# Auto Inspect AI: Vehicle Damage Assessment System

## 1. Front Page

Title: Auto Inspect AI: AI-Powered Vehicle Damage Inspection

Subtitle: Automating Car Damage Detection & Severity Estimation

Team Members: Mohammed Zeeshan (225U1A6680)

Rakesh Reddy (225U1A66B8)

Shaik Mudassir (225U1A668A9)

Satya Krishna (225U1A6685)

Image: A car with damaged areas highlighted using AI.

## 2. Introduction

### Slide 1: Problem Statement

• Current Issue: Manual vehicle damage assessment is time-consuming, costly, and inconsistent.

• Need for AI: Automating the process can improve efficiency, accuracy, and cost estimation.

### Slide 2: Objective

• Develop an AI-powered web platform for real-time vehicle damage detection and severity estimation.

• Use computer vision and deep learning models for accurate damage analysis.

## 3. Literature Survey

### Slide 3: Related Work (Base Paper Insights)

• Traditional Approaches: Manual inspection, insurance adjusters, and expert assessment.

• AI-Based Approaches: YOLO, Mask R-CNN, XGBoost for damage estimation.

• Gaps in Existing Work:

- Lack of real-world datasets for in-the-wild damage detection.

- No severity estimation models integrated with cost prediction.

### Slide 4: Real-Time Motivation

• Insurance Industry: Faster claim processing.

• Car Rentals: Automated inspection before & after rentals.

• Service Centres : AI-assisted damage reporting for repair estimation.

## 4. Proposed System

### Slide 5: Key Features

✅ Vehicle Detection (YOLOv4) – Identifies and crops the car from images.

✅ Car Angle Classification (ResNet-18) – Determines the car’s orientation.

✅ Damage & Parts Segmentation (Mask R-CNN) – Identifies and localizes damages.

✅ Feature Fusion (IoU-based scoring) – Combines image-based and structured data.

✅ Damage Severity Estimation (LightGBM) – Predicts damage level and cost.

✅ Web Dashboard (Django, React.js) – User-friendly interface for damage reports.

## 5. Requirements

### Slide 6: Hardware & Software Requirements

#### Hardware:

• GPU (NVIDIA RTX 3060 / A100) – For training and inference.

• Cloud Storage (AWS S3, Google Cloud) – For image data storage.

• Server (Ubuntu/Linux-based) – Running Django & AI models.

#### Software:

• Deep Learning: TensorFlow, PyTorch.

• Computer Vision: OpenCV, Pillow, Numpy.

• Backend: Django, Flask.

• Frontend: React.js, HTML, CSS.

• Database: PostgreSQL, Firebase.

### Slide 7: Functional & Non-Functional Requirements

#### Functional Requirements

• Users can upload car images for analysis.

• The system should detect damaged areas and classify severity.

• The system should provide a repair cost estimate.

#### Non-Functional Requirements

• Accuracy: AI models should have at least 85% accuracy.

• Scalability: System should handle multiple image uploads at once.

• Security: Secure user data & insurance claims.

### Slide 8: Other Requirements

• Integration with insurance companies & car rental services.

• APIs for third-party applications (e.g., mobile apps).

## 6. Architecture

### Slide 9: System Architecture Overview

• Input: User uploads vehicle images.

• Processing:

• - YOLOv4 detects the car.

• - Mask R-CNN segments damage & parts.

• - Feature fusion combines image & structured data.

• - XGBoost predicts severity & cost.

• Output: Generates damage report & cost estimate.

📌 Include a system diagram showing how models interact.

## 7. Design (UML Diagrams)

### Slide 10: UML Diagrams

#### Class Diagram:

• Shows interaction between User, AI Engine, Damage Detector, and Database.

#### Use Case Diagram:

• Actors: User, Insurance Agent, Repair Shop.

• Use Cases: Image Upload, Damage Detection, Cost Estimation, Report Generation.

## Final Slide: Conclusion & Future Scope

### Conclusion:

• Auto Inspect AI automates vehicle damage detection with high accuracy.

• Reduces manual inspection time and improves cost estimation.

### Future Scope:

• VR-based inspection models.

• Integration with mobile apps.

• Multi-language support.