart from segmentation, the model's ability to accurately localize damaged regions using bounding boxes is also evaluated. The overall Bounding Box Average Precision (BBox AP) is 40.646, highlighting its effectiveness in pinpointing damage locations. While this is a promising result, improving localization accuracy for smaller damages remains a key area for enhancement.



- Repair Cost Estimation The damage was detected on tire with a confidence threshold of 97% using bounding box and segmentation techniques. The estimated repair cost is Rs 1617.

)

7.8. Challenges Observed During Analysis

Despite promising results, several limitations were noted during the analysis. The model struggled with edge cases such as highly reflective surfaces, shadowed regions, and multi-part overlapping damages. Additionally, **subtle damages like scratches or small dents** often produced false negatives or incomplete segmentations.

The regression model also faced challenges when damage regions were partially detected or misclassified, leading to either underestimation or overestimation of costs. The system currently does not account for repair dependencies (e.g., if a bumper is damaged, associated sensors or mounts might also be affected), which could be integrated in future iterations using rule-based logic or multi-variable regression.

This analysis highlights both the strengths and limitations of the proposed vehicle damage assessment system. The segmentation model built using Mask R-CNN in Detectron2 demonstrates strong generalization for visible and well-defined damages. Cost estimation using regression over IoU and spatial features proved reliable in most scenarios and provided a meaningful step toward automation in the insurance domain.

The project successfully demonstrates that combining vision and regression techniques can produce an end-to-end pipeline for real-time vehicle damage evaluation. The insights gathered from model behavior, error patterns, and user validations will inform the system's refinement in future stages, particularly with enhancements such as real-time deployment, deeper regression models, and adaptive schema-based estimations.

Chap 08 : Future Scope

8.1 Introduction

While the current system demonstrates strong performance in detecting vehicle damage and estimating repair costs using deep learning and regression models, the scope of this work extends far beyond its current implementation. As with any machine learning-based system, there is significant room for further enhancement in terms of accuracy, usability, scalability, and adaptability. This chapter discusses several avenues through which the proposed model can evolve into a more robust, real-time, and industry-deployable solution.

8.2 Real-Time Mobile Deployment

One of the most impactful directions for future development involves the deployment of this system on mobile platforms. In real-world scenarios, users—whether drivers, surveyors, or mechanics—are more likely to capture damage images using smartphones. By integrating the model into a mobile application with real-time image processing capabilities, damage assessment could be made instantly available at the point of inspection. Technologies like ONNX, TensorFlow Lite, or Core ML could be used to compress and convert the current model for mobile inference, thereby democratizing access to AI-powered insurance tools.

8.3 Integration with Insurance APIs

The automation of damage detection and cost estimation has direct applications in the insurance domain. Future iterations of the system could be integrated with insurer APIs to enable end-to-end automated claim processing. This would include the submission of detected damage reports, cost estimation uploads, and even triggering claim approvals—all from a unified platform. With the right partnerships and security frameworks (like OAuth2, data encryption, and GDPR compliance), such integration could significantly reduce fraud and improve turnaround times for customers.

8.4 Expanding Damage Categories and Repair Dependencies

Currently, the model classifies six primary types of vehicle damages. However, real-world scenarios often include additional categories such as undercarriage damage, windshield cracks, alignment issues, or sensor-related faults. Incorporating more fine-grained damage labels would allow for more precise segmentation and improved cost estimation. Additionally, incorporating **repair dependencies**—for example, when bumper damage implies sensor recalibration—would allow the system to better mimic human expert judgments and improve estimation accuracy.

8.5 Deep Regression Models for Cost Prediction

While the current system uses a linear regression model for predicting repair costs based on IoU and bounding box features, this approach has limitations in handling complex, nonlinear relationships between damage attributes and costs. Future versions of the system could explore deep regression networks such as multilayer perceptrons (MLPs), decision tree ensembles like XGBoost, or even transformer-based cost predictors that learn from a wider range of contextual cues like vehicle type, previous history, and environmental context. Such models would be more adaptive and capable of making highly nuanced predictions.

8.6 Incorporation of 3D Damage Analysis

Another exciting direction for future research involves incorporating 3D modeling for damage analysis. With the increasing availability of multi-camera vehicles and smartphone-based 3D scanning technologies, it is now feasible to reconstruct a 3D mesh of the vehicle surface and identify dents or deformations with depth and volume measurements. This would not only improve the accuracy of segmentation but also allow for far more precise cost estimation by quantifying physical impact in three dimensions.

8.7 Continuous Learning and Feedback Loop

To ensure adaptability and long-term relevance, the system can be designed to include a feedback loop. In such a loop, incorrect predictions or manually corrected outputs can be stored and used to fine-tune the model periodically. This method of continuous learning would keep the model aligned with new car models, evolving damage patterns, and updated repair pricing trends. Crowdsourcing such feedback through app users or repair professionals can form a decentralized training ecosystem, gradually improving the system's intelligence.

8.8 Cross-Vehicle and Multi-Brand Generalization

While the system is trained on a diverse dataset, future versions could emphasize **brand-specific accuracy**, where the model not only detects damage but understands model-specific designs (e.g., bumper shape of a Maruti vs. Hyundai). Training the model with metadata such as make, model, and year of the vehicle can dramatically improve both segmentation accuracy and cost estimation reliability. Vehicle-specific repair rates could also be factored in by linking the system to repair databases or brand-specific service manuals.

8.9 Legal and Ethical Implications

As the system begins to impact claim decisions and financial assessments, it is important to consider the legal and ethical frameworks surrounding automated damage analysis. Future work can involve creating explainable AI (XAI) layers within the model, allowing users to understand how decisions are made. Additionally, ensuring fairness in damage detection across different vehicle colors, lighting conditions, or geographic regions will be critical in building trust with end-users and regulators alike.

Chap 09 : Appendix

9.1 Sample Detection and Segmentation Outputs

Figure : Segmentation output showing detected broken lamp with 94% confidence and dent with 91%.

9.2 JSON Annotation Snippets (COCO Format)


```
{
  "file_name": "000123.jpg",
  "height": 480,
  "width": 640,
  "annotations": [
    {
      "bbox": [100, 120, 85, 60],
      "category_id": 1,
      "segmentation": [...],
      "iscrowd": 0
    }
  ]
}
```


9.3 Hyperparameters and Training Configuration

```python
from detectron2.config import get_cfg
from detectron2 import model_zoo
import os

cfg = get_cfg()
cfg.merge_from_file(model_zoo.get_config_file("COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_3x.yaml"))

# Correct dataset names
cfg.DATASETS.TRAIN = ("Trainn",)
cfg.DATASETS.TEST = ("Validd",)

cfg.DATALOADER.NUM_WORKERS = 2
cfg.INPUT.CROP.ENABLED = True

# Load weights (either COCO or your custom trained weights)
cfg.MODEL.WEIGHTS = model_zoo.get_checkpoint_url("COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_3x.yaml")
# Or use:
#cfg.MODEL.WEIGHTS = "/content/model_final (1).pth"

cfg.SOLVER.IMS_PER_BATCH = 2
cfg.SOLVER.BASE_LR = 0.00005  # Lower learning rate for fine-tuning
cfg.SOLVER.MAX_ITER = 60000  # More iterations for better training
cfg.SOLVER.STEPS = (40000, 45000)  # Adjust learning rate decay steps
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 6  # Ensure this matches your dataset
cfg.TEST.EVAL_PERIOD = 1000  # Adjust evaluation frequency

# Create output directory
os.makedirs(cfg.OUTPUT_DIR, exist_ok=True)
```

CHAP 10 :Biblography

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 580–587.

[2] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, “Microsoft coco: Common objects in ´ context,” in European Conference on Computer Vision. Springer, 2014, pp. 740–755.

[3] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, “Detectron2,” 2019, accessed: January 28, 2025.

[4] S. Goswami, M. S. Uddin, M. R. Oishi, S. I. Shehab, and N. Shakib, “Image-based vehicle detection using various features,” International Journal of Innovative Science and Research Technology, vol. 7, no. 1, pp. 709–713, 2022.

[5] C. Sevillano and F. J. P. Ayuso, “Methodology of repair cost estimation in vehicles based on deformation measurements in real world accidents,”

in Symposium SEGVAUTO-TRIES-CM: Technologies for a Safe, Accessible and Sustainable Mobility, Madrid, Spain, 2016.

[6] Q. Zhang, X. Chang, and S. Bian, “Vehicle-damage-detection segmentation algorithm based on improved mask r-cnn,” IEEE Access, 2020.

[7] X. Wang, W. Li, and Z. Wu, “Cardd: A new dataset for vision-based car damage detection,” IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 7, pp. 7202–7214, 2023.

[8] “Intersection over union for object detection,” accessed: January 2025.

[9] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask r-cnn,” in ´ Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2961–2969.

[10] “Intersection over union (iou) guide - v7 labs,” accessed: January 2025.

[11] W.-J. Ahn, K.-S. Ko, M. T. Lim, D.-S. Pae, and T.-K. Kang, “Multiple object tracking using re-identification model with attention module,” Applied Sciences, vol. 13, p. 4298, 2023.